CICS-JUGGLER™

Technical Reference Guide

Release 5.4

Publication Number
GP51-0420-1

WORLD HEADQUARTERS
UNICOM Systems Inc.
1032 Cove Way
Beverly Hills, CA 90210
(818) 838-0326

UNICOM Systems, Inc
UNICOM PLAZA, Suite 310
15535 San Fernando Mission Blvd.
Mission Hills, CA 91345
(818) 838-0606, Fax: (818) 838-0776

FAR EAST DIVISION
UNICOM Systems Int'l Korea, Ltd.
Hawangshimni Dong 301 Suite 201
Songdong-ku Seoul, Korea
(02) 296-5476, Fax: (02) 294-5446

UNICOM SYSTEMS, INC.

Unicom Systems, Inc.

15535 San Fernando Mission Blvd.
Mission Hills, California **91345**

Sixth Edition – November 13, 1998

The information in this document applies to Release 5.4
Of the CICS-JUGGLER™ product for all operating environments.

Published by UNICOM SYSTEMS Inc.
15535 San Fernando Mission Blvd. Suite 310
Mission Hills, California 91345

CICS-JUGGLER

Release 5.4

Table of Contents

PAGE

INTRODUCTION

A POWERFUL PRODUCTIVITY TOOL

CICS-JUGGLER is one of the most significant software developments for improving productivity in the world of on-line systems. It is a multiple session manager that allows you to conduct up to nine concurrent CICS applications. There is no need for any hardware changes, nor is there any need to modify your existing software. And, of course, there is no degradation to the system nor to user response time.

With **CICS-JUGGLER** it is now possible to:

- **HAVE JUGGLER FACILITIES ON EVERY MAINFRAME TERMINAL WITHOUT ANY CHANGES TO HARDWARE OR SOFTWARE**

- **EASILY TRANSFER INFORMATION FROM ONE TRANSACTION TO ANOTHER USING THE COPY AND PASTE CAPABILITY.**

- **EXPAND TERMINAL CAPABILITIES BY HAVING UP TO NINE VIRTUAL TERMINALS ON EACH PHYSICAL TERMINAL**

- **ELIMINATE THE TIME CONSUMING REQUIREMENT OF LOGGING ON AND OFF WHEN SWITCHING BETWEEN APPLICATIONS.**

You may have up to 9 different virtual terminals on the same physical screen. The effect is similar to having nine separate physical terminals. You can instantly toggle between the sessions using only one keystroke.

With **CICS-JUGGLER** you can increase productivity by having your needed information only one keystroke away ... other applications, help screens, menus, prior transactions, editors, note pads, anything to which you may want to refer.

CICS-JUGGLER is very user friendly and easy to use. It operates by using a PF or PA key. You have complete control over whichever application is currently selected, and you may easily move from one session to another, using only one keystroke.

OPERATION

INITIATING CICS-JUGGLER

The **CICS-JUGGLER** product is very easy to initiate. This would normally be performed once, at the start of your working day and before any other applications have been processed. With the Auto-Start option, the initialization can be performed automatically when you sign on to CICS. It can, of course, be initiated at any time throughout the day.

The command to initiate or start **CICS-JUGGLER** is:

JUGL,ON

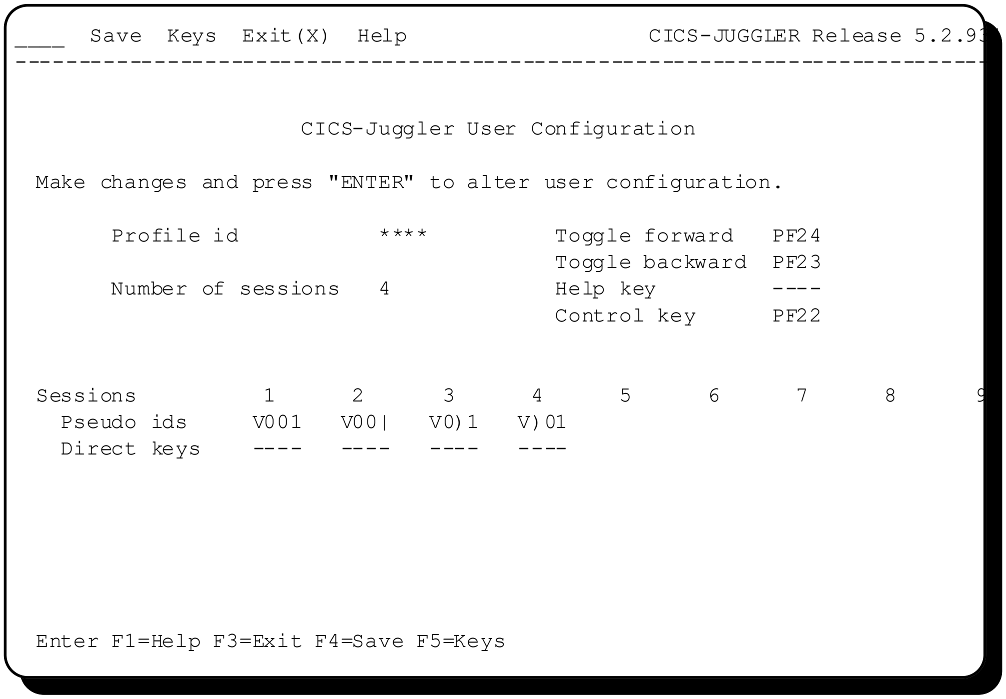
When the **JUGL,ON** command is issued, CICS-JUGGLER will search for a profile to determine the operational specifications to be used on this terminal.

[Note]: If an explicit or default profile is not found, the JUGL,ON command will be treated like a JUGL,INIT command. The JUGL,INIT command is explained in detail in section 13 - *SPECIAL PURPOSE COMMANDS*.

At this point, **CICS-JUGGLER** initialization is complete and the User Configuration Display will appear.

THE CICS-JUGGLER USER CONFIGURATION DISPLAY

Upon completion of the JUGL,INIT or JUGL,ON commands or any time a JUGL,INQ or JUGL (no command) is issued, the following User Configuration Display will appear:



```
____  Save  Keys  Exit(X)  Help                   CICS-JUGGLER Release 5.2.9
-------------------------------------------------------------------------------


                       CICS-Juggler User Configuration

 Make changes and press "ENTER" to alter user configuration.

       Profile id           ****          Toggle forward   PF24
                                          Toggle backward  PF23
       Number of sessions   4             Help key         ----
                                          Control key      PF22


 Sessions          1      2      3      4      5      6      7      8      9
   Pseudo ids     V001   V00|   V0)1   V)01
   Direct keys    ----   ----   ----   ----






 Enter F1=Help F3=Exit F4=Save F5=Keys
```

Initially the fields of the display will correspond to the fields of the User Profile that is active for this terminal (or will correspond to the responses made during the JUGL,INIT process). Some values can be changed, and once alterations to this screen are performed, those changes will remain in effect until CICS-JUGGLER is deactivated on this terminal.

FIELDS OF THE USER CONFIGURATION DISPLAY

The fields of the User Configuration display are described below. For information on operating the Action Bar, see *OPERATION OF ACTION BARS*, later in this section.

CONTROL KEY This is used to specify a PF or PA key to be pressed that will invoke the CICS-JUGGLER Control Window. For more information on the Control Window, see *THE CICS-JUGGLER CONTROL WINDOW*, later in this section.

Valid entries are any PF or PA key or CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 11 - *SPECIAL CONSIDERATIONS*).

HELP KEY This is used to specify a PF or PA key to be used as the CICS-JUGGLER Help Access key. For more information on the Help Access key, see section 07 - *THE HELP-WINDOWS FEATURE*.

Valid entries are any PF or PA key or CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 11 - *SPECIAL CONSIDERATIONS*).

NUMBER OF SESSIONS
This is the number of sessions (virtual terminals) that are available to the operator. Note that this field can not be altered. Valid values displayed are:

number	A number from 2 to 9, inclusive.

PROFILE ID This is the ID of the profile that was selected for this terminal or user as specified in the Auto-Init table. Note that the Profile ID can not be changed.

SESSIONS DIRECT KEYS
Direct session toggle keys. If this feature is desired, code a PF or PA key for each virtual terminal present on this physical terminal. The keys correspond one for one with each virtual terminal. That is, the first key coded will transfer control directly to virtual terminal 1, the second to virtual terminal 2, etc.

You may omit one or more keys in the list by skipping that position. If this is done, the virtual terminal corresponding to that key position will not have a direct key assigned to it.

Valid keys are any PF or PA key and CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 11 - *SPECIAL CONSIDERATIONS)*. The key selected must not be the same as any other designated key.

See the subject entitled *DIRECT SESSION KEY OPERATION* for an explanation of the use of Direct Session keys.

SESSIONS PSEUDO IDS
These fields display the pseudo terminal IDs that are in use for this terminal. These fields cannot be altered.

TOGGLE BACKWARD KEY

This is the PF or PA key to be used to move "backward" from one virtual terminal to the next lower terminal number. Each time the Toggle-Backward key is pressed, control moves from the current virtual terminal to the previous one in sequence until terminal number 1 is reached, at which time control moves to virtual terminal number (NUMBER OF SESSIONS).

Valid entries are any PF or PA key or CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 11 - *SPECIAL CONSIDERATIONS*).

TOGGLE FORWARD KEY

This is the PF or PA key to be used to move "forward" from one virtual terminal to the next higher terminal number. Each time the Toggle-Forward key is pressed, control moves from the current virtual terminal to the next one in sequence until terminal number (NUMBER OF SESSIONS+1) is reached, at which time control moves to virtual terminal number one. Valid entries are:

Any PF or PA key or CSEL (see USING THE CURSOR-SELECT KEY AS A HOT KEY in section 11 - SPECIAL CONSIDERATIONS).

ENTERING COMMANDS AT THE USER CONFIGURATION DISPLAY

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

ENTER Apply any changes that you have made to the configuration.

PF1 HELP Display a Help window. If the cursor is positioned in an unprotected field, the help will pertain to that field, otherwise the help will pertain to the JUGL display.

PF3 EXIT Return to CICS.

PF4 SAVE Save the current settings of this User Configuration in the profile specified in the PROFILE field. This will permanently alter the profile, and affect all other users of this profile. Note that since multiple users may share a profile, this profile may be protected against the SAVE function. For more information on the User Profile, see *THE USER PROFILE TABLE* in section 10 - *CUSTOMIZATION*.

PF5 KEYS Display the User Function Key Assignment screen. (This is explained in more detail below, in *THE USER FUNCTION KEY ASSIGNMENT DISPLAY).*

THE USER FUNCTION KEY ASSIGNMENT DISPLAY

The User Function Key Assignment display allows an operator to assign nearly any valid CICS-JUGGLER command to a PF key. This display is accessed by pressing PF5 from the User Configuration display, or selecting the KEYS pull-down menu from the action bar.

FUNCTION KEY ASSIGNMENTS

The current settings of keys and commands will display in a popup window. You can change, add or erase any field. The following is a list of valid assignments along with a brief description. The list has been broken down by type of command.

TOGGLE AND CONTROL COMMANDS

TB	Toggle-Backward
TF	Toggle-Forward
Tx	Toggle directly to session x.
HELP	Specify as HELP key.
KEYS	Display the PF key assignments.
MENU	Display Control Window.
ST	Display the toggle function window.

CUT AND PASTE COMMANDS

COPY	Copy the field at the position of the cursor.
PASTE	Paste the clipboard at the position of the cursor.
STACK	Copy the field at the position of the cursor and append to the bottom of the clipboard.
UNPROTECT	Unprotect all fields on the screen.

TERMINATING CICS-JUGGLER

CICS-JUGGLER may be terminated in either of two ways:

1). Enter JUGL,OFF. This completely terminates CICS-JUGGLER at this terminal.

2). Do a CICS SIGN-OFF in any virtual terminal. When this happens, the following message will appear:

DO YOU WANT TO TERMINATE CICS-JUGGLER?

The default response is yes ("Y"). To terminate **CICS-JUGGLER** on this terminal, simply press ENTER. However, if you do not wish to terminate **CICS-JUGGLER**, enter a no ("N"). This will allow you to sign off CICS while leaving all virtual terminals intact.

[Note]: If the REQUIRE JUGL,OFF AT LOGOFF option has been defined in the User Option Table (see section 10 - *CUSTOMIZATION)*, the following message will be displayed:

CICS-JUGGLER MUST BE TERMINATED TO LOG-OFF.

If this message appears, the terminal user must terminate **CICS-JUGGLER**, prior to CICS sign off.

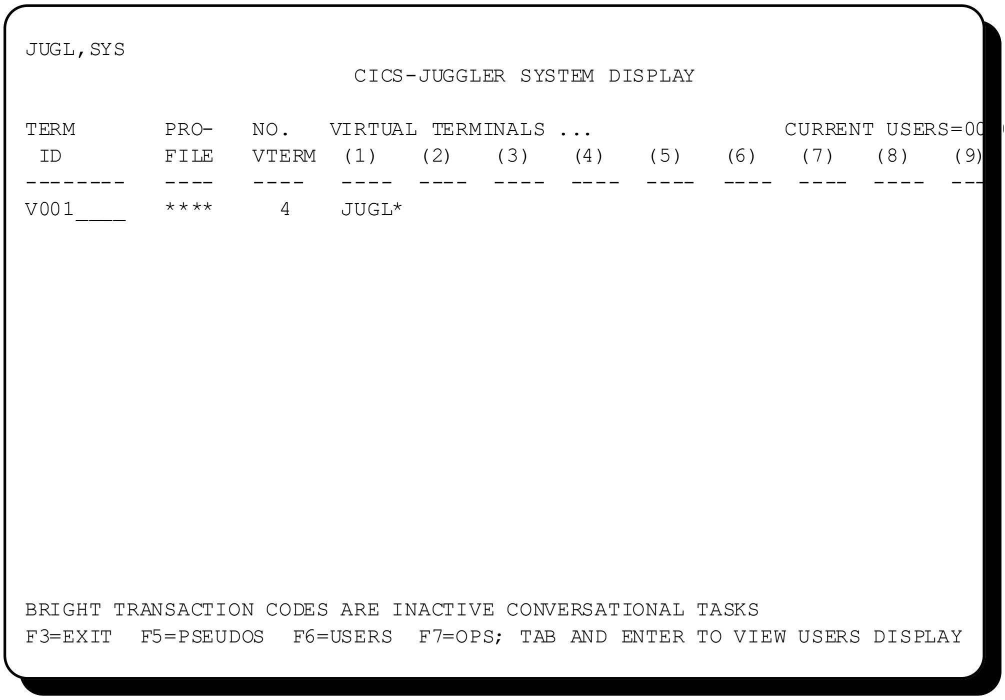
[Note]: If FORCE PURGE AT SIGNON/SIGNOFF has been defined, CICS-JUGGLER will be automatically terminated when a signoff occurs.

THE CICS-JUGGLER SYSTEM DISPLAY

The System Display provides a means of displaying the current status of **CICS-JUGGLER** users in the system environment. The display is invoked by entering the command:

JUGL,SYS

The display appears generally as follows:



```
JUGL,SYS
                          CICS-JUGGLER SYSTEM DISPLAY

TERM        PRO-   NO.   VIRTUAL TERMINALS ...              CURRENT USERS=00
 ID         FILE   VTERM  (1)   (2)   (3)   (4)   (5)   (6)   (7)   (8)   (9)
--------    ----   ----   ----  ----  ----  ----  ----  ----  ----  ----  ---
V001____    ****     4    JUGL*









BRIGHT TRANSACTION CODES ARE INACTIVE CONVERSATIONAL TASKS
F3=EXIT  F5=PSEUDOS  F6=USERS  F7=OPS; TAB AND ENTER TO VIEW USERS DISPLAY
```

FIELDS OF THE SYSTEM DISPLAY

The following information is displayed for every terminal in the system that is currently using **CICS-JUGGLER**:

1). Physical Terminal ID.
2). Profile ID in use for each terminal.
3). The transaction code of the current transaction activated in each virtual terminal.

An asterisk (*) following the transaction code indicates the virtual terminal at which the operator is currently working.

If a transaction code is high intensity, this transaction is in a conversational state.

Information on up to 16 terminals may be displayed on one screen. Pressing ENTER will continue the display with the next page of terminals.

You may begin the display with a specific terminal, by entering the command:

JUGL,SYS,xxxx

where xxxx is the terminal ID of the desired physical terminal. If that terminal is not currently using **CICS-JUGGLER**, the display will start with the first terminal, just as though you had entered the command: JUGL,SYS.

DISPLAYING THE PSEUDO TERMINAL IDS

When the System Display is on the screen as described above, you may press PF5, which will cause the Active Transaction codes displayed for each virtual terminal to change to the Pseudo Terminal ID that is assigned to each Virtual Terminal. Pressing ENTER will change the display back to transaction codes.

VIEWING ANOTHER TERMINAL SESSION

The JUGL,SYS display screen can also be used on your terminal to view the screen display of another physical terminal. This can be a useful feature for problem analysis, operator training, etc.

In order to use this feature, the terminal to be viewed must be using CICS-JUGGLER. To perform a terminal session view, do the following:

1). From a clear screen, enter the JUGL,SYS command, to display the CICS-JUGGLER System Display.

2). Locate the terminal to be viewed (page forward if necessary).

3). Use the TAB key to position the cursor on the first character of the transaction code displayed in the virtual terminal to be viewed.

4). Press ENTER.

Upon pressing ENTER, the screen display that was last seen in the designated virtual session for that terminal will be displayed on your terminal. You can not respond to the screen. That is, the application program is not active on your terminal, you are simply looking at the terminal display.

When you press ENTER again, you will be returned to the beginning of the JUGL,SYS display.

[Note]: If the virtual terminal where you position the cursor has an asterisk (*) by it, it means that this is the session which the operator at that terminal is currently using. Normally, the display that you see upon pressing ENTER is not what the operator is currently looking at. What you are seeing is the screen that was saved when the operator last pressed the toggle.

However, if the READ BUFFER option is not in effect for that terminal (see section 10 - *CUSTOMIZATION*), you will be viewing the same screen that is currently displayed on the virtual terminal.

USING THE CONTROL CHARACTER TO PERFORM COMMANDS

The control character can be used to perform functions such as Cut & Paste or toggling to a different session. It is keyed in an unprotected field of any screen along with the mnemonic for the function to be performed.

The Control Character is the tilde sign (~) by default. However, this is a customization option that may have been changed at your installation. For more information, see *THE USER OPTIONS TABLE* in section 10 - *CUSTOMIZATION*.

To use, simply tab to an unprotected field, key the control character along with the function to be performed, and press ENTER. For instance, to toggle forward to the next session, enter ~TF.

It can be used only if CICS-JUGGLER is active on the terminal. The control character function will not disturb the data that was already present in the field in which it was keyed.

See *APPENDIX A - SUMMARY OF CICS-JUGGLER COMMANDS* for a list of the commands that can be used in conjunction with the control character.

OPERATION OF CICS-JUGGLER HELP

CICS-JUGGLER provides an on-line Help facility for quickly accessing operator instructions. Although it is not meant to replace this Reference Guide, the Help facility is quite extensive.

The main types of help displays are as follows:

Field Level Help
This is specific help that applies to one field of a CICS-JUGGLER display. This type of help is accessed by tabbing the cursor to the field for which help is desired, then pressing the help key.

Screen Level Help
This is general help that applies to an entire CICS-JUGGLER display. This type of help is accessed by moving the cursor out of any field (such as to the title of the screen), then pressing the help key.

Nested Help This type appears as a help menu while already in help mode. It is accessed by tabbing to the desired option, then pressing ENTER.

Help Index The Help Index is a type of Nested Help that provides detailed help for functions such as Window Mode Operation and Cut and Paste Procedures, and offers a basic description of functions such as Customization. It is accessed from an action bar choice or by issuing the or **JUGL,HELP** commands (explained below).

As briefly explained above, several methods may be used to access help. The method used will determine the type of help.

If a CICS-JUGGLER screen (such as the User Configuration Screen) is displayed, you may tab the cursor to any field in question, and press the help key (for field level help), or you may use the action bar and select any of the help choices on the pull-down menu.

If you are wanting help on a CICS-JUGGLER function and CICS-JUGGLER is active on the terminal, you may key the control character (see note below) along with the an 'H' in any field of the screen and press ENTER. The Help Index will then display. When help is exited, the **~H** will not appear in the field in which it was keyed. Or you may issue the **JUGL,HELP** command from a clear screen.

[Note] The Control Character is the tilde sign (~) by default. However, this is a customization option that may have been changed at your installation. For more information, see *THE USER OPTIONS TABLE* in section 10 - *CUSTOMIZATION*.

All types of help may be exited by pressing PF3 or the CLEAR key. While in nested help, the help key may be pressed to return from the current help display to the previous level.

THE CICS-JUGGLER CONTROL WINDOW

The Control Window allows operators to perform functions such as Direct Toggling without the need to memorize commands and/or PF keys. The Control Window can be displayed by keying the Control Character (see note below), then the letters "MENU", anywhere on the screen, then pressing ENTER. Thus, if the control character is the tilde sign (~), the command **~MENU** (or simply **~M**) typed anywhere on the screen would invoke the Control Window, provided CICS-JUGGLER is active on the terminal. In addition, you may also have a PF or PA key that will invoke the Control Window.

[Note] The Control Character is the tilde sign (~) by default. However, this is a customization option that may have been changed at your installation. For more information, see *THE USER OPTIONS TABLE* in section 10 - *CUSTOMIZATION*.

When invoked, the Control Window is displayed in a popup window. The functions that are available are detailed below. Note that since the Control Window is created by the CICS-JUGGLER Menu generation facility, the menu may have been further customized at your installation.

DISPLAY SESSION SELECTION MENU

This may be used as an alternative to the toggle keys or sessions direct keys. When this option is selected, information about each session such as virtual terminal number and active transactions will be displayed. To perform the toggle, tab the cursor to the desired session number and press ENTER.

DISPLAY FUNCTION KEY ASSIGNMENTS

This option is the same as pressing the KEYS key while viewing the User Configuration Display. For more information, see *THE USER FUNCTION KEY ASSIGNMENT DISPLAY*, earlier in this section.

DISPLAY MESSAGES

This option can be used to display a message that has been broadcast to this terminal. For more information, see section 09 - MESSAGE BROADCASTING.

EXIT CONTROL WINDOW

To exit the Control Window, select this option.

OPERATION OF ACTION BARS

Action Bars are located throughout CICS-JUGGLER and operate according to CUA standards. An Action Bar is a list of available actions that appears at the top of a display. It is much like a function key area, except the action bar does not list the actions directly. Instead, the actions are grouped together on pull-down menus, and the group names are what is listed in the action bar.

The remainder of this topic describes the use of action bars in general terms. This discussion does not attempt to detail the actual contents of every action bar.

INITIATING AN ACTION THROUGH A PULL-DOWN MENU

The technique for initiating actions with an action bar, is to pull down the desired menu, then from the pull-down menu, select the action to be performed. In order to pull down a menu, there are two methods:

- Tab the cursor to the name of the menu, then press ENTER.
- Key the first letter of the name of the menu in the action bar entry field, then press ENTER.

  Note that some action bars may have conflicting menu mnemonic selections. In this case, a menu name will have a letter enclosed in parenthesis. This is the letter to be keyed in the entry field.

Upon selecting a menu, a pull-down menu will drop down from the action bar. Now you may select an action with any of the following methods:

- Tab the cursor to the name of the action, then press ENTER.
- Key the selection number of the action to be performed in the menu entry field, then press ENTER.
- In some instances, an action may be initiated by pressing a function key. For these actions, the function key will be listed in the menu along with the action, and may be pressed to select the desired action.

If you do not wish to invoke an action while a pull-down menu is displayed, you may either select a different menu, select the RESUME function, or press CLEAR to remove the pull-down menu.

INITIATING AN ACTION THROUGH THE FAST PATH METHOD

As an alternate technique for initiating an action with the action bar, the fast path method may be used. For this method, simply key the menu mnemonic and the selection number of the action to be performed in the action bar entry field, then press ENTER.

USING APPLICATIONS WITH CICS-JUGGLER ACTIVE

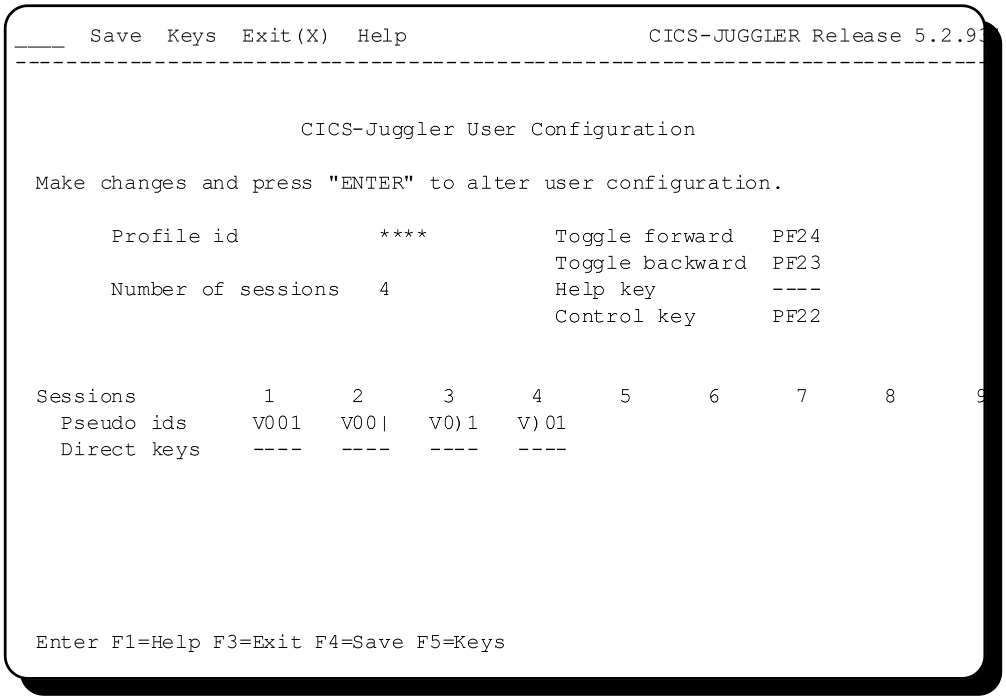
This section will discuss how to operate applications while using **CICS-JUGGLER.** Examples will be given to aid the user.

TUTORIAL

First, sign-on to CICS using your normal operator name and password. Signing on prior to activating CICS-JUGGLER will allow the same security to be established for all virtual terminals.

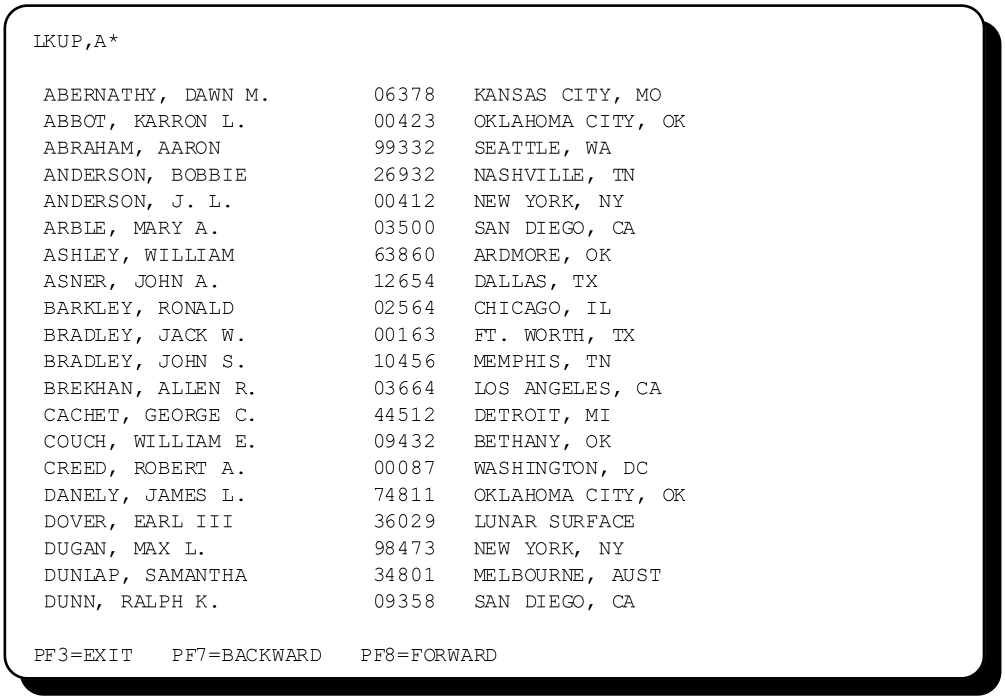
Now initiate CICS-JUGGLER with the JUGL,ON command.

Upon completion, the User Configuration Display will appear with the configuration that has been defined for your terminal.



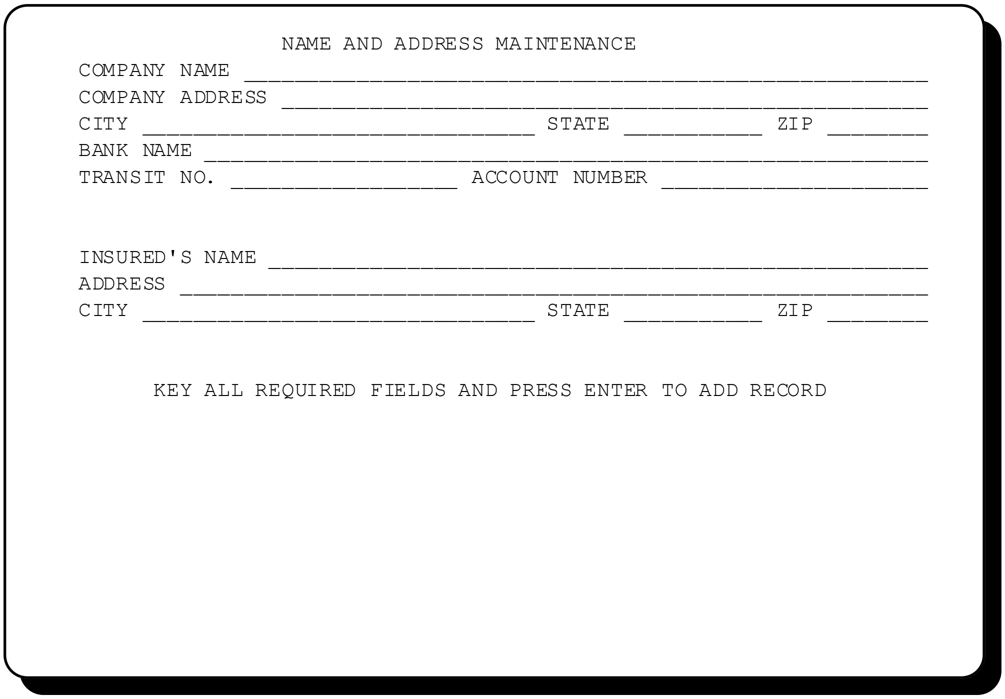
```
____  Save  Keys  Exit(X)  Help                 CICS-JUGGLER Release 5.2.9
------------------------------------------------------------------------------


                     CICS-Juggler User Configuration

 Make changes and press "ENTER" to alter user configuration.

       Profile id           ****          Toggle forward   PF24
                                          Toggle backward  PF23
       Number of sessions   4             Help key         ----
                                          Control key      PF22


 Sessions          1      2      3      4      5      6      7      8      9
   Pseudo ids     V001   V00|   V0)1   V)01
   Direct keys    ----   ----   ----   ----






 Enter F1=Help F3=Exit F4=Save F5=Keys
```

At this point you may press PF3 to exit the display and proceed with whatever application is desired. Follow your normal procedures. For example, the following transaction may be initiated:



```
LKUP,A*

 ABERNATHY, DAWN M.        06378   KANSAS CITY, MO
 ABBOT, KARRON L.          00423   OKLAHOMA CITY, OK
 ABRAHAM, AARON            99332   SEATTLE, WA
 ANDERSON, BOBBIE          26932   NASHVILLE, TN
 ANDERSON, J. L.           00412   NEW YORK, NY
 ARBLE, MARY A.            03500   SAN DIEGO, CA
 ASHLEY, WILLIAM           63860   ARDMORE, OK
 ASNER, JOHN A.            12654   DALLAS, TX
 BARKLEY, RONALD           02564   CHICAGO, IL
 BRADLEY, JACK W.          00163   FT. WORTH, TX
 BRADLEY, JOHN S.          10456   MEMPHIS, TN
 BREKHAN, ALLEN R.         03664   LOS ANGELES, CA
 CACHET, GEORGE C.         44512   DETROIT, MI
 COUCH, WILLIAM E.         09432   BETHANY, OK
 CREED, ROBERT A.          00087   WASHINGTON, DC
 DANELY, JAMES L.          74811   OKLAHOMA CITY, OK
 DOVER, EARL III           36029   LUNAR SURFACE
 DUGAN, MAX L.             98473   NEW YORK, NY
 DUNLAP, SAMANTHA          34801   MELBOURNE, AUST
 DUNN, RALPH K.            09358   SAN DIEGO, CA

PF3=EXIT   PF7=BACKWARD   PF8=FORWARD
```

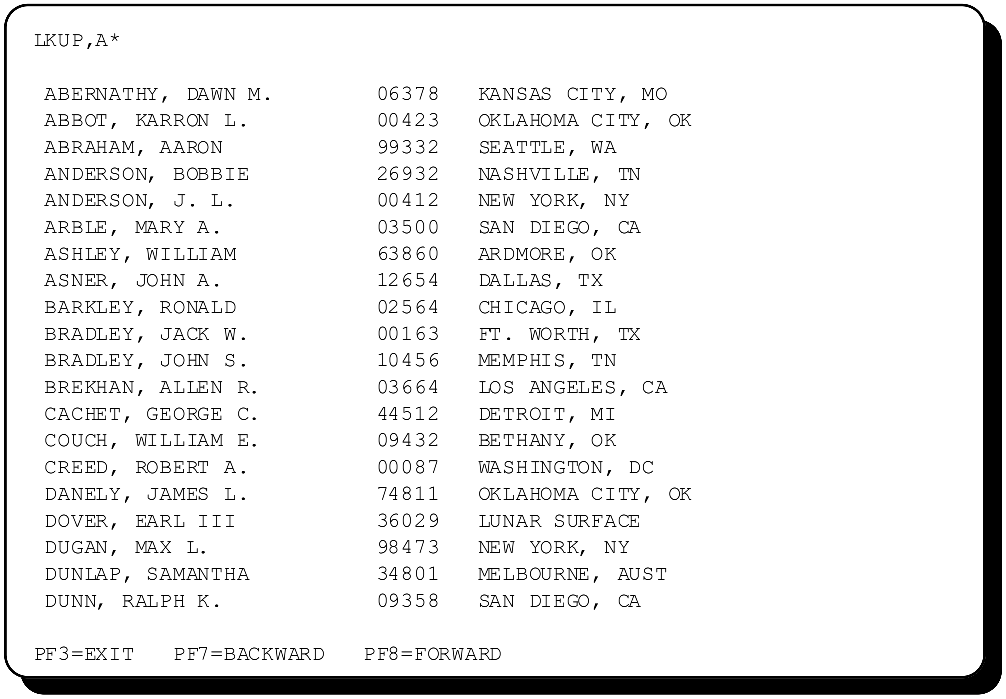
At any time you may press the "Toggle forward" key. Your current application will be saved and CICS-JUGGLER will proceed to the next virtual terminal. It will initially be a blank screen. You may now initiate another CICS application on this virtual terminal.



```
                    NAME AND ADDRESS MAINTENANCE
    COMPANY NAME ______________________________________________________
    COMPANY ADDRESS ___________________________________________________
    CITY _______________________________ STATE ___________ ZIP ________
    BANK NAME _________________________________________________________
    TRANSIT NO. __________________ ACCOUNT NUMBER _____________________


    INSURED'S NAME ____________________________________________________
    ADDRESS ___________________________________________________________
    CITY _______________________________ STATE ___________ ZIP ________


          KEY ALL REQUIRED FIELDS AND PRESS ENTER TO ADD RECORD
```

Each time the "toggle forward" key is pressed, CICS-JUGGLER moves to the next virtual terminal.

If you are positioned at the last virtual terminal (equal to the number of sessions available on this terminal) and press the toggle key you will be positioned back to virtual terminal number one.



```
LKUP,A*

 ABERNATHY, DAWN M.         06378   KANSAS CITY, MO
 ABBOT, KARRON L.           00423   OKLAHOMA CITY, OK
 ABRAHAM, AARON             99332   SEATTLE, WA
 ANDERSON, BOBBIE           26932   NASHVILLE, TN
 ANDERSON, J. L.            00412   NEW YORK, NY
 ARBLE, MARY A.             03500   SAN DIEGO, CA
 ASHLEY, WILLIAM            63860   ARDMORE, OK
 ASNER, JOHN A.             12654   DALLAS, TX
 BARKLEY, RONALD            02564   CHICAGO, IL
 BRADLEY, JACK W.           00163   FT. WORTH, TX
 BRADLEY, JOHN S.           10456   MEMPHIS, TN
 BREKHAN, ALLEN R.          03664   LOS ANGELES, CA
 CACHET, GEORGE C.          44512   DETROIT, MI
 COUCH, WILLIAM E.          09432   BETHANY, OK
 CREED, ROBERT A.           00087   WASHINGTON, DC
 DANELY, JAMES L.           74811   OKLAHOMA CITY, OK
 DOVER, EARL III            36029   LUNAR SURFACE
 DUGAN, MAX L.              98473   NEW YORK, NY
 DUNLAP, SAMANTHA           34801   MELBOURNE, AUST
 DUNN, RALPH K.             09358   SAN DIEGO, CA

PF3=EXIT   PF7=BACKWARD   PF8=FORWARD
```

Now, each time you press the "Toggle Forward" key, control will proceed to the next sequential virtual terminal and re-display it exactly as it was left.

If "Direct Session" keys or the "backward toggle" key have been defined for this terminal, there are additional methods of moving from one virtual terminal to another.

BACKWARD TOGGLE KEY OPERATION

The "backward toggle" key works exactly like the "forward toggle" key in reverse. Each time it is pressed, control moves to the previous sequential terminal until virtual terminal 1 is reached. When pressed again, control moves to the last virtual terminal.

For example, assume four virtual terminals are established and virtual terminal 4 is the current active session. Pressing the backward-toggle key four times would cause a switch from terminal 4 to 3, terminal 3 to 2, terminal 2 to 1, then from terminal 1 back to 4 again.

The User Configuration Display will show if Direct Session or Backward Toggle keys are available on your terminal. If not, you may alter the appropriate field(s) of the User Configuration Display for the keys that you wish (for more information, see *THE USER CONFIGURATION DISPLAY* in section 02 - *OPERATION*). Your MIS contact person or system programmer should be able to make these keys automatically available upon JUGGLER initialization if you want to use them (see THE USER PROFILE TABLE in section 10 - *CUSTOMIZATION*).

DIRECT SESSION KEY OPERATION

Direct Session keys provide a unique PF or PA key for one or all of the virtual terminals that are in use. Pressing one of these keys will transfer control directly to the corresponding virtual terminal without the necessity of moving sequentially through all sessions.

To illustrate, suppose you have four virtual terminals established, PF21 is the Direct Key for session 1, PF22 for session 2, PF23 for session 3 and PF24 for session 4. If you are currently operating a transaction in session 1, pressing PF24 would immediately switch to session 4, without toggling through sessions 2 and 3. Pressing PF22 would switch to session 2, etc.

EXAMPLES OF USE

Example 1: Program development and testing.

One excellent use of **CICS-JUGGLER** is in the area of on-line (or batch) program development. The advantage is seen in setting up three separate sessions during application development and testing. Session One could contain the application program text editor. Session Two could display the data base. Session Three could be where the application is run. **CICS-JUGGLER** eliminates the need for logging in and out of the text editor as changes are made. It also eliminates the need to fetch the application and to position to the appropriate location. During application development you watch Session Three to see the application run while Session Two shows the results. As changes are required you toggle over to Session One to use the text editor.

Example 2: Data Entry Exiting.

Many data-entry applications and packages keep running totals of records entered, amounts, and other data. In order to exit in the middle of such a batch, most of these applications require special commands, such as SAVE, RESTORE, or RESTART-BATCH. With **CICS-JUGGLER** the data-entry operator can move to another application and then return to the entry screen without disturbing the data-entry application in any way.

Example 3: Saving your place in a multi-system environment.

When CICS is only one of several on-line systems on the machine, or when multiple CICS systems are in use, it may be desirable to save the current application status prior to exiting CICS altogether. This can be accomplished with **CICS-JUGGLER**.

To illustrate: Suppose a programmer in an OS system is testing a CICS transaction, and needs to log off of CICS in order to go to TSO to edit the program. If the programmer "toggles" out of the current CICS transaction before signing off of CICS, the application in progress will be saved. After signing back on to CICS, a single keystroke of the "toggle" key will refresh the screen to exactly as it was when the screen was left.

Note that this is one of the features of **CICS-JUGGLER** which can be deactivated by the User Option table (see the discussions on *REQUIRE JUGL,OFF AT LOGOFF*, *REQUIRE CLEAR SESSIONS BEFORE OFF*, *REQUIRE TRANSACTION END BEFORE OFF* and *FORCE PURGE AT SIGNON AND SIGNOFF* in section 10 - *CUSTOMIZATION*).

Summary

There are many other uses for **CICS-JUGGLER** in the CICS environment. In summary it can be said that if you need to invoke a different application than the one you are in and would like to not "lose your place", or if you need to retrieve and display information from several applications at once, that's what **CICS-JUGGLER** is all about.

The more you use **CICS-JUGGLER**, the more uses you will find for it and each use will bring a corresponding increase in productivity.

USING THE CUT AND PASTE FEATURE

Cut and paste, or more properly, copy and paste, is a feature of CICS-JUGGLER which allows you to copy selected data from one application display and insert this data into another application display, or elsewhere in the same display.

Cut and paste has many operational benefits and uses, some of which are:

- Avoid duplicate keying of data by copying redundant information from another application display.

- Capture partial or entire screen displays and insert them into an editor or word processor for text manipulation or printing.

- Propagate fields of data into other fields.

Cut and paste operates by command entry and use of the ENTER key, but the function may optionally be assigned to PF keys in the User Function Key Assignment display.

Copied fields are held in a scratch-pad area until a successive copy is performed. There can be any number of transaction entries between a COPY command and a PASTE command.

Once the PASTE command is completed, the scratch pad area containing the copied field(s) is retained. This allows one to paste the same information on multiple displays without the necessity of performing multiple copies.

COMMANDS USED IN CUT AND PASTE

Two commands are used to accomplish a cut and paste operation. They are COPY ("C") and PASTE ("P").

If you attempt to PASTE without first performing a COPY, the following message will display:

NOTHING TO PASTE

One additional command may be used when the field to be copied is a protected field. This is the 'unprotect' (UNP) command. UNP will set all attributes on the screen to unprotected, thereby allowing the copy command to be entered in any field.

Note that the transaction in the session will not operate following an unprotect command. The next time ENTER or a function key is pressed, the screen will restore with the correct attribute structure and the transaction may continue.

All cut and paste commands are entered by typing over some data of the application display, though after the function is performed, the previous data in the display will not be disturbed. The command must be preceded by the command character. The default command character is a tilde (~), but it may be changed in the customization table.

THE COPY COMMAND

The COPY command is entered in a data field of the application display as the command character followed by the COPY command code. Thus, if the default values have not been changed in the customization table, the command would be ... ~C (tilde, 'C').

For individual field copies, the data copied begins with the character that occupied the space where the command character is keyed, then continues to the end of the field (until the next attribute is found).

For line copies (see *TYPES OF CUT AND PASTE OPERATIONS*), the data copied begins with the first character of the line, or row, where the command was entered. For line copies, it does not matter where in the line the COPY command is placed.

For full-screen copies, the COPY command can be placed anywhere on the screen. The data copied begins with the first character on the screen (row 1, column 1) and continues to the end of the screen.

THE STACK COMMAND

The STACK command may be used to append copied information to the scratch pad area without deleting previously copied data.

The STACK command operates exactly like the COPY command (with the above mentioned exception). The STACK command is entered in a data field of the application display as the command character followed by the STACK command code. Thus, the command would be ... ~S (tilde, 'S').

THE PASTE COMMAND

The PASTE command is entered in a data field of the application display as the command character followed by the PASTE command code. Thus, if the default values have not been changed in the customization table, the command would be ... ~P (tilde, 'P').

For individual field pastes, the data pasted begins at the location where the command character is keyed, then continues to the end of the field (until the next attribute is found). If the data that was copied exceeds the length of the paste field, it is truncated on the right. If the copied data is shorter than the paste field, the remainder of the paste field is padded with spaces.

For line copies (see *TYPES OF CUT AND PASTE OPERATIONS*), the data pasted begins at the location where the command character is keyed, then continues to the end of the line. Consecutive characters of the copied line (starting with the first character) will be pasted into all unprotected fields on the line (unless a field count value was entered with the PASTE command). If the copied line exceeds the length of the paste line, excess characters of the copied line are truncated on the right. Consecutive lines of the copied data, starting with the first character of each copied line, will be pasted into consecutive lines of the screen, each pasted line beginning at the same location on its respective line where the command character was placed on the line containing the PASTE command. If the number of lines copied exceeds the number of lines available to be pasted, excess copied lines are dropped.

For full-screen copies, the data pasted begins at the location where the command character is keyed, then continues to the end of the screen. If the data copied exceeds the paste screen size, excess characters of the copied screen are dropped.

[Note]: For an understanding of individual field copies, line copies and full-screen copies, see *TYPES OF CUT AND PASTE OPERATIONS*, below.

THE UNPROTECT COMMAND

The unprotect command can be entered at any location of the application display. It is entered as the command character followed by 'UNP'. Thus, if the default command character is in use, the command would be ~UNP.

When the UNP command is entered, all field attributes on the screen are set to unprotect. The intensity value remains unchanged. This allow you to enter a COPY command in any data field of the screen, even title and text fields.

Following an unprotect command, the transaction in this session is temporarily suspended. You cannot operate the transaction until you press ENTER or some other function key with no command character on the screen. At that time, the original screen is refreshed with the correct attribute structure, whereupon the transaction may be continued.

TYPES OF CUT AND PASTE OPERATIONS

The COPY and PASTE commands can accomplish any of the following functions:

1). Copy a single field from a display and paste it elsewhere maintaining the same attribute structure.

2). Copy multiple fields from a display and paste them elsewhere maintaining the same attribute structure.

3). Paste the same field into multiple other fields (field propagation).

4). Copy one or more lines (copying all fields on the line) of a display, clear all field attributes and paste the line(s) elsewhere.

5). Copy an entire screen display, maintain the same attribute structure and paste the entire screen over another screen or a blank screen.

6). Copy an entire screen display, clear all attributes and paste the entire screen over another screen or a blank screen.

Each of these functions differ in their usage and operation and it is important to understand each in order to know when to use the correct type of copy and paste to accomplish your purpose.

SINGLE-FIELD CUT/PASTE

To copy a single field, type the COPY command (~C if the default command character is in use) starting at the first position of the field to be copied. This does not have to be the first byte of the field. All data in the field from the location of the command character to the end of the field will be copied and stored in a scratch-pad area.

Now you may toggle to another session or move to another field of the same application display, then type the PASTE command (~P if the default command character is in use) starting at the first position of the field where the data is to be inserted. This does not have to be the first byte of the field. All data in the field from the location of the command character to the end of the field will be replaced by the data of the field that was copied. If the data that was copied exceeds the length of the paste field, it is truncated on the right. If the copied data is shorter than the paste field, the remainder of the paste field is padded with spaces.

When a field is pasted, the MDT (modified data tag) for that field is set on. Thus, when ENTER or a function key is pressed, that data field will transmit to the receiving application, just as if it were keyed by the operator.

MULTIPLE-FIELD CUT/PASTE

To copy multiple fields, type the COPY command starting at the first position of the first field to be copied. This does not have to be the first byte of the field. Now place the command character in all subsequent fields to be copied. It is only necessary to have the COPY command on the first field to be copied. You can enter it is subsequent fields if desired, but just the presence of the command character is sufficient. All data in each field from the location of the command character to the end of the field will be copied and stored in a scratch-pad area.

Now you may toggle to another session or move to another field of the same application display, then type the PASTE command starting at the first position of the first field where the data is to be inserted. This does not have to be the first byte of the field. Now place the command character in all subsequent fields to be pasted. It is only necessary to have the PASTE command on the first field to be pasted. You can enter it is subsequent fields if desired, but just the presence of the command character is sufficient. All data in the field from the location of the command character to the end of the field will be replaced by the data of the corresponding field that was copied. If there are less paste fields than the number of copied fields, the excess copied data is dropped. If there are more paste fields than the number of copied fields, the last copied field is propagated into the excess paste fields.

For each pasted field, the MDT (modified data tag) for that field is set on. Thus, when ENTER or a function key is pressed, those data fields will transmit to the receiving application, just as if they were keyed by the operator.

FIELD PROPAGATION CUT/PASTE

Field propagation is simply duplicating a single field multiple times on the same or another screen. It is accomplished with the PASTE command, after doing a single or multiple field COPY.

If only one field was copied, you can enter a PASTE command in a receiving field, then continue to key the command character in additional fields (any number). When ENTER is pressed, the copied field will be pasted into every location where the PASTE command and the additional command characters were placed.

If multiple fields are copied, field propagation will occur if you key more PASTE locations then the total number of fields copied. In this case, the copied fields are pasted one-for-one until they are exhausted. Then, the final copied field continues to be pasted into the additional locations until all are complete.

LINE CUT/PASTE

To copy multiple lines of a screen display, type the COPY command starting anywhere on the first line to be copied. Follow the COPY command with a comma and a 2-digit number which is the number of lines to be copied. Thus, to copy 9 lines, the command would be entered as ~C,09

Starting with row 1 of this line, all data on every line from this row through the ending row will be copied and stored in a scratch-pad area. All attributes present on each copied line are set to spaces. Thus, the attribute structure of individual fields on the line is lost. The lines become, in effect, pure textual data.

Now you may toggle to another session or move to another field of the same application display, then type the PASTE command starting at the first position on the line where the data is to be inserted.

Starting with the position on the first line where the command character was placed, each copied lines will be pasted, beginning with the first byte of data of the copied line. The paste will continue to the end of the line, pasting into unprotected fields only. Any excess data of the copied line that will not fit on the paste line is dropped. The paste then continues to the next line, starting with the same relative position on this line corresponding to the position of the command character on the first line.

The paste will continue until either the end of the paste screen is reached or the total number of lines that were copied is exhausted.

For each pasted field, the MDT (modified data tag) for that field is set on. Thus, when ENTER or a function key is pressed, those data fields will transmit to the receiving application, just as if they were keyed by the operator.

One variation of the PASTE command, when used with a line copy, is to restrict the number of unprotected fields to be pasted on the receiving lines. By following the PASTE command with a comma and a 1-digit number, the number of fields of each line to be pasted can be specified.

Thus, if the PASTE command is entered as ~P,1

it would mean that only one field in each line is to be pasted and all remaining data of the copied line is to be discarded.

The line copy and paste operation is designed to be used with full-screen editors or word processors. It allows you to copy any sort of data, such as an application screen display, and paste this data into a series of lines of a text editor, thereby capturing the application screen display and converting it into textual data that could be manipulated or printed with the editor. It could also be used to copy multiple lines from one text editor into another.

FULL SCREEN CUT/PASTE

There are two variations of the full-screen cut/paste operation which may be used. The first method will copy an entire screen display and leave the attribute structure intact. The second method will copy the entire screen and clear all attributes to spaces, thereby converting the display to pure textual data.

To copy an entire screen leaving the attribute structure intact, enter the COPY command anywhere on the screen. The command is entered as the normal COPY command followed by a comma and the word 'ALL'. Thus, if the default command character is in force, the command would be entered as ~C,ALL

All data on the screen from row 1, column 1 to the end of the screen, including all field attributes, will be copied and stored in a scratch-pad area.

Now you may toggle to another session or switch to another window (if in window mode), then type the PASTE command anywhere on the receiving screen. The receiving screen could be a blank screen, if desired.

All data of the receiving screen is completely replaced with the copied screen, with no modification made to the attributes of the copied screen.

This function has limited usefulness. It may be possible in some cases to copy a screen into a blank screen, then press ENTER to initiate the same application in another session. This will only work, however, if the screen has the MDT set on the appropriate field(s) to activate the application (such as the transaction code, for CICS transactions). The PASTE operation will not set MDT on for any fields when used in this manner. It is possible, after pasting the screen, to key over any unprotected fields of the display, thereby turning on the MDT, and then press ENTER, causing those fields to be transmitted.

To copy an entire screen converting all attributes to spaces enter the COPY command anywhere on the screen. The command is entered as the normal COPY command followed by a comma and the word 'ALLCLR'. Thus, if the default COPY command code and command character are in force, the command would be entered as ~C,ALLCLR

All data on the screen from row 1, column 1 to the end of the screen will be copied and stored in a scratch-pad area. In the process, all attributes and nulls will be converted to spaces, thereby transforming the screen image into textual data.

Now you may toggle to another session then type the PASTE command anywhere on the receiving screen. The receiving screen could be a blank screen, if desired.

All data of the receiving screen is completely replaced with the copied screen, and the MDT will be set on for the beginning of the screen, row 1 column 1. The pasted screen will now be formatted as one contiguous field from row 1 to the end of the screen.

This function also has limited usefulness. You cannot usually use this function to copy a screen into a text editor, since the control fields of the text editor will be destroyed in the process. You would need to use the line copy for that. It may be possible in some cases to copy a screen into some sort of screen-paint application, such as an application development system, then re-key the necessary control fields and thereby copy that screen into the application.

EXAMPLES OF CUT/PASTE OPERATIONS

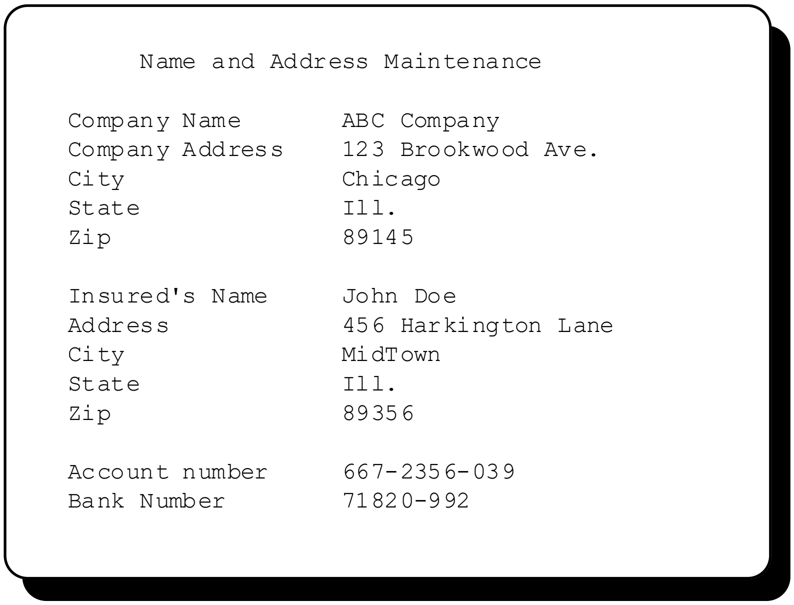
Following are examples of each of the various types of cut and paste operations. Single field copy, multiple field copy, field propagation, and line copy are illustrated.

These examples are intended to illustrate a few meaningful situations where cut and paste can be used to reduce keying time, or capture information that normally could not be captured at all without extensive keying.

As you use the cut and paste feature, you will find many applications where it is useful in your environment.

SINGLE FIELD CUT/PASTE EXAMPLE

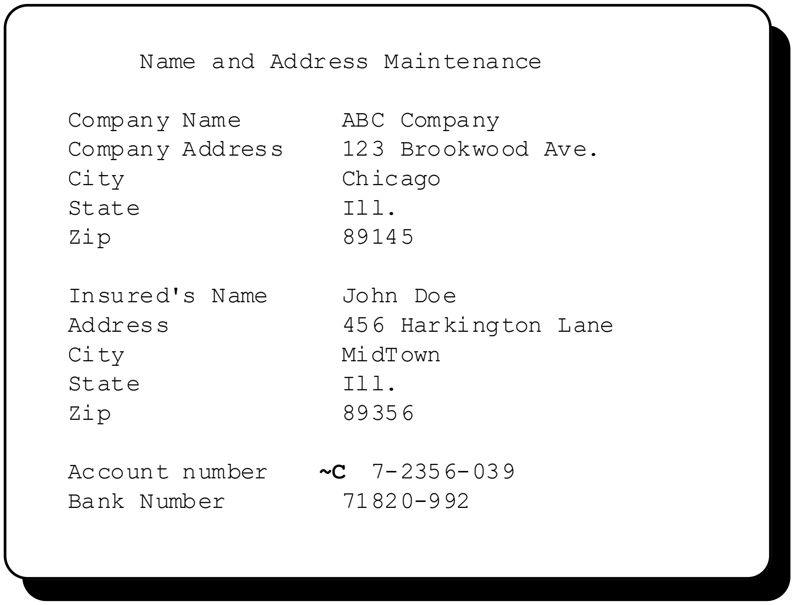
Session 1 contains the following application display:



```
     Name and Address Maintenance

Company Name       ABC Company
Company Address    123 Brookwood Ave.
City               Chicago
State              Ill.
Zip                89145

Insured's Name     John Doe
Address            456 Harkington Lane
City               MidTown
State              Ill.
Zip                89356

Account number     667-2356-039
Bank Number        71820-992
```

We want to copy the account number field and paste it into another application. To do this, we move the cursor to the field and enter the COPY command, preceded by the command character.

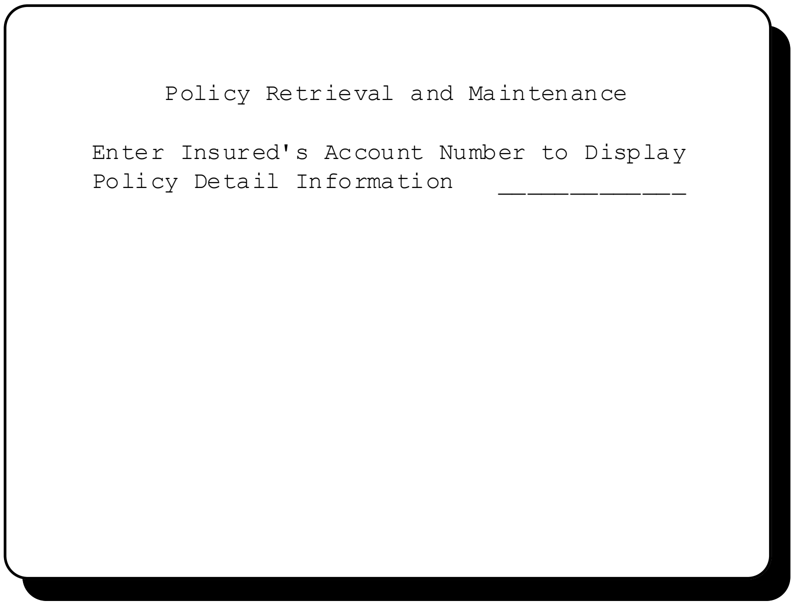
It will appear as follows:



```
     Name and Address Maintenance

Company Name       ABC Company
Company Address    123 Brookwood Ave.
City               Chicago
State              Ill.
Zip                89145

Insured's Name     John Doe
Address            456 Harkington Lane
City               MidTown
State              Ill.
Zip                89356

Account number    ~C  7-2356-039
Bank Number        71820-992
```

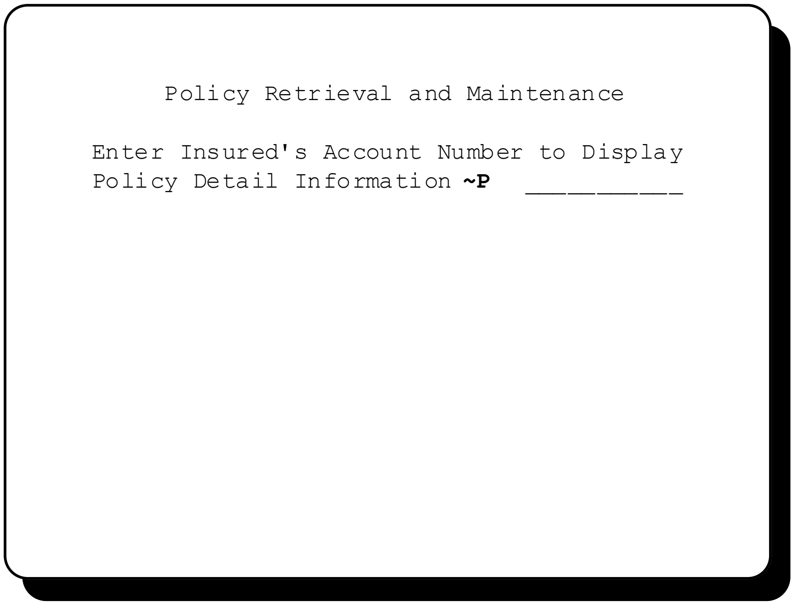
Now we press ENTER, which will re-display the original screen with the message COPY FUNCTION COMPLETED on the bottom row.

Now we toggle to session 2, where the following key-entry application display is waiting:



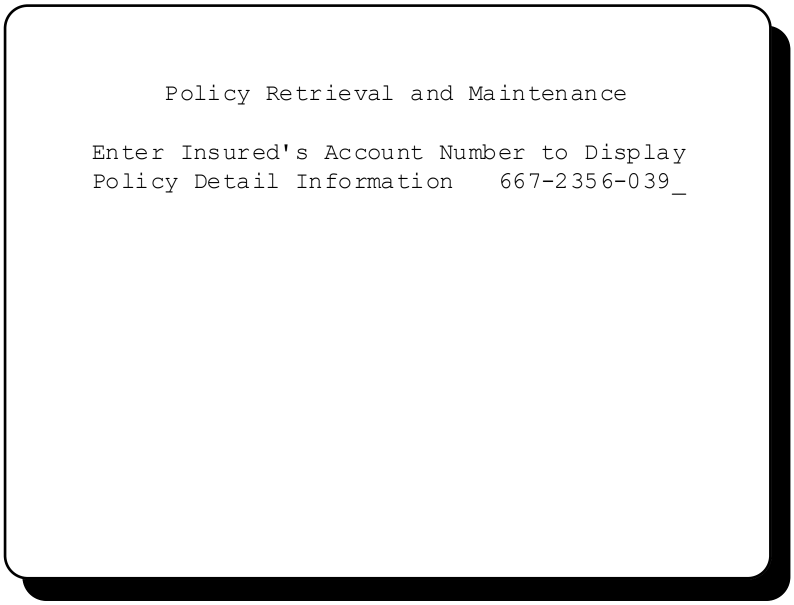
```
     Policy Retrieval and Maintenance

Enter Insured's Account Number to Display
Policy Detail Information   _____________
```

We move to the entry field and key the PASTE command as follows:



```
     Policy Retrieval and Maintenance

Enter Insured's Account Number to Display
Policy Detail Information ~P  ___________
```

Upon pressing ENTER, the copied account number will be inserted into the receiving field, whereupon the operator may press ENTER to transmit the number to the application.

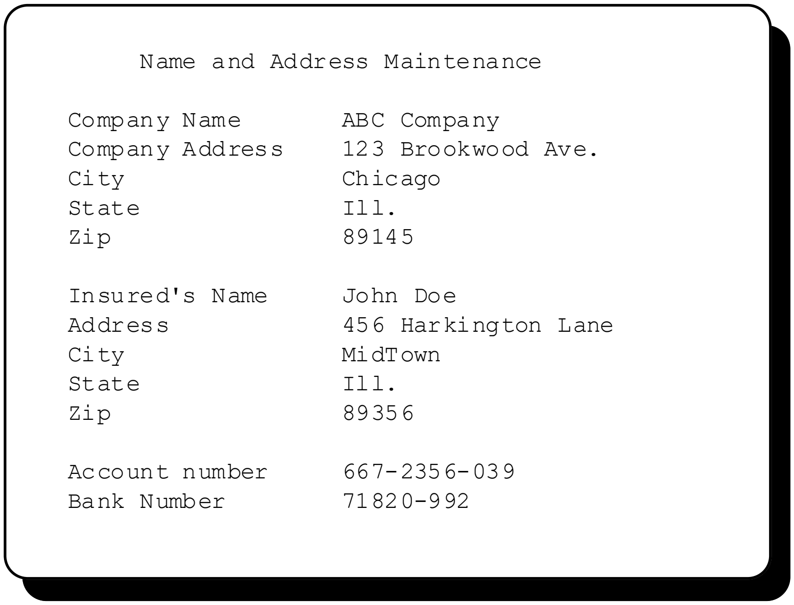


```
     Policy Retrieval and Maintenance

Enter Insured's Account Number to Display
Policy Detail Information   667-2356-039_
```

This example illustrate both the saving of key strokes and the insurance of accuracy when re-keying an account number for a different application.

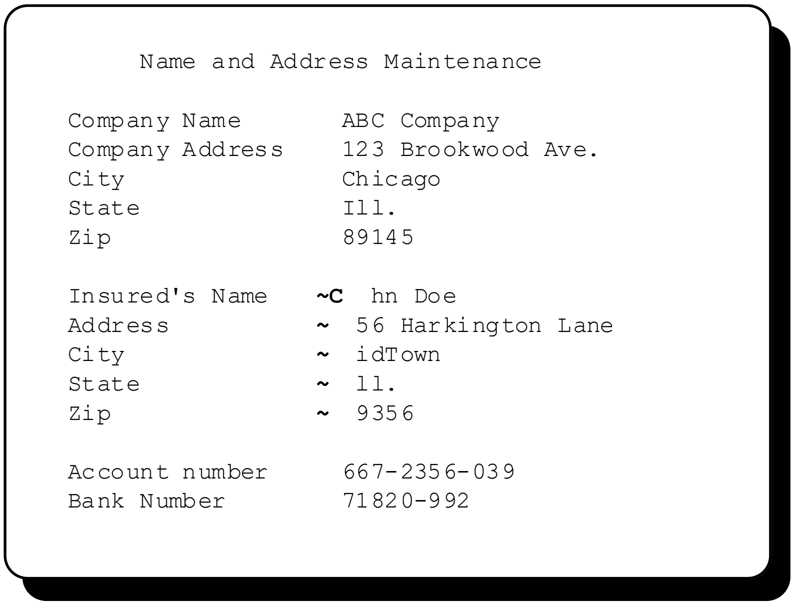
MULTIPLE FIELD CUT/PASTE EXAMPLE

Session 1 contains the following application display:



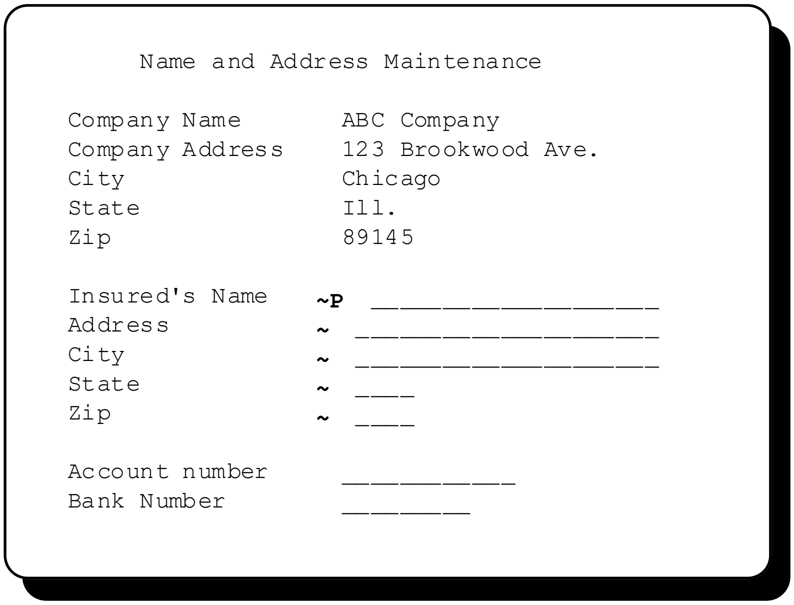
```
     Name and Address Maintenance

Company Name       ABC Company
Company Address    123 Brookwood Ave.
City               Chicago
State              Ill.
Zip                89145

Insured's Name     John Doe
Address            456 Harkington Lane
City               MidTown
State              Ill.
Zip                89356

Account number     667-2356-039
Bank Number        71820-992
```

We want to copy the Insured's name and address and paste it into another input screen of the same application. To do this, we place the COPY command in the insured's name field, then key the command character in the subsequent fields of the address, as follows:



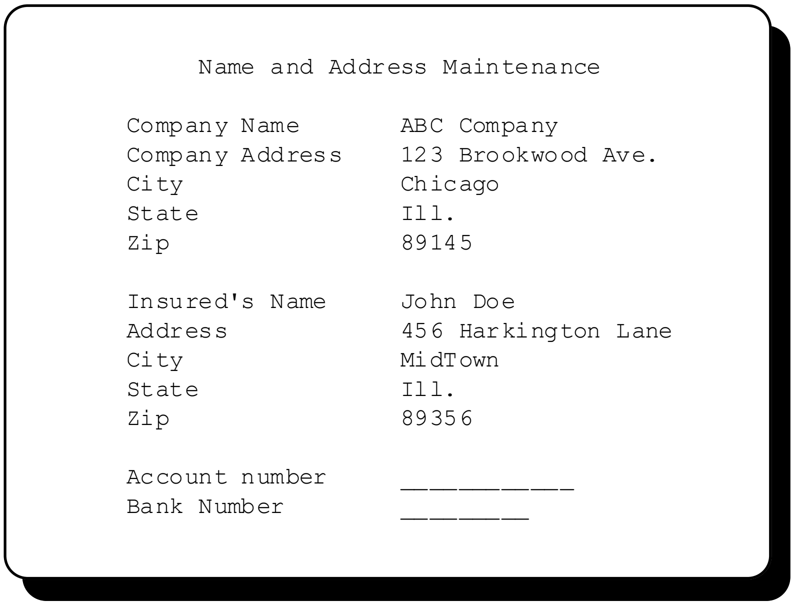
```
     Name and Address Maintenance

Company Name       ABC Company
Company Address    123 Brookwood Ave.
City               Chicago
State              Ill.
Zip                89145

Insured's Name   ~C  hn Doe
Address          ~  56 Harkington Lane
City             ~  idTown
State            ~  ll.
Zip              ~  9356

Account number     667-2356-039
Bank Number        71820-992
```

Upon pressing ENTER, we may now toggle to session 2, activate the application to do a record addition, key the company name/address, the place the PASTE command in the insured's name and the command character in the subsequent address fields, as follows:



```
     Name and Address Maintenance

Company Name       ABC Company
Company Address    123 Brookwood Ave.
City               Chicago
State              Ill.
Zip                89145

Insured's Name   ~P  ____________________
Address          ~  _____________________
City             ~  _____________________
State            ~  ____
Zip              ~  ____

Account number     ____________
Bank Number        _________
```

When ENTER is pressed, the copied fields will be inserted into the corresponding screen fields, as follows:

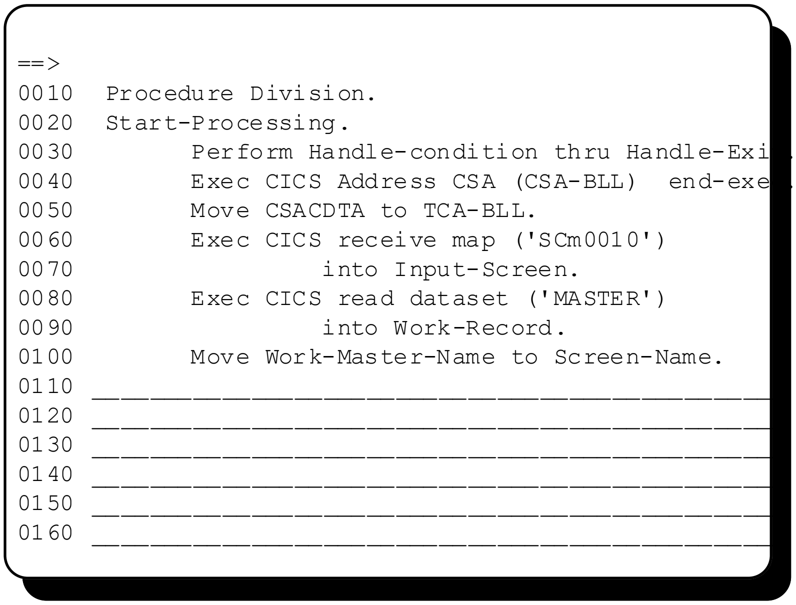


```
     Name and Address Maintenance

Company Name       ABC Company
Company Address    123 Brookwood Ave.
City               Chicago
State              Ill.
Zip                89145

Insured's Name     John Doe
Address            456 Harkington Lane
City               MidTown
State              Ill.
Zip                89356

Account number     ____________
Bank Number        _________
```

Now the remaining two fields can be keyed and ENTER pressed to complete the transaction.

FIELD PROPAGATION CUT/PASTE EXAMPLE

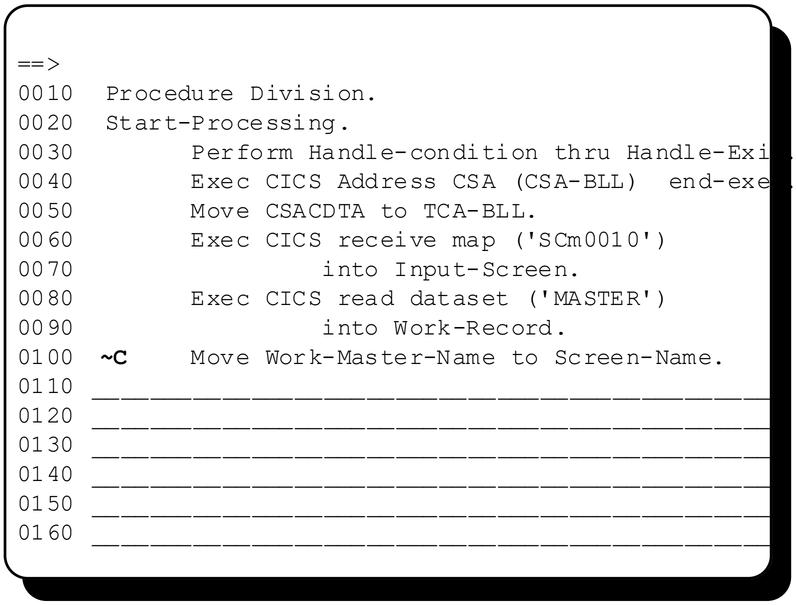
Session 1 contains the following text editor display:



```
==>
0010  Procedure Division.
0020  Start-Processing.
0030        Perform Handle-condition thru Handle-Exi
0040        Exec CICS Address CSA (CSA-BLL)  end-exe
0050        Move CSACDTA to TCA-BLL.
0060        Exec CICS receive map ('SCm0010')
0070                 into Input-Screen.
0080        Exec CICS read dataset ('MASTER')
0090                 into Work-Record.
0100        Move Work-Master-Name to Screen-Name.
0110 _______________________________________________
0120 _______________________________________________
0130 _______________________________________________
0140 _______________________________________________
0150 _______________________________________________
0160 _______________________________________________
```

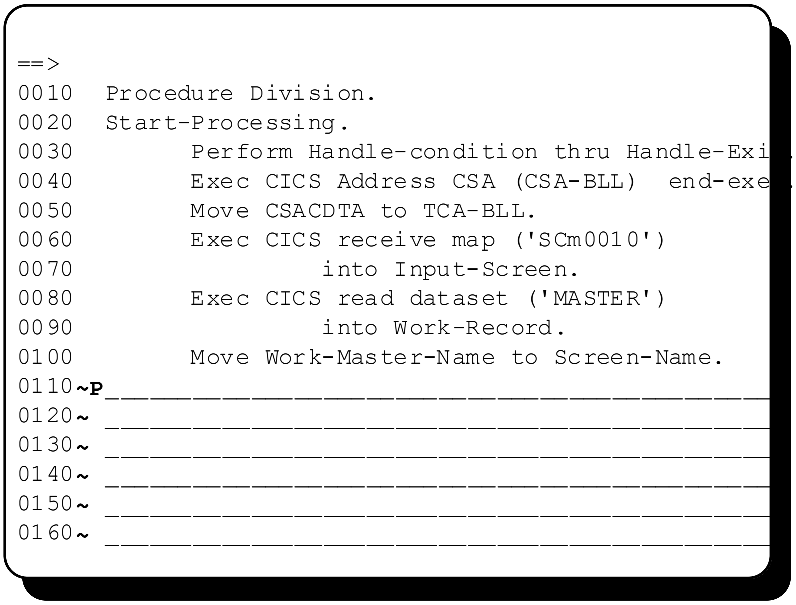
We want to copy the last line (number 0100) and replicate it into the remaining six lines in order to avoid keying "Move Work-Master ..... to Screen" six more times.

To do this, we can do a single field copy on the text of line 0100 as follows:



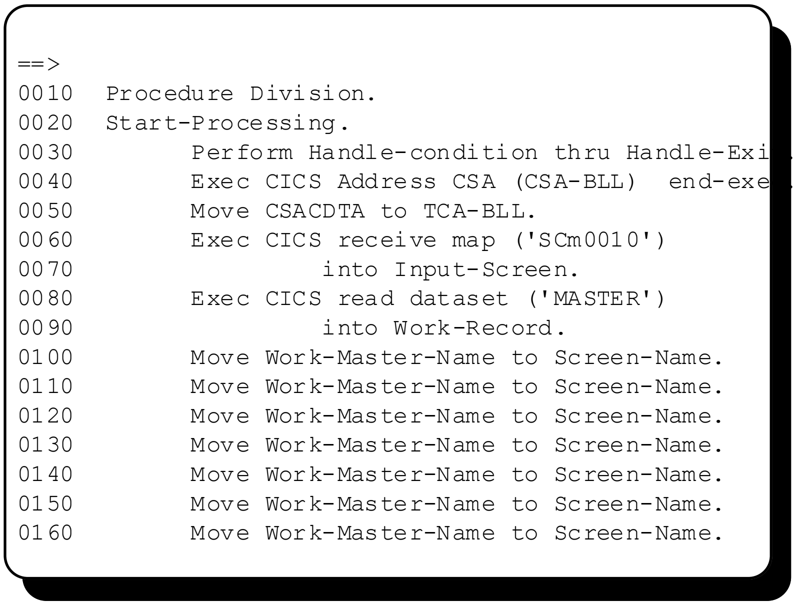
```
==>
0010  Procedure Division.
0020  Start-Processing.
0030        Perform Handle-condition thru Handle-Exi
0040        Exec CICS Address CSA (CSA-BLL)  end-exe
0050        Move CSACDTA to TCA-BLL.
0060        Exec CICS receive map ('SCm0010')
0070                 into Input-Screen.
0080        Exec CICS read dataset ('MASTER')
0090                 into Work-Record.
0100  ~C    Move Work-Master-Name to Screen-Name.
0110 _______________________________________________
0120 _______________________________________________
0130 _______________________________________________
0140 _______________________________________________
0150 _______________________________________________
0160 _______________________________________________
```

Then, after pressing ENTER to complete the copy, place a PASTE command on line 0110 and place the command character on all subsequent lines, as follows:



```
==>
0010  Procedure Division.
0020  Start-Processing.
0030        Perform Handle-condition thru Handle-Exi
0040        Exec CICS Address CSA (CSA-BLL)  end-exe
0050        Move CSACDTA to TCA-BLL.
0060        Exec CICS receive map ('SCm0010')
0070                 into Input-Screen.
0080        Exec CICS read dataset ('MASTER')
0090                 into Work-Record.
0100        Move Work-Master-Name to Screen-Name.
0110~P______________________________________________
0120~ ______________________________________________
0130~ ______________________________________________
0140~ ______________________________________________
0150~ ______________________________________________
0160~ ______________________________________________
```

Now, when ENTER is pressed, line 0100 will be propagated into the remaining six lines.

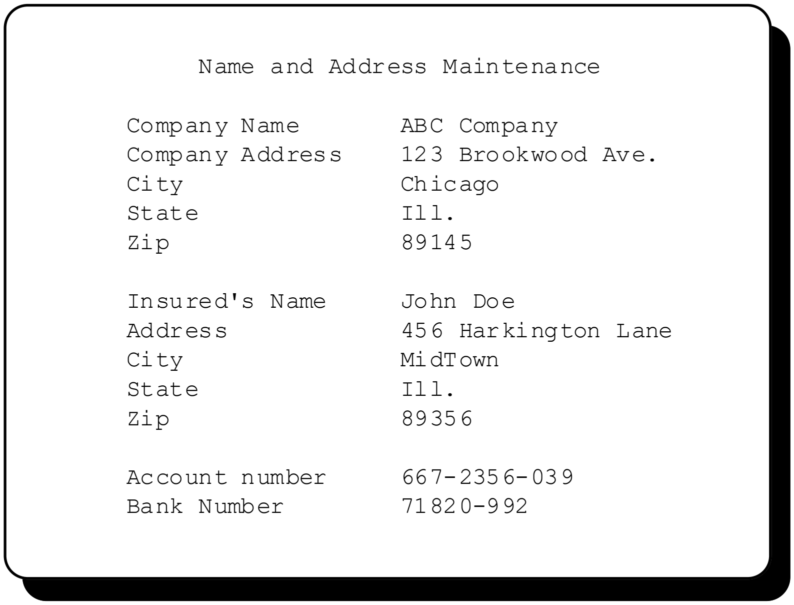


```
==>
Procedure Division.
Start-Processing.
      Perform Handle-condition thru Handle-Exi
      Exec CICS Address CSA (CSA-BLL)  end-exe
      Move CSACDTA to TCA-BLL.
      Exec CICS receive map ('SCm0010')
                into Input-Screen.
      Exec CICS read dataset ('MASTER')
                into Work-Record.
      Move Work-Master-Name to Screen-Name.
      Move Work-Master-Name to Screen-Name.
      Move Work-Master-Name to Screen-Name.
      Move Work-Master-Name to Screen-Name.
      Move Work-Master-Name to Screen-Name.
      Move Work-Master-Name to Screen-Name.
      Move Work-Master-Name to Screen-Name.
```

Now we need only change the final qualifier of each field name to accomplish the six additional move statements.

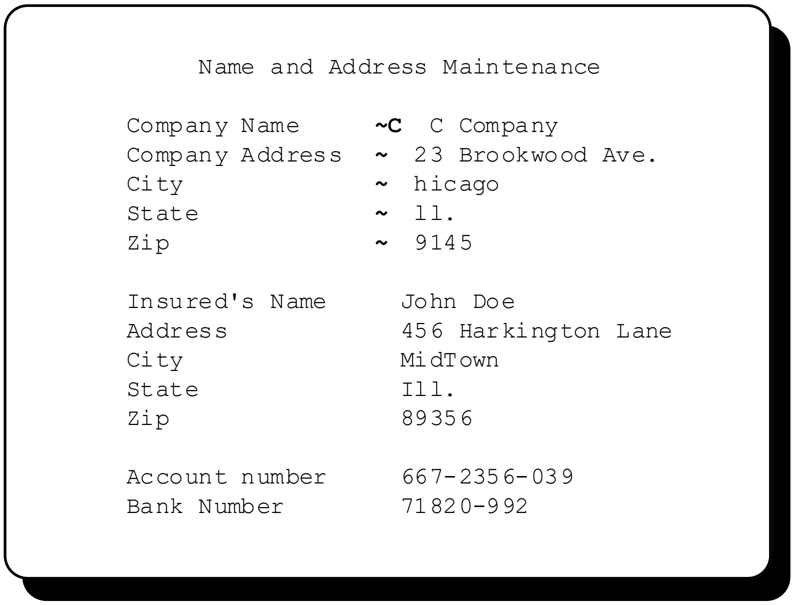
FIELD CUT/PASTE EXAMPLE USING THE STACK COMMAND

Session 1 contains the following application display:



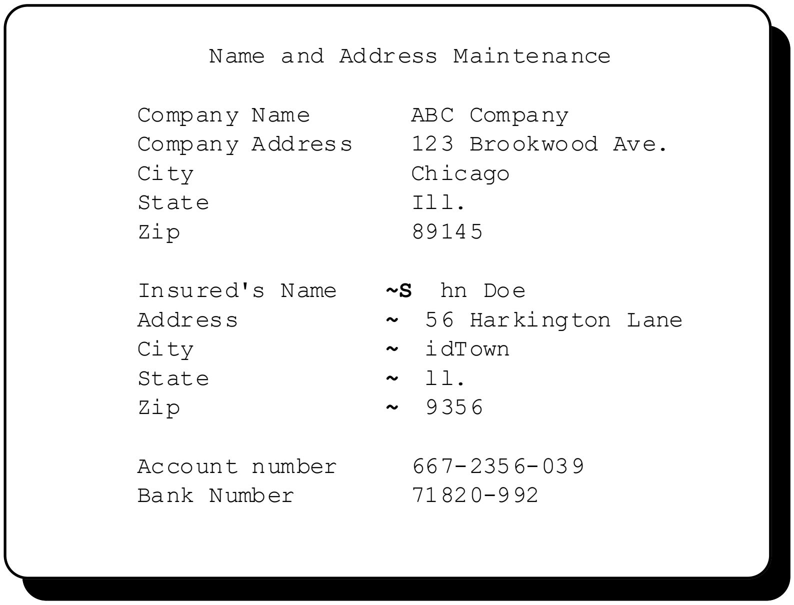
```
             Name and Address Maintenance

        Company Name       ABC Company
        Company Address    123 Brookwood Ave.
        City               Chicago
        State              Ill.
        Zip                89145

        Insured's Name     John Doe
        Address            456 Harkington Lane
        City               MidTown
        State              Ill.
        Zip                89356

        Account number     667-2356-039
        Bank Number        71820-992
```

We want to copy the Company name and address and paste it into another input screen of the same application. To do this, we place the COPY command in the Company name field, then key the command character in the subsequent fields of the address, as follows:



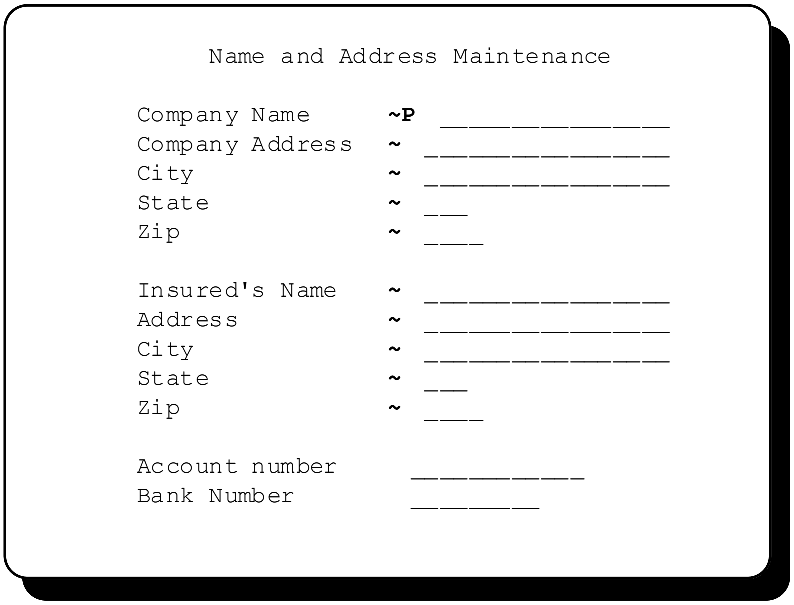
```
     Name and Address Maintenance

Company Name      ~C  C Company
Company Address   ~  23 Brookwood Ave.
City              ~  hicago
State             ~  ll.
Zip               ~  9145

Insured's Name      John Doe
Address             456 Harkington Lane
City                MidTown
State               Ill.
Zip                 89356

Account number      667-2356-039
Bank Number         71820-992
```

Upon pressing ENTER, If we decide that we should also copy the Insured's name and address we could place the STACK command in the Insured's name field, then key the command character in the subsequent fields of the address, as follows:



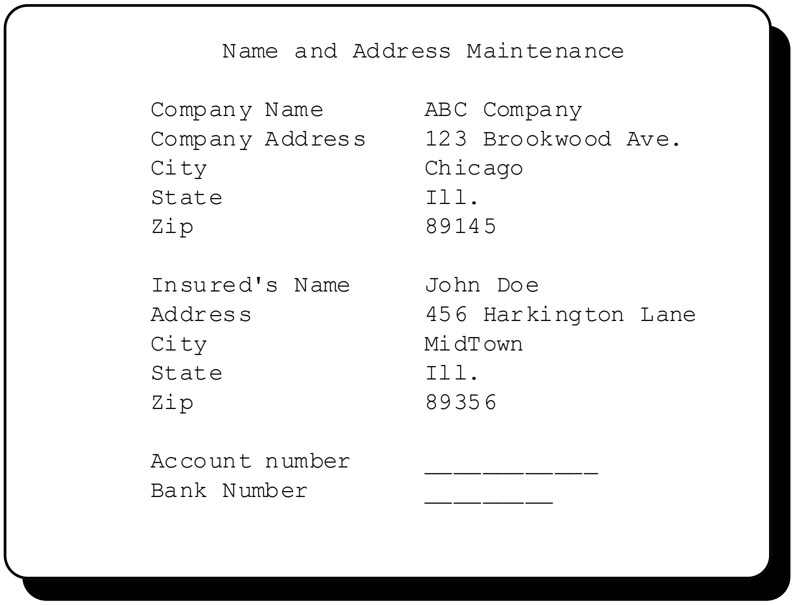
```
     Name and Address Maintenance

Company Name        ABC Company
Company Address     123 Brookwood Ave.
City                Chicago
State               Ill.
Zip                 89145

Insured's Name    ~S  hn Doe
Address           ~  56 Harkington Lane
City              ~  idTown
State             ~  ll.
Zip               ~  9356

Account number      667-2356-039
Bank Number         71820-992
```

We may now toggle to session 2, activate the application to do a record addition and the place the PASTE command in the Company name and the command character in the subsequent address fields, as follows:



```
     Name and Address Maintenance

Company Name      ~P  _______________
Company Address   ~  ________________
City              ~  ________________
State             ~  ___
Zip               ~  ____

Insured's Name    ~  ________________
Address           ~  ________________
City              ~  ________________
State             ~  ___
Zip               ~  ____

Account number      ___________
Bank Number         ________
```

When ENTER is pressed, the copied fields will be inserted into the corresponding screen fields, as follows:

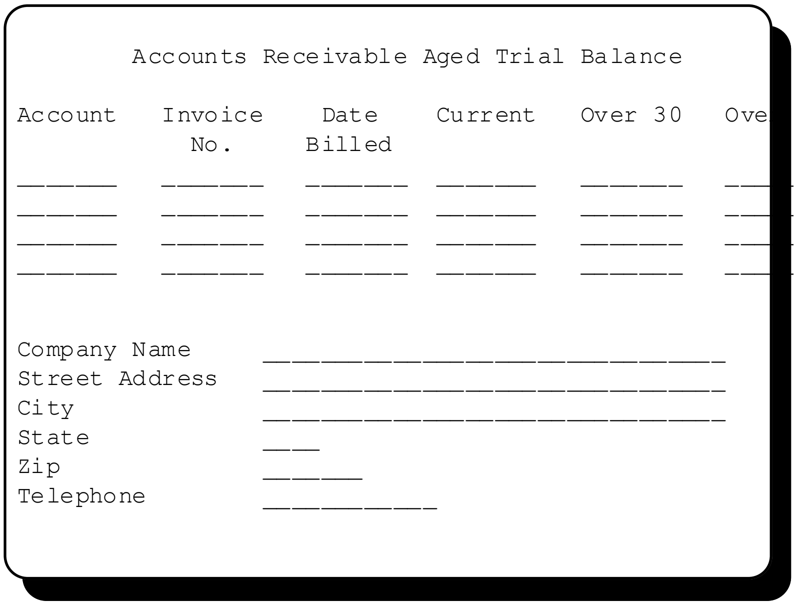


```
     Name and Address Maintenance

Company Name       ABC Company
Company Address    123 Brookwood Ave.
City               Chicago
State              Ill.
Zip                89145

Insured's Name     John Doe
Address            456 Harkington Lane
City               MidTown
State              Ill.
Zip                89356

Account number     ___________
Bank Number        ________
```

Now the remaining two fields can be keyed and ENTER pressed to complete the transaction.

LINE CUT/PASTE EXAMPLE USING THE UNP COMMAND

Session 1 contains the following application display:



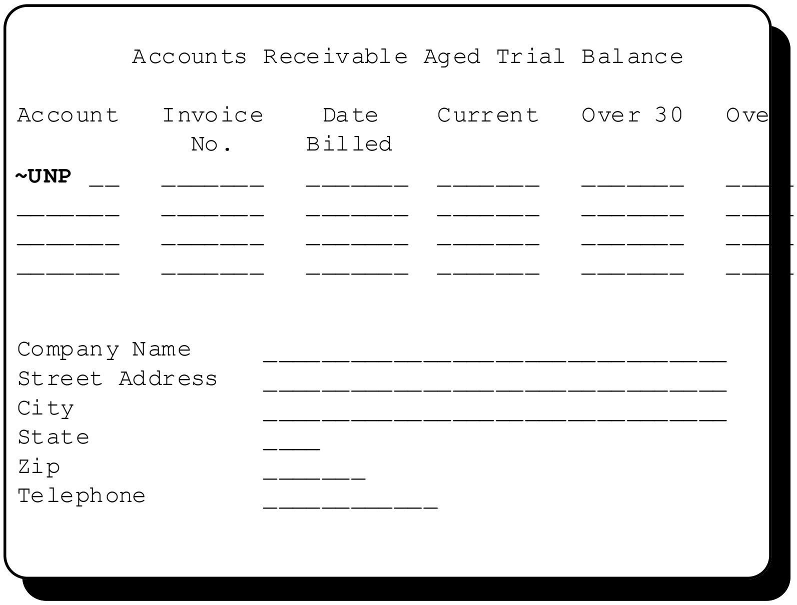
```
       Accounts Receivable Aged Trial Balance

Account   Invoice    Date    Current   Over 30   Ove
            No.     Billed
_______   _______   _______  _______   _______   ____
_______   _______   _______  _______   _______   ____
_______   _______   _______  _______   _______   ____
_______   _______   _______  _______   _______   ____


Company Name      ________________________________
Street Address    ________________________________
City              ________________________________
State             ____
Zip               _______
Telephone         ____________
```

We're working on a documentation project and want to capture this screen image in textual form so that it can be printed as a part of the user manual.

To do this, we need to use a line copy to copy 16 lines, starting with the first line, the header description. With the line copy, you must place the COPY command on the first line to be copied, then follow it with a comma and the number of lines to copy. It does not matter where on the line we place the COPY command.

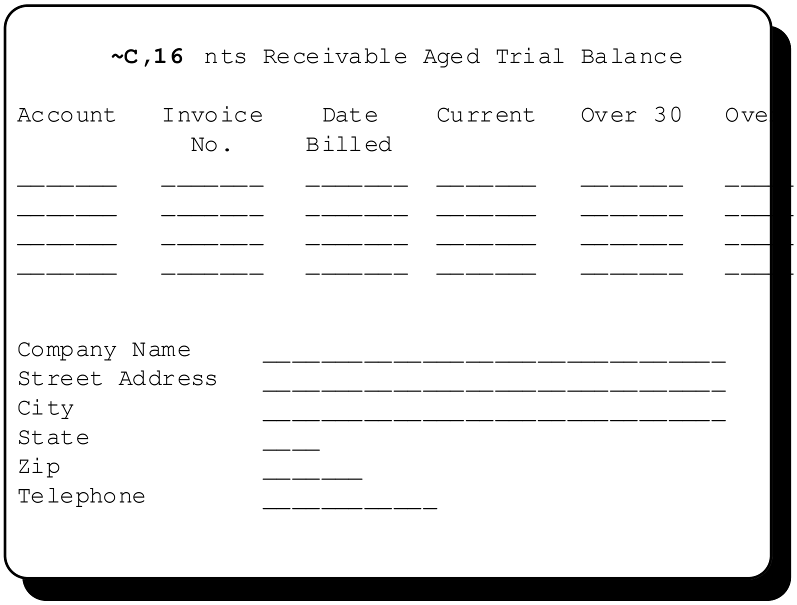
Since the first line is protected and we cannot key the COPY command on the line, we must first use the UNP command to unprotect this screen. This will cause all screen attributes of the display to change to unprotected, meaning that we can key over the title fields of the screen.

We do this by keying the UNP command anywhere on the screen, as follows:



```
       Accounts Receivable Aged Trial Balance

Account   Invoice    Date    Current   Over 30   Ove
            No.     Billed
~UNP __   _______   _______  _______   _______   ____
_______   _______   _______  _______   _______   ____
_______   _______   _______  _______   _______   ____
_______   _______   _______  _______   _______   ____


Company Name      ________________________________
Street Address    ________________________________
City              ________________________________
State             ____
Zip               _______
Telephone         ____________
```

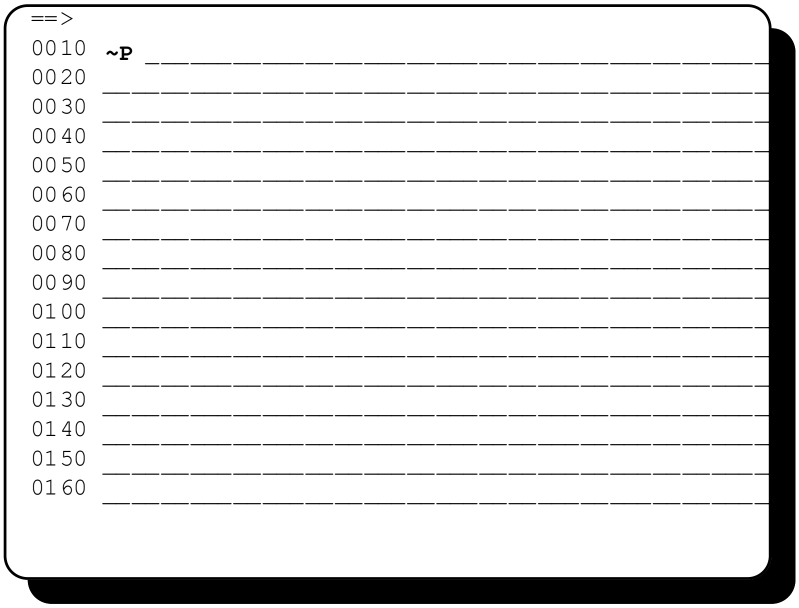
After pressing ENTER, all fields on the screen are unprotected, allowing us to key the line copy command on the header line, as follows:



```
      ~C,16  nts Receivable Aged Trial Balance

Account    Invoice    Date    Current    Over 30    Ove
             No.     Billed
_______    _______   _______  _______    _______    ____
_______    _______   _______  _______    _______    ____
_______    _______   _______  _______    _______    ____
_______    _______   _______  _______    _______    ____


Company Name       ________________________________
Street Address     ________________________________
City               ________________________________
State              ____
Zip                _______
Telephone          ____________
```

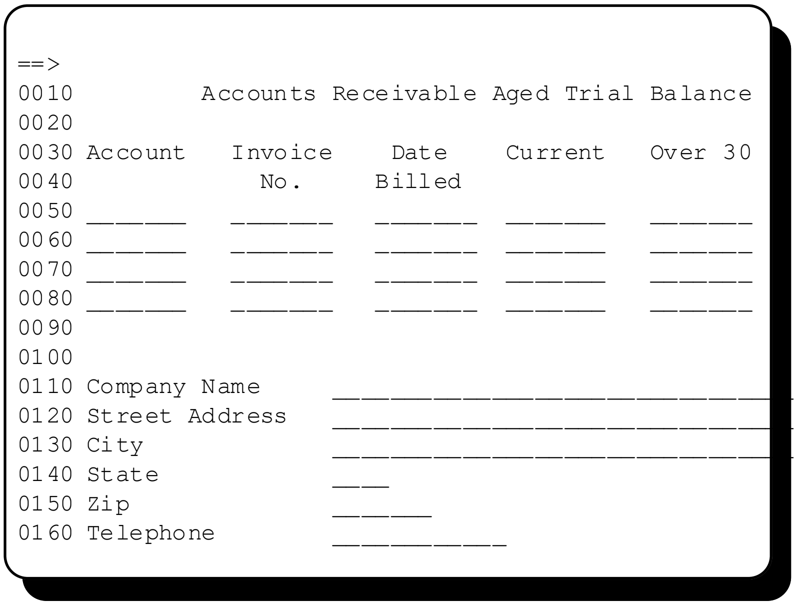
This causes 16 rows to be copied, starting with the first row of data. All attributes and nulls are translated to spaces.

Now we toggle to session 2 where we have a text editor input screen ready to receive the data. We place the PASTE command in the starting position of the first text field, as follows:



```
==>
0010 ~P ____________________________________________
0020 _______________________________________________
0030 _______________________________________________
0040 _______________________________________________
0050 _______________________________________________
0060 _______________________________________________
0070 _______________________________________________
0080 _______________________________________________
0090 _______________________________________________
0100 _______________________________________________
0110 _______________________________________________
0120 _______________________________________________
0130 _______________________________________________
0140 _______________________________________________
0150 _______________________________________________
0160 _______________________________________________
```

After pressing ENTER, the copied lines are pasted into 16 consecutive lines. There will be a few positions truncated off the right, since the text entry field did not start at position 1 of the line. It will appear as follows:



```
==>
0010         Accounts Receivable Aged Trial Balance
0020
0030 Account   Invoice    Date    Current   Over 30
0040            No.     Billed
0050 _______   _______   _______  _______   _______
0060 _______   _______   _______  _______   _______
0070 _______   _______   _______  _______   _______
0080 _______   _______   _______  _______   _______
0090
0100
0110 Company Name     ________________________________
0120 Street Address   ________________________________
0130 City             ________________________________
0140 State            ____
0150 Zip              _______
0160 Telephone        ____________
```

When we press ENTER to the text editor, each pasted line will be transmitted just as if we had keyed all of the data. The display could now be manipulated and printed as desired.

THE HELP-WINDOWS FEATURE

An optionally licensed feature of CICS-JUGGLER is the HELP-WINDOWS feature, which enables users to create on-line help displays for any CICS transaction.

Help displays can be created to display either in full-screen mode or in a pop-up window on the screen. Help can be accessed at the transaction level, the screen level and/or the field level.

The process of creating help is interactive. That is, you initiate the transaction to be documented, display the desired screen, then proceed to add your help text directly to that screen through the use of the AUTO-DEFINE process.

PREPARING TO DEFINE A HELP DISPLAY

Before you can create a help display, you must have two PF keys available. These are the Help Definition (DEFINE) key and the Help Access (HELP) key.

The Define key is used in the creation of help only and is not needed by the end user who views and uses the help. The Help key is used to "test" your help displays as you create them, and by the end user to view them.

The HELP-WINDOWS feature of CICS-JUGGLER is a subset of the powerful HELP-WINDOWS package available from Unicom Systems. It is designed to support on-line help displays only, rather than full documentation systems, manual printing, etc.

Some of the features and functions of the HELP-WINDOWS feature are:

- ASSOCIATE HELP TEXT WITH A TRANSACTION, A SCREEN OF A TRANSACTION, OR A FIELD OF A SCREEN.

- ASSOCIATE HELP TEXT WITH A WORD OR PHRASE WITHIN OTHER HELP TEXT, THEREBY CREATING MULTIPLE LEVELS OF HELP.

- DISPLAY HELP TEXT IN A WINDOW ON THE SCREEN.

- CREATE AND USE HELP MENUS, EITHER IN WINDOW MODE OR FULL-SCREEN MODE, FOR QUICK ACCESS TO VARIOUS SECTIONS OF HELP TEXT.

- "WRAP" THE TEXT TO FIT THE WINDOW, IF IT EXCEEDS THE WINDOW BOUNDARIES.

- SCROLL FORWARD OR BACKWARD IN A HELP DISPLAY.

- NO LIMIT TO THE NUMBER OF PAGES OF HELP TEXT IN A SINGLE HELP DEFINITION.

- USE ROW/COLUMN ADDRESS OR ASSOCIATE THE CURSOR POSITION WITH TEXT ON THE SCREEN TO DEFINE A HELP DISPLAY.

- DEFINE "CROSS-DOCUMENT" FIELDS. FIELDS THAT APPEAR ON MULTIPLE SCREENS OR MULTIPLE TRANSACTIONS, ALWAYS REFERENCING A COMMON SET OF HELP TEXT.

- DEFINE HELP DISPLAYS BY INVOKING THE USER TRANSACTION SCREEN, POSITIONING THE CURSOR AS DESIRED AND PRESSING A "HELP DEFINITION" KEY.

- "CHAIN" SECTIONS OF HELP TEXT TOGETHER IN VARIOUS WAYS TO CREATE MULTIPLE PATHS THROUGH THE DOCUMENTATION.

- CREATE MULTI-WINDOW DISPLAYS. WINDOWS MAY OVERLAP IN ANY CONFIGURATION DESIRED. THERE IS NO LIMIT TO THE NUMBER OF WINDOWS ON ONE SCREEN.

- CREATE AND USE A TABLE OF CONTENTS FOR EASY ACCESS TO MANY HELP TOPICS.

- "INCLUDE" MODULES OF HELP TEXT, LINKING THEM TOGETHER TO FORM A CONTIGUOUS HELP DISPLAY.

- IMPORT TEXT FROM SOME OTHER FILE INTO THE HELP-WINDOWS TEXT FILE. CONTROL STATEMENTS MAKE IT EASY TO DEFINE AND SUB-DEFINE THE IMPORTED DOCUMENT.

DEFINING THE HELP KEY

The choice of which PF key to use for help retrieval is performed at the CICS-JUGGLER User profile, as follows.

1) Enter the JAUX trancode from a clear screen.
2) Select User Profiles.
3) Locate the profile or profiles to be used, or make the same change to all profiles.
4) Enter the PF key mnemonic for the key to be used in the HELP KEY field.

[Note] For CUA compatibility, PF1 should be used as the Help key, which will not conflict with the on-line help available in CICS-JUGGLER.

DEFINING THE HELP-DEFINITION (DEFINE) KEY

The choice of which PF key to use for help definition can also be performed at the CICS-JUGGLER User profile, if desired.

1) Enter the JAUX trancode from a clear screen.
2) Select User Profiles.
3) Locate the profile or profiles to be used
4) Select the KEYS pull-down menu, then select Display Function Keys.
5) At the Function Keys display, enter the DEFINE command beside the PF key you wish to use.
6) Exit the Keys display with PF3.

[Note] This method will permanently assign this PF key as the Define key whenever this profile is in use.

AN ALTERNATE METHOD

You can temporarily assign a Define key with the following command: JHLP,DEFINE,PFxx
where PFxx is the mnemonic of the PF key to be used for the Define key (PF1, PF2 ... PF24). With this method, it does not matter what profile is in use. The chosen Define key will remain in force on this terminal until a JUGL,OFF is performed or JHLP,DEFINE=OFF.

ENTERING AUTO-DEFINE MODE

Before you can create a help definition, you must establish your terminal as authorized to use Auto-Define. This is performed with the command: HLP,DEFINE
As previously stated, you can optionally follow the command with the mnemonic of the PF key to be used as the Define key. If you are authorized to use the JHLP transaction code, you can perform Auto-Define.

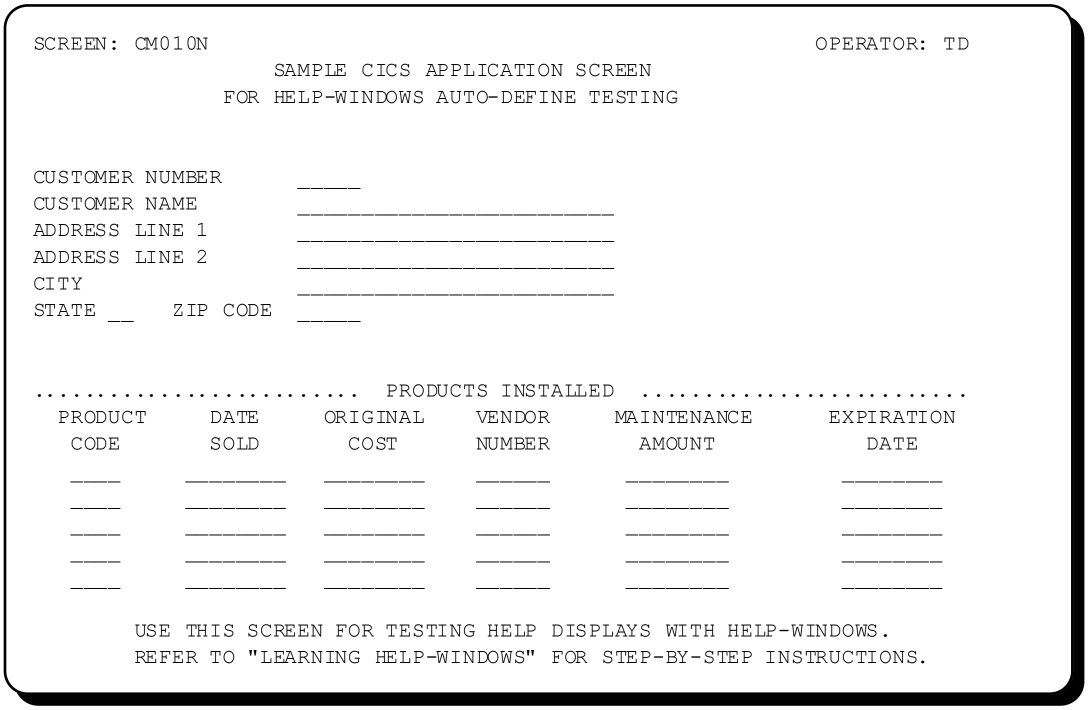
CREATING A FIELD LEVEL HELP DISPLAY, A QUICK OVERVIEW

You are now ready to create and test an on-line help display. For our example, we will use a user transaction screen called HDMO. This display transaction is distributed with the CICS-JUGGLER product. It is intended to be a practice vehicle in learning to create help definitions.

STEP 1. INVOKING THE USER TRANSACTION

Enter a transaction code which will display a screen that you would like to document. For our purposes in this illustration, we will use the HDMO transaction.

Upon keying HDMO and pressing ENTER, the HDMO display will appear as follows:



```
SCREEN: CM010N                                                OPERATOR: TD
                   SAMPLE CICS APPLICATION SCREEN
               FOR HELP-WINDOWS AUTO-DEFINE TESTING


CUSTOMER NUMBER      _____
CUSTOMER NAME        _________________________
ADDRESS LINE 1       _________________________
ADDRESS LINE 2       _________________________
CITY                 _________________________
STATE __   ZIP CODE  _____


..........................  PRODUCTS INSTALLED  ..........................
  PRODUCT     DATE     ORIGINAL    VENDOR     MAINTENANCE      EXPIRATION
   CODE       SOLD       COST      NUMBER       AMOUNT            DATE

   ____      ________  ________    ______      ________         ________
   ____      ________  ________    ______      ________         ________
   ____      ________  ________    ______      ________         ________
   ____      ________  ________    ______      ________         ________
   ____      ________  ________    ______      ________         ________

        USE THIS SCREEN FOR TESTING HELP DISPLAYS WITH HELP-WINDOWS.
        REFER TO "LEARNING HELP-WINDOWS" FOR STEP-BY-STEP INSTRUCTIONS.
```

STEP 2. POSITIONING THE CURSOR

Now position the cursor to the desired field on the screen where help is to be made available. For this exercise, let's use the CUSTOMER NAME field.

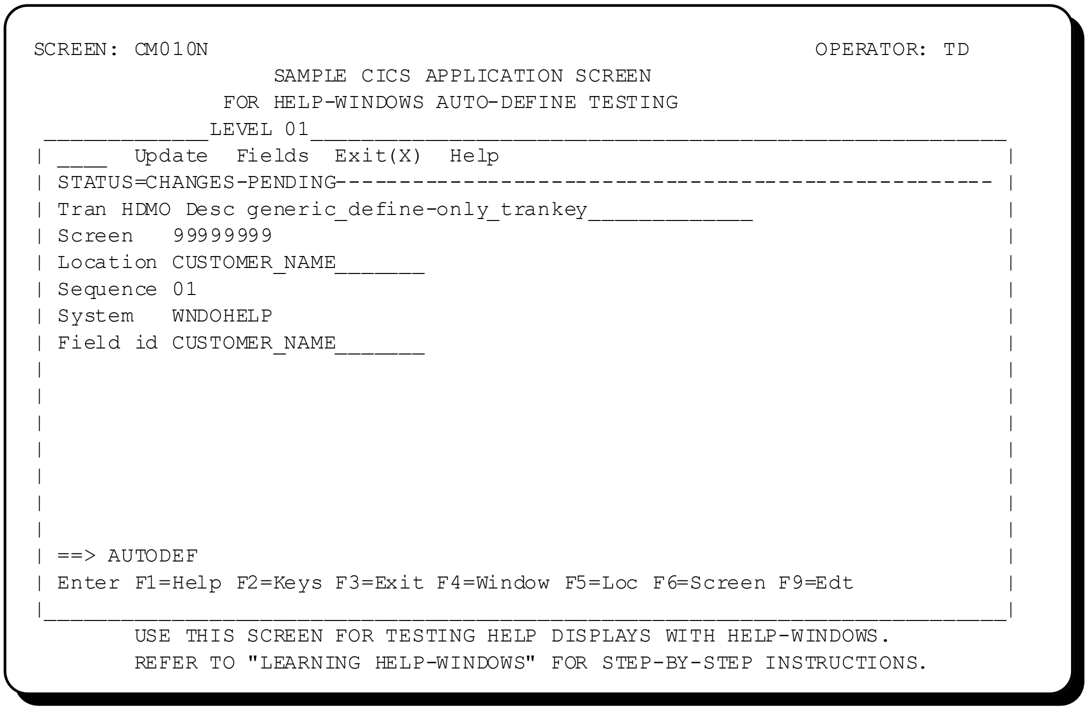
There are two methods of correlating help to a field. One is using absolute row/column cursor position. For this method, you should position the cursor to the first position of the actual data entry field.

However, for this example, we will correlate a literal on the screen to the help text that is to display for this field. For this method, position the cursor to the "C" in CUSTOMER NAME.

STEP 3. PRESS THE HELP-DEFINITION HOT KEY

Now, without moving the cursor, press the PF or PA key that you designated as your "Define" key.

Upon pressing the help definition key, the following window will appear on the screen:



```
SCREEN: CM010N                                              OPERATOR: TD
                  SAMPLE CICS APPLICATION SCREEN
              FOR HELP-WINDOWS AUTO-DEFINE TESTING
 _____________LEVEL 01_____________________________________________________
| ____  Update  Fields  Exit(X)  Help                                      |
| STATUS=CHANGES-PENDING-------------------------------------------------- |
| Tran HDMO Desc generic_define-only_trankey_____________                  |
| Screen   99999999                                                        |
| Location CUSTOMER_NAME_______                                            |
| Sequence 01                                                              |
| System   WNDOHELP                                                        |
| Field id CUSTOMER_NAME_______                                            |
|                                                                          |
|                                                                          |
|                                                                          |
|                                                                          |
|                                                                          |
|                                                                          |
|                                                                          |
| ==> AUTODEF                                                              |
| Enter F1=Help F2=Keys F3=Exit F4=Window F5=Loc F6=Screen F9=Edt          |
|__________________________________________________________________________|
       USE THIS SCREEN FOR TESTING HELP DISPLAYS WITH HELP-WINDOWS.
       REFER TO "LEARNING HELP-WINDOWS" FOR STEP-BY-STEP INSTRUCTIONS.
```

Explanation of results:

The size and position of the window is determined from the help definition that was last created for this transaction, if any. Otherwise it is determined from the distributed "Model" definition.

The STATUS information is provided to indicate the current status of this transaction key record. NEW RECORD indicates that a new help definition is being created and has not yet been added to the file.

The TRANSACTION is filled in with HDMO, the transaction in progress.

The SCREEN ID is all nines, indicating that a unique screen identification is currently not defined.

LOCATION contains CUSTOMER NAME. If we were defining absolute row/column cursor position help, this field would contain 07023, which is row 7 column 23.

The SEQUENCE field contains 01 since this is the first transaction key record to be defined at this position of this screen.

The SYSTEM field contains JUGLHELP, which is the default system ID. System names are used merely for grouping different help subjects together. This is explained in more detail under *THE AUTODEF WINDOW*.

The FIELD ID contains CUSTOMER NAME. This field is used as a cross document name. Cross documents are explained later under *THE AUTODEF WINDOW.*

The ==> in the window is the window command area. It currently contains the AUTODEF command, which must be there in order to respond to the display using the PF key prompts on the bottom row.

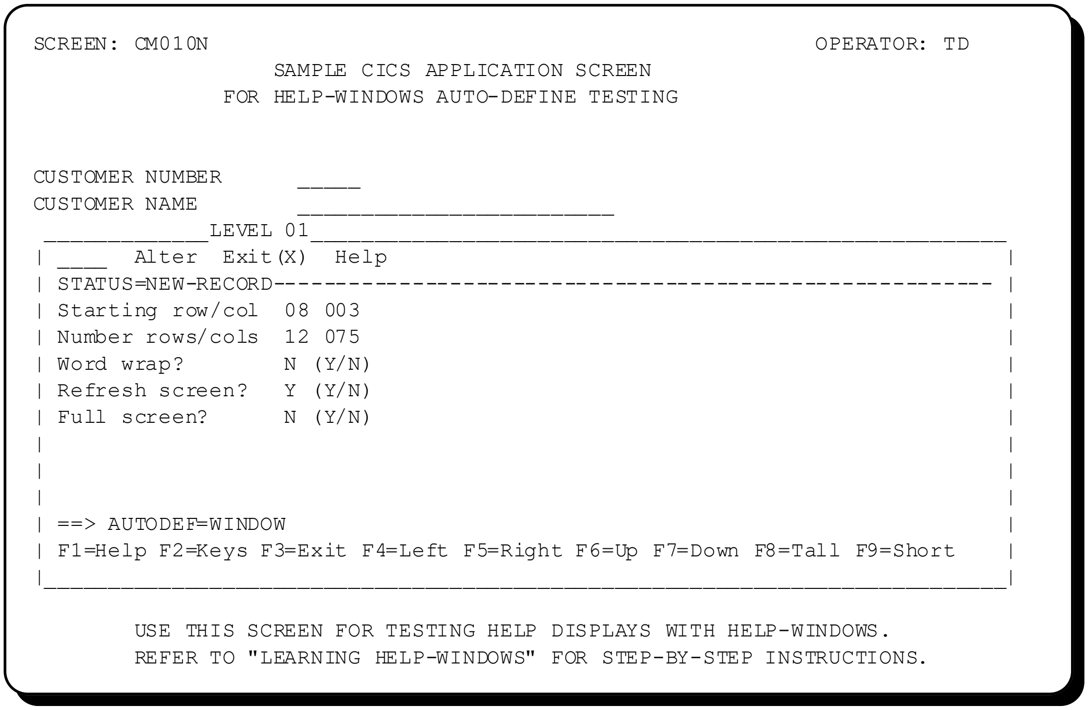
The bottom row of the window contains the directional options (key prompts) which are available at this stage of auto-define.

STEP 4. CHANGING THE WINDOW

Since we're going to create a field help definition, we need to make the window smaller and place it so that the CUSTOMER NAME field can be seen.

Press PF4 to enter change-window mode. Then, either use the PF keys, window commands or direct entry of row/column information to size the window like you want it.

By setting the STARTING ROW/COL to '12 003', NUMBER OF ROWS to '08' and NUMBER OF COLUMNS to '075' we will have a window which appears as follows:

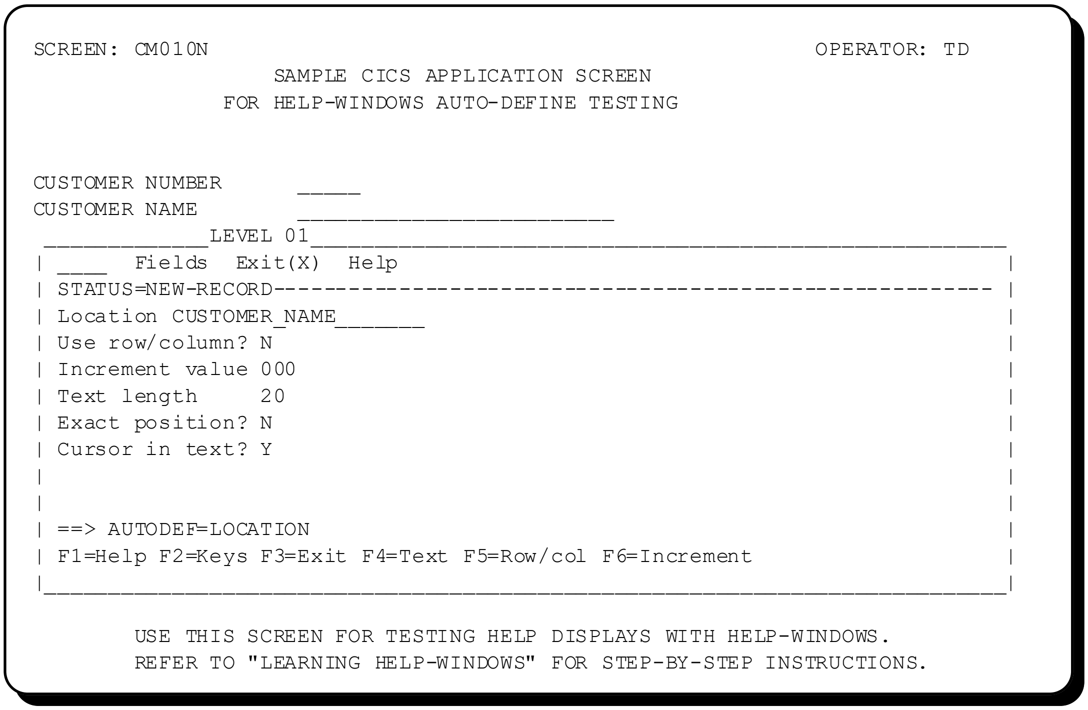


```
SCREEN: CM010N                                              OPERATOR: TD
                   SAMPLE CICS APPLICATION SCREEN
               FOR HELP-WINDOWS AUTO-DEFINE TESTING


CUSTOMER NUMBER      _____
CUSTOMER NAME        _________________________
 _____________LEVEL 01___________________________________________________
| ____  Alter  Exit(X)  Help                                             |
| STATUS=NEW-RECORD----------------------------------------------------- |
| Starting row/col  08 003                                               |
| Number rows/cols  12 075                                               |
| Word wrap?        N (Y/N)                                              |
| Refresh screen?   Y (Y/N)                                              |
| Full screen?      N (Y/N)                                              |
|                                                                        |
|                                                                        |
|                                                                        |
| ==> AUTODEF=WINDOW                                                     |
| F1=Help F2=Keys F3=Exit F4=Left F5=Right F6=Up F7=Down F8=Tall F9=Short |
|________________________________________________________________________|

        USE THIS SCREEN FOR TESTING HELP DISPLAYS WITH HELP-WINDOWS.
        REFER TO "LEARNING HELP-WINDOWS" FOR STEP-BY-STEP INSTRUCTIONS.
```

Now, press PF3 until the main AUTODEF display reappears.

STEP 5. IDENTIFY THE HELP FIELD

Earlier we pressed the Define key while the cursor was positioned to the field descriptor. However, for help access, we will want the operator to be able to press the Help Access key while the cursor is in the actual data entry field. Now we need to specify the offset from CUSTOMER NAME to the actual data entry field. For this press PF5, the Location Window will appear as follows:



```
SCREEN: CM010N                                               OPERATOR: TD
                   SAMPLE CICS APPLICATION SCREEN
               FOR HELP-WINDOWS AUTO-DEFINE TESTING


CUSTOMER NUMBER      _____
CUSTOMER NAME        _________________________
 ____________LEVEL 01_______________________________________________________
| ____  Fields  Exit(X)  Help                                               |
| STATUS=NEW-RECORD-------------------------------------------------------- |
| Location CUSTOMER_NAME_______                                             |
| Use row/column? N                                                         |
| Increment value 000                                                       |
| Text length     20                                                        |
| Exact position? N                                                         |
| Cursor in text? Y                                                         |
|                                                                           |
|                                                                           |
| ==> AUTODEF=LOCATION                                                      |
| F1=Help F2=Keys F3=Exit F4=Text F5=Row/col F6=Increment                   |
|___________________________________________________________________________|

        USE THIS SCREEN FOR TESTING HELP DISPLAYS WITH HELP-WINDOWS.
        REFER TO "LEARNING HELP-WINDOWS" FOR STEP-BY-STEP INSTRUCTIONS.
```

Now position the cursor to the first position of the actual data entry field and press PF6 (Increment), which will automatically compute the offset from the data entry field to the associated literal on the screen (the position of the cursor when the Define key was pressed).

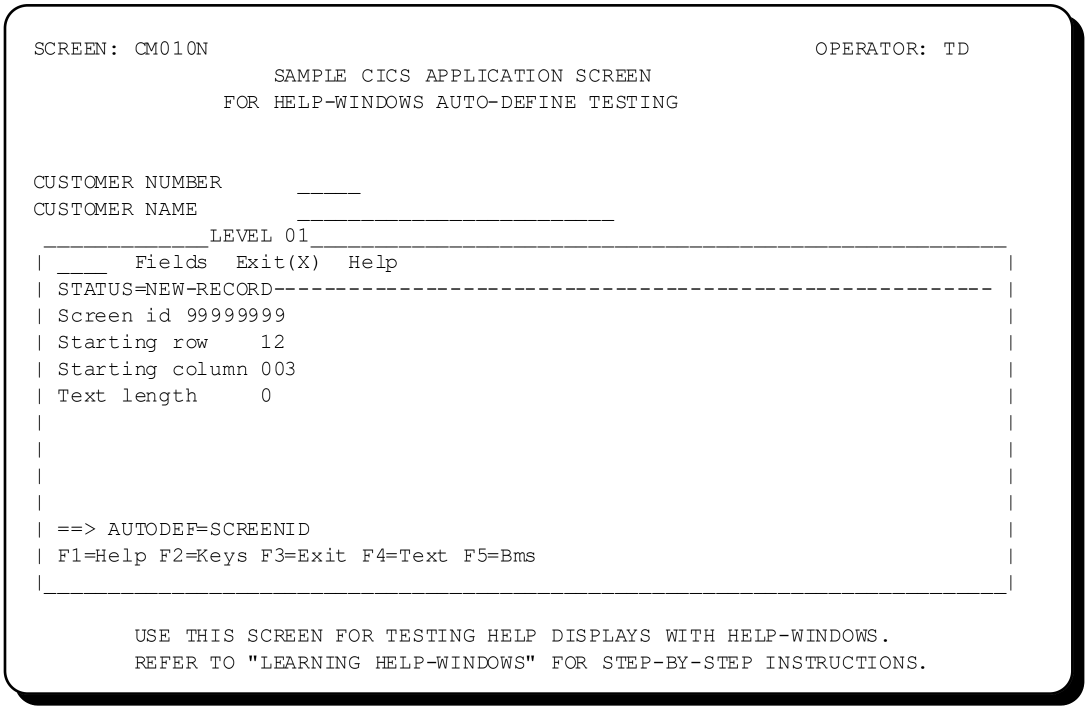
Upon pressing PF6, the value '21-' should appear in the INCREMENT VALUE field. This indicates that the literal 'CUSTOMER NAME' appears 21 positions prior to the cursor location.

Note that if EXACT POSITION is left as 'N', the cursor can be anywhere in the customer name data field when the Help key is pressed, and still associate properly with the CUSTOMER NAME literal.

Now, press PF3 to redisplay the main AUTODEF window.

STEP 6. IDENTIFY THE SCREEN

To ensure that this field level help is not accessible on a different screen of this transaction, to which this help text may not be exactly accurate, you can specify a screen identifier. To do this press PF6 from the main AUTODEF window. The Screen Identifier window will appear:



```
SCREEN: CM010N                                              OPERATOR: TD
                   SAMPLE CICS APPLICATION SCREEN
               FOR HELP-WINDOWS AUTO-DEFINE TESTING


CUSTOMER NUMBER      _____
CUSTOMER NAME        _________________________
 _____________LEVEL 01___________________________________________________
| ____  Fields  Exit(X)  Help                                            |
| STATUS=NEW-RECORD----------------------------------------------------- |
| Screen id 99999999                                                     |
| Starting row    12                                                     |
| Starting column 003                                                    |
| Text length     0                                                      |
|                                                                        |
|                                                                        |
|                                                                        |
|                                                                        |
| ==> AUTODEF=SCREENID                                                   |
| F1=Help F2=Keys F3=Exit F4=Text F5=Bms                                 |
|________________________________________________________________________|

        USE THIS SCREEN FOR TESTING HELP DISPLAYS WITH HELP-WINDOWS.
        REFER TO "LEARNING HELP-WINDOWS" FOR STEP-BY-STEP INSTRUCTIONS.
```

Now locate a literal on the screen that is specific to this screen at the position that it occurs. In this example we have an actual screen ID at row 1, column 10. Position the cursor to this literal and press PF4. This will paste the literal in the SCREEN ID field and insert the position of the literal in the appropriate fields.

Now, press PF3 to redisplay the main AUTODEF window.

STEP 7. ENTER THE HELP TEXT

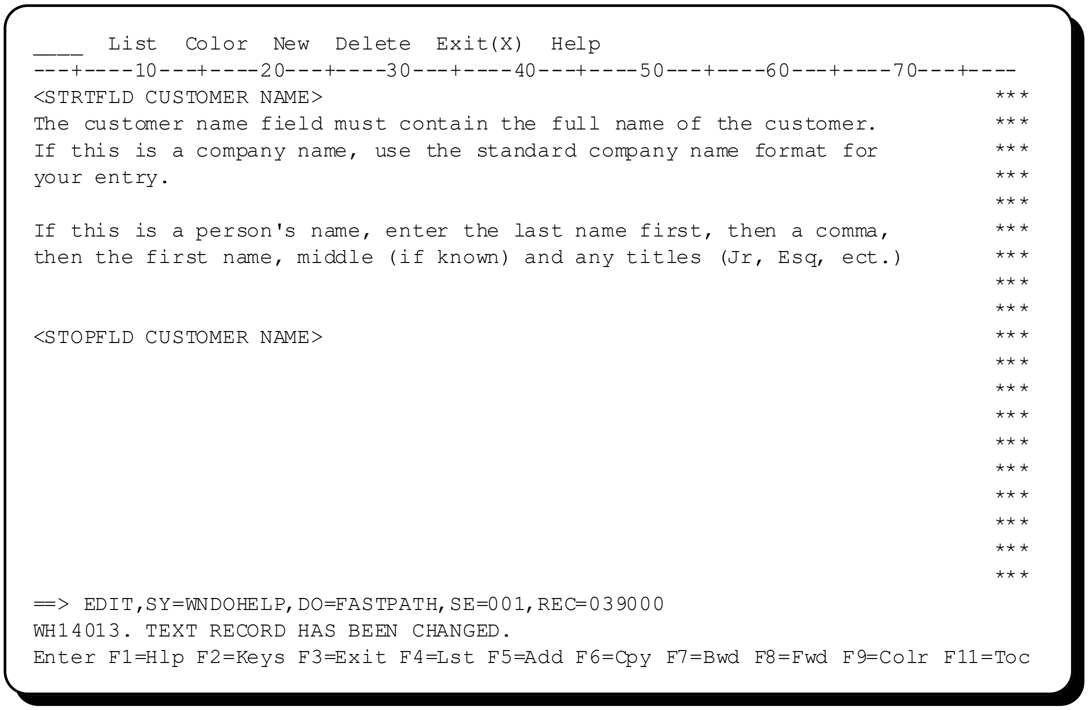
Now we need to key the text that is to display when the Help Access key is pressed. For this press PF9 (EDIT). A full screen text editor will display, with the following statements:

<STRTFLD CUSTOMER NAME>

<STOPFLD CUSTOMER NAME>

These statements define the beginning and end of the help text that is to display for this field. All help text must fall between these statements. The initial number of lines between these two statements is the number of lines that will display in the currently defined window size; however, the STOPFLD statement may be moved to accommodate more help text.

When we have keyed all of the text that we want, it might appear as follows:



```
____  List  Color  New  Delete  Exit(X)  Help
---+----10---+----20---+----30---+----40---+----50---+----60---+----70---+----
<STRTFLD CUSTOMER NAME>                                                      ***
The customer name field must contain the full name of the customer.          ***
If this is a company name, use the standard company name format for          ***
your entry.                                                                  ***
                                                                             ***
If this is a person's name, enter the last name first, then a comma,         ***
then the first name, middle (if known) and any titles (Jr, Esq, ect.)        ***
                                                                             ***
                                                                             ***
<STOPFLD CUSTOMER NAME>                                                      ***
                                                                             ***
                                                                             ***
                                                                             ***
                                                                             ***
                                                                             ***
                                                                             ***
                                                                             ***
                                                                             ***
                                                                             ***
==> EDIT,SY=WNDOHELP,DO=FASTPATH,SE=001,REC=039000
WH14013. TEXT RECORD HAS BEEN CHANGED.
Enter F1=Hlp F2=Keys F3=Exit F4=Lst F5=Add F6=Cpy F7=Bwd F8=Fwd F9=Colr F11=Toc
```

Let's assume that this is all we want to say for this help display and it will fit in the window size displayed. Upon completion of keying the text, we press ENTER. This causes the transaction key record, with the text that we keyed, to be added to the file.

Now press PF3 to exit the text editor and return to the main AUTODEF display. We have now completed the creation of a field-level help display.

STEP 8. EXIT AUTO-DEFINE MODE

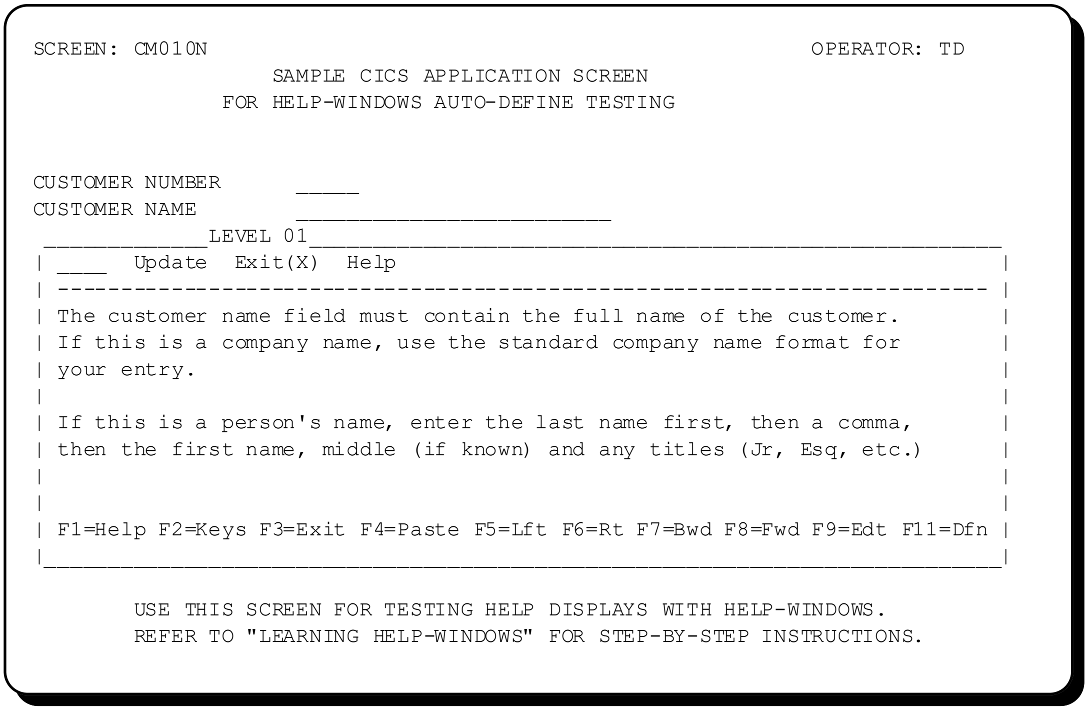
To exit auto-define mode, press PF3. If changes have been made to the help specifications, a message will display asking if you wish to save the changes. Enter "Y" or "N" accordingly. Then the original HDMO display will reappear.

STEP 9. TEST THE HELP DISPLAY

Now, position the cursor anywhere in the data entry field for CUSTOMER NAME, and press the Help Access key.

Upon pressing the help key, you should see the help display appear as the end-user will see it. That is, only the text displays in the window, with all control statements removed.

It should appear as follows:



```
SCREEN: CM010N                                              OPERATOR: TD
                   SAMPLE CICS APPLICATION SCREEN
               FOR HELP-WINDOWS AUTO-DEFINE TESTING


CUSTOMER NUMBER      _____
CUSTOMER NAME        _________________________
 _____________LEVEL 01__________________________________________________
| ____  Update  Exit(X)  Help                                           |
| --------------------------------------------------------------------- |
| The customer name field must contain the full name of the customer.   |
| If this is a company name, use the standard company name format for   |
| your entry.                                                           |
|                                                                       |
| If this is a person's name, enter the last name first, then a comma,  |
| then the first name, middle (if known) and any titles (Jr, Esq, etc.) |
|                                                                       |
|                                                                       |
| F1=Help F2=Keys F3=Exit F4=Paste F5=Lft F6=Rt F7=Bwd F8=Fwd F9=Edt F11=Dfn |
|_______________________________________________________________________|

        USE THIS SCREEN FOR TESTING HELP DISPLAYS WITH HELP-WINDOWS.
        REFER TO "LEARNING HELP-WINDOWS" FOR STEP-BY-STEP INSTRUCTIONS.
```

CREATING A SCREEN LEVEL HELP DISPLAY

Screen level help is defined much the same as field level help; however, instead of Step 5 above, *IDENTIFY THE HELP FIELD*, you should simply code the LOCATION field of the main AUTODEF window with all nines (9) or erase it. This will specify a generic field ID, and will be accessed only if no field level help is defined for the cursor position at the time the Help key is pressed.

CREATING A TRANSACTION LEVEL HELP DISPLAY

Transaction level help is defined as screen level help, except that nines should be coded in the SCREEN field, as well as the LOCATION field of the main AUTODEF window. Transaction level help is only displayed if no field level help is available for the cursor position, and no screen level help is available for the current display.

CREATING HELP DISPLAYS, DETAILED REFERENCE

THE AUTODEF WINDOW

This is the first window to appear when the DEFINE key is pressed. Here is where you identify the transaction code and location on the screen where the cursor will be when the help key is pressed.

Default values are provided for all fields, depending on the location of the cursor when the DEFINE key was pressed, but any of these may be changed if desired.

Following is a description of each field:

STATUS The STATUS indicator shows the current condition of the definition record, whether it has been updated or not, etc.

Status messages can be ...

NEW-RECORD	-	A new help definition is being created.
CHANGES PENDING	-	Changes have been made and not yet recorded. Press Enter to update the definition.
NO CHANGES	-	No changes have been made to this definition.
RECORD CHANGED	-	The updates have been successfully performed.

In addition, various error messages can appear in the STATUS indicator. For example, when PF3 is pressed from the main AUTODEF window and changes are pending, a message appears asking if the changes are to be saved or not.

TRAN The TRAN field defines the CICS transaction code of the screen for this help display. It will be filled in from the transaction in progress when the DEFINE key is pressed, but may be changed.

Transaction codes may be generic. You can substitute a question mark (?) for any character in the trancode. The remaining characters that are not question marks must match exactly in the corresponding positions. If the trancode is a PF or PA key (TASKREQ option), code the mnemonic value for the key as PA1, PA2 or PA3, PF1, PF2 ... PF24.

[Note] The use of generic transaction codes should be carefully considered before using them. Additional overhead is incurred, for all non-generic trancodes are searched first. In addition, it is sometimes difficult to control and retrieve the correct help display due to confusion of similar transaction codes. New transactions in CICS may use trancodes that unintentionally match the generic help definitions.

DESC The Description field is not required but is recommended for ease in tracking and maintaining help definitions. You should briefly describe the field or screen of this help display.

The description will display on the help directory, which is obtained by entering transaction code JHLP or JHLP,D.

The description is an aid to find the help definition record for a given help display. It is particularly useful if you need to perform direct maintenance to the definition record (the transaction key record).

SCREEN The SCREEN field is an additional means of qualifying a help display, besides the transaction code. You can create help displays that pertain only to certain screens of a transaction.

The field is filled with nines initially. This means that there is no unique screen identifier. To identify a screen, you must use either the BMS map name or some identifying text at a fixed location of the screen. Screen can be from one to eight positions long.

[Note] You should not use a unique screen identifier when defining nested help Nested help is help on help ... selections of a help menu, for instance, or further explanations of a given field. Nested help definitions should have all nines in the SCREEN field.

LOCATION The LOCATION designates whether this is to be transaction-level, screen-level or field-level help. If LOCATION is filled with nines, it is either transaction-level or screen-level, depending on the contents of SCREEN.

Otherwise, this help display is to be invoked when the help key is pressed while the cursor is in or near a particular field of the screen.

The value in LOCATION is either a row/column display address, or some identifying text on the screen. For text on the screen, the cursor can be designated as IN the text, or in position RELATIVE to the text (either before or after it). Row/col is entered as a 5-digit number (RRCCC).

Text identification is recommended, wherever possible, because it offers much more flexibility for maintenance purposes. Also, text identification must be used for nested help definitions.

SEQUENCE The SEQUENCE number is used for help window chaining. This allows the size and/or position of the help window to change automatically as the operator browses forward through the help text.

The initial sequence number of a new definition is always '01'. At any point, by increasing the value by one, a new definition record is created for the same transaction, screen and field identifiers.

By changing the window configuration for the new definition, subsequent help text will display in the new window. You can designate whether the old window is to remain on the screen (cascade effect) or disappear when the new window pops up by use of the Refresh Screen option of the Window alteration function (PF4).

If help window chaining is not desired, leave the sequence number as '01'.

SYSTEM The SYSTEM field is the high-level qualifier of the help text. Help text is identified by five hierarchical qualifiers in the full HELP-WINDOWS product: SYSTEM, DOCUMENT, SECTION, SUBJECT, and RECORD NUMBER.

In the CICS-JUGGLER Help-Windows feature, the DOCUMENT, SECTION and SUBJECT are implicit. Document will always be FASTPATH, Section is '001' and Subject is blank. You can, however, define new SYSTEMS, which is a way to segregate your help text into logical sets.

If you're defining help for all transactions of the accounting system, for instance, you could define a new system called ACCTNG, enter all your help text, then define a different system for inventory. etc.

The default SYSTEM is JUGLHELP, which is supplied. To define a new system, simply enter a different system name.

FIELD IDThe FIELD ID is a unique name to be given to this set of help text. It is twenty positions long and may contain any characters, including imbedded spaces.

When the DEFINE key is pressed, the text at the cursor location is the default value placed in FIELD ID, if there is any. If this is not the name you want to assign to this field text, enter a different name.

When the EDIT key (PF9) is pressed, to enter the help text, the FIELD ID is used to automatically construct the STRTFLD and STOPFLD statements, which are used to bracket the help text for this display. The field is not considered to be 'defined' until a text record is created containing a STRTFLD and

STOPFLD statement using the FIELD ID. At that point, a pointer record called the Cross Document (XDOC) record is created.

The names used for FIELD ID also display in the Table of Contents, when the table of contents is displayed in Define mode.

DIRECTIONAL OPTIONS AT THE AUTODEF WINDOW

The last line of the window is used to display the directional options available at this point. You can invoke one of these options by pressing the corresponding PF key. The available options are:

PF2 - (KEYS) PF key prompts are displayed at the bottom of the window, however if the window is to narrow to display all available prompts, you may press PF2 to temporarily display the prompts in a vertical manner within the current window. Press Enter or PF3 to exit the Keys display, any other PF key will exit the keys display then perform the function shown in the Keys window.

PF3 - (EXIT) Use this key to exit from the current display to the previous logical level.

PF4 - (WINDOW) Pressing this key will display the Window Sizing window. This display is used to specify the window size and position, whether the help display should appear as a full screen (rather than a window), etc. For more information, see *ALTERING THE WINDOW SIZE AND POSITION*, later in this section.

PF5 - (LOC) Pressing this key will display the Field Location window. This display is used to specify the location of the field on the application screen for which this help display is to be associated. For more information, see *IDENTIFYING THE FIELD LOCATION*, later in this section.

PF6 - (SCREEN) Pressing this key will display the Screen Identifier window. This display is used to ensure that this help display can only be accessed from a screen (or screens) with the same identifying information. For more information, see *IDENTIFYING THE SCREEN*, later in this section.

PF9 - (EDIT) Pressing this key will display the full screen text editor, which is used to create the help text that is to display. For more information, see *THE HELP TEXT EDITOR*, later in this section.

ALTERING THE WINDOW SIZE AND POSITION

This window appears when PF4 is pressed from the AUTODEF window. It's purpose is to change the size and/or position of the help window on the screen. In addition, it defines how text will display in the window, if previous juggler are to be removed when this one displays, and whether the text should display in full-screen or window mode.

The current values in the fields, which describe the current configuration of the window, are derived from the model window that was located when the DEFINE key was pressed. This can be the original distributed model, or it will use the values from previous help defined for this transaction.

METHODS OF ALTERING THE WINDOW

You can change the window configuration in any of three ways:

1) Use PF keys PF4 through PF11 to adjust the window left, right, up, down, taller, shorter, wider or narrower. These keys move one row or column at a time until a window adjustment command is issued.

2) Issue a window adjustment command in the command area (==>). These commands are WL, WR, WU, WD, WT, WS, WW and WN, for left, right, up, down, taller, shorter, wider or narrower. You can optionally follow the command with =xx where xx is the number of rows or columns to move.

3) Directly change the Starting row/col and/or Number rows/cols fields to position the window where you want it.

STARTING ROW/COL

The Starting row/col defines the row/column address of the upper left-hand corner of the window.

The row field is two positions and may contain a value from one to 43.

The column field is three positions and may contain a value from one to 132.

The starting row/col will not automatically adjust for different screen sizes. If you are creating help for terminals where some are model 2 (24 x 80) and others are model 3, 4 or 5, you should tailor your help window position to the model 2 screens. Otherwise it will not display correctly.

NUMBER ROWS/COLS

The Number rows/cols defines the depth and width of the window.

The first field is the number of rows that the window is deep. It is the number of text rows, including the action bar and the prompt line. Thus, if number of rows is set to 21, for instance, the maximum lines of help text that will display in the window is 18, since the action bar and prompt line require a total of three lines.

The second field is the number of columns the window is wide. This is a count of the number of text columns, not including the side window borders. The side window borders require five positions, so the maximum window width on a 80-character line is 75.

Number of rows can be up to 43, number of columns can be up to 132.

WORD WRAP The Word wrap field can contain a yes or no (Y or N) to specify whether word-wrap is to be activated for this window or not.

Word-wrap removes the requirement that the text in the window extend no wider than the window by causing the text to break on word boundaries when displayed in a help window. In other words, it allows you to create text wider than the window, but the operator will not have to pan left and right in order to read it all. They will only need to scroll down.

You can mix text that wraps with text that does not wrap, which is often desirable when columnar data is presented in a window. To do this, code an 'N' (no-wrap) in the line command field of the text editor for each line that is not to be wrapped.

REFRESH SCREEN

The Refresh screen field can contain a yes or no (Y or N) to specify if other help windows present on the screen are to be removed when this windows is displayed.

If 'Y' is coded, the original transaction screen is refreshed prior to displaying this window, thereby removing all lingering windows from other displays.

If 'N' is coded, this window will overlay the screen with no refresh, so that previously displayed windows will still appear. This allows you to create cascade windows that partially overlay one another.

The Refresh screen option is only applicable when you are defining nested help, such as a help menu, or further expansion of a current display.

FULL SCREEN	The Full screen field can contain a yes or no (Y or N) to specify whether the help text is to display in a window or not.

If 'Y' is coded, the text will display in full screen mode, with no borders or action bars.

If 'N' is coded, the text displays in a window, according to the size and position specifications of Starting row/col and Number rows/cols.

DIRECTIONAL OPTIONS AT THE WINDOW SIZING WINDOW

The last line of the window is used to display the directional options available at this point. You can invoke one of these options by pressing the corresponding PF key. The available options are:

PF2 - (KEYS) PF key prompts are displayed at the bottom of the window, however if the window is to narrow to display all available prompts, you may press PF2 to temporarily display the prompts in a vertical manner within the current window. Press Enter or PF3 to exit the Keys display, any other PF key will exit the keys display then perform the function shown in the Keys window.

PF3 - (EXIT) Use this key to exit from the current display to the previous logical level.

PF4 - (LEFT) Pressing PF4 will move the window left by one position. To move more than one position, enter WL=nn in the command area, where nn is the number of positions move.

PF5 - (RIGHT) Pressing PF5 will move the window right by one position. To move more than one position, enter WR=nn in the command area, where nn is the number of positions move.

PF6 - (UP) Pressing PF6 will move the window up by one position. To move more than one position, enter WU=nn in the command area, where nn is the number of positions to move.

PF7 - (DOWN) Pressing PF7 will move the window down by one position. To move more than one position, enter WD=nn in the command area, where nn is the number of positions to move.

PF8 - (TALL) Pressing PF8 will enlarge the window by one row. To increase by more than one position, enter WT=nn in the command area, where nn is the number of rows.

PF9 - (SHORT) Pressing PF8 will shorten the window by one row. To decrease by more than one position, enter WS=nn in the command area, where nn is the number of rows.

IDENTIFYING THE FIELD LOCATION

This window appears when PF5 is pressed from the AUTODEF window. Its purpose is to further define the screen location for which this help is to apply, or to define a new screen location for another help definition.

This function must be used if you are defining help with relative cursor position or if you want to change from text recognition to row/column or vice-versa.

If you are defining help using cursor in text, you can avoid this function by simply placing the cursor in the field to be documented prior to pressing the DEFINE key.

Fields can be associated with help displays using three methods. All pertain to the position of the cursor when the help key is pressed. The three techniques are:

Row/Column address	-	Absolute position on the screen
Cursor in Text	-	Cursor is in a field containing unique text.
Relative Cursor Position	-	Cursor is either preceding or following a field containing unique text.

Row/Column Address

To specify that this help is to be retrieved when the cursor is at a fixed location on the screen, you must use row/column recognition. At the Identify Location function, simply place the cursor at the desired spot and press PF5 (Row/col).

Alternatively, you can key the row/col address directly in the Location field of the AUTODEF window. It is coded as RRCCC, where RR is the row number and CCC is the column number (5 positions required).

Row/column recognition will locate the correct text if the cursor is anywhere within the field at the designated screen address. If you only want this help to appear if the cursor is at this exact location, you must code EXACT POSITION as 'Y'.

Cursor in Text

Text recognition provides greater flexibility than row/column, since it does not matter where on the screen the field appears. There must simply be some unique identifying text. The text can be up to 20 positions long and is coded in the Location field of the AUTODEF window.

To specify text recognition, place the cursor in the desired text and press PF4 (Text) at the Identify Location function. It does not matter whether the text on the screen is lower case or upper case.

Cursor in text recognition means that the cursor must be placed somewhere within the field containing the text. If Exact position is coded 'Y', the cursor must be positioned on the first character of the text for the help to be retrieved. To use cursor in text recognition, code a 'Y' in the CURSOR IN TEXT? field.

Relative Cursor Position

Relative cursor position means that the cursor is placed somewhere before or after the identifying text. This is a common situation with many screens, when the field title precedes the data fields, such as:

NAME: ____________________________________

To specify relative cursor position, you must first use text recognition (place the cursor on the text and press PF4). Then you must specify an Increment factor, as a positive or negative number. A positive increment means the data field precedes the identifying text, where as a negative number means the data field follows it, as in the example above.

You can compute the increment factor yourself and key it, or you can let HELP-WINDOWS compute it for you, as follows:

1) Before pressing the DEFINE key, position the cursor on the first position of the text. In our preceding example, place the cursor on the 'N' of NAME. Now press the define key. 'NAME' will appear in the Location field.

2) Now press PF5 (Loc) to invoke the Identify Location function.

3) Move the cursor back to the first position of the data field.

4) Press PF6 (Increment). The correct positive or negative increment value will be placed in Increment value.

   This help display will now appear when the cursor is anywhere in the data field. To limit it to only the first position, code EXACT POSITION as 'Y'.

Following is a description of each of the fields of the Identify Location function:

LOCATION LOCATION is the identifying text or row/column address where the cursor will be when the help key is pressed.

If text recognition is used, there can be up to twenty characters of text. The only limitation is that the text must be contained in a single field of the screen. There can be no imbedded attributes in the text.

If row/column recognition is used, LOCATION must contain a five-digit number in the form RRCCC, where RR is the two-position row number and CCC is the three-position column number (precede with zeros if needed).

USE ROW/COLUMN?
USE ROW/COLUMN must contain 'Y' if the value in LOCATION is to be a row/column address. If it is text, this field should contain 'N'.

If USE ROW/COLUMN contains 'N', even though the value in LOCATION is coded as RRCCC, it will still be considered to be identifying text.

When PF5 is pressed in the Identify Location function, this field is automatically updated with a 'Y'.

INCREMENT VALUE
INCREMENT VALUE contains a non-zero integer if Relative Cursor Position is being used. A positive number indicates that the cursor is positioned prior to the identifying text on the screen. A negative value means that the cursor appears after the text.

Negative numbers can be entered with the minus sign (-) either before or after the number. It is displayed with the minus sign following.

If PF6 (set increment) is pressed in the Identify Location function, the increment value is computed from the current cursor position to the cursor position at the time the DEFINE key was pressed.

[Note] For Row/column recognition, this field does not apply.

TEXT LENGTH This field designates the number of characters in the screen field where the cursor will be when the help key is pressed.

For row/column recognition, the field length determines how far from the absolute row/column address the cursor can be and still considered a "hit" for this help text. This only applies if EXACT POSITION is coded 'N'.

For text recognition, field length is the number of characters in the identifying text to be considered. It must be equal to or less than the actual text length, but not greater.

With row/column recognition, you can specify a field length much longer than the true distance between attributes. This allows the same help text to be found for several like fields on a horizontal row.

EXACT POSITION?

EXACT POSITION must contain 'N' (no) if you want the capability to have the cursor anywhere within a field and still retrieve this help text. That is, the cursor does not have to be in the exact row/column address or positioned on the first character of identifying text to still work.

If coded 'Y', it works as follows for each method:

Row/column recognition	-	Cursor must be in the absolute row/column.
Cursor in text	-	Cursor must be on the first text character.
Relative cursor position	-	Cursor must be exactly the number of characters removed from the text as specified by the increment value.

CURSOR IN TEXT?

The CURSOR IN TEXT field designates, for text recognition, whether the cursor must be positioned in the field containing the identifying text or if it can be outside the field, using Relative Cursor Position.

Code 'Y' if the cursor is to be positioned somewhere in the text, otherwise code 'N'. If 'N' is coded, there must be a non-zero integer in the INCREMENT field.

DIRECTIONAL OPTIONS AT THE FIELD LOCATION WINDOW

The last line of the window is used to display the directional options available at this point. You can invoke one of these options by pressing the corresponding PF key. The available options are:

PF2 - (KEYS) PF key prompts are displayed at the bottom of the window, however if the window is to narrow to display all available prompts, you may press PF2 to temporarily display the prompts in a vertical manner within the current window. Press Enter or PF3 to exit the Keys display, any other PF key will exit the keys display then perform the function shown in the Keys window.

PF3 - (EXIT) Use this key to exit from the current help display to the previous logical level. (This may be an earlier help display or the transaction is progress.)

PF4 - (TEXT) When using Cursor In Text or Cursor Relative To Text methods of defining help, you may position the cursor to the text on the application screen that is to be used for identifying this help display and press this PF key. This will place the text in the LOCATION field and the length in the TEXT LENGTH field.

PF5 - (ROW/COL) When using the Row/Column method of defining help, you may position the cursor to the row/column position on the application screen and press this PF key. This will place the row/column in the LOCATION field and place a 'Y' in the USE ROW/COLUMN field.

PF6 - (INCREMENT) When using Cursor In Text or Cursor Relative To Text methods of defining help, you may position the cursor to the field on the application screen in which the cursor should be when the Help Access key is pressed and press PF6. This will place the proper increment in the INCREMENT VALUE field.

IDENTIFYING THE SCREEN

This window appears when PF6 is pressed from the AUTODEF window. It's purpose is to define this help definition as screen level or screen specific. In other words, this help definition is only to be invoked for a particular screen of a transaction, even though the field location information might match correctly on other screens.

There are two methods of identifying a screen. They are:

1) Unique text at a fixed screen location
2) BMS map name

Text identification is assumed. For BMS map identification, press PF5.

SCREEN ID The SCREEN ID is the identifying text or the BMS map name which will uniquely identify this screen display.

The information in the SCREEN ID field is taken from the SCREEN field of the AUTODEF window. If this is an existing (already created) definition, changing the SCREEN ID will result in the creation of a new definition. In other words, you cannot change the SCREEN ID once a definition has been added to the file. You must create a new one, then delete the old.

The SCREEN ID can be one to eight characters in length and may contain any alphanumeric characters. To change the value in the field, you can key new data or move the cursor to the desired text on the screen and press PF4 (Text).

STARTING ROW

STARTING ROW is the row number on the display screen where the text in the SCREEN ID field will be found. It can contain a number from 1 to 43.

Be aware that HELP-WINDOWS will not automatically adjust for different screen sizes as it pertains to locating the screen identifier. For instance, if the identifying text appears at row 3, column 5 on a model 2 (24 x 80) screen, but row 1, column 35 on a model 5 (27 x 132) screen, you must make two definitions for all help unique to that screen. If you have this condition, try to find identifying text on row 1.

If you place the cursor on the first position of identifying text and press PF4 (Text), the starting row number will be automatically supplied.

STARTING COLUMN

STARTING COLUMN is the column number on the display screen where the text in the SCREEN ID field will be found. It can contain 001 to 132.

Be aware that HELP-WINDOWS will not automatically adjust for different screen sizes as it pertains to locating the screen identifier. For instance, if the identifying text appears at row 3, column 5 on a model 2 (24 x 80) screen, but row 1, column 35 on a model 5 (27 x 132) screen, you must make two definitions for all help unique to that screen. If you have this condition, try to find identifying text on row 1.

If you place the cursor on the first position of identifying text and press PF4 (Text), the starting column will be automatically supplied.

TEXT LENGTH This field is the number of positions of the identifying text in the SCREEN ID field to be considered. Normally it would be the length of the data in screen ID, but it can be less.

To uniquely identify a screen, HELP-WINDOWS goes to the row/column address specified by STARTING ROW and STARTING COLUMN, compares the data at that location to the data in SCREEN ID for the length specified in TEXT LENGTH.

START OF MAP NAME

This field only occurs on the BMS screen identifier window (PF5 from the Screen Identifier window).

START OF MAP NAME designates the first position of the BMS map name to be used to identify the screen.

Normally this would be 1, but this option allows you to control one map name generically. It might be the case that you have several maps where the name is the same in the last four positions. You could define screen specific help that would apply to all those maps by designating the starting position as four and the length as four.

LENGTH TO USE

This field only occurs on the BMS Screen Identifier window (PF5 from the Screen Identifier window).

LENGTH TO USE designates the length of the BMS map name to be used to identify the screen.

Normally this would be the full name length, but you can control one map name generically. It might be the case that you have several maps where the name is the same in the last four positions. You could define screen specific help that would apply to all those maps by designating the starting position as four and the number of bytes as four.

DIRECTIONAL OPTIONS AT THE SCREEN IDENTIFICATION WINDOW

The last line of the window is used to display the directional options available at this point. You can invoke one of these options by pressing the corresponding PF key. The available options are:

PF2 - (KEYS) PF key prompts are displayed at the bottom of the window, however if the window is to narrow to display all available prompts, you may press PF2 to temporarily display the prompts in a vertical manner within the current window. Press Enter or PF3 to exit the Keys display, any other PF key will exit the keys display then perform the function shown in the Keys window.

PF3 - (EXIT) Use this key to exit from the current help display to the previous logical level. (This may be an earlier help display or the transaction is progress.)

PF4 - (TEXT) You may position the cursor to the text on the application screen that is to be used for identifying this screen and press this PF key. This will fill all fields in the window to the appropriate values.

PF5 - (BMS) If using a BMS mapped screen, press this PF key to display the BMS Mapped Screen Identification window.

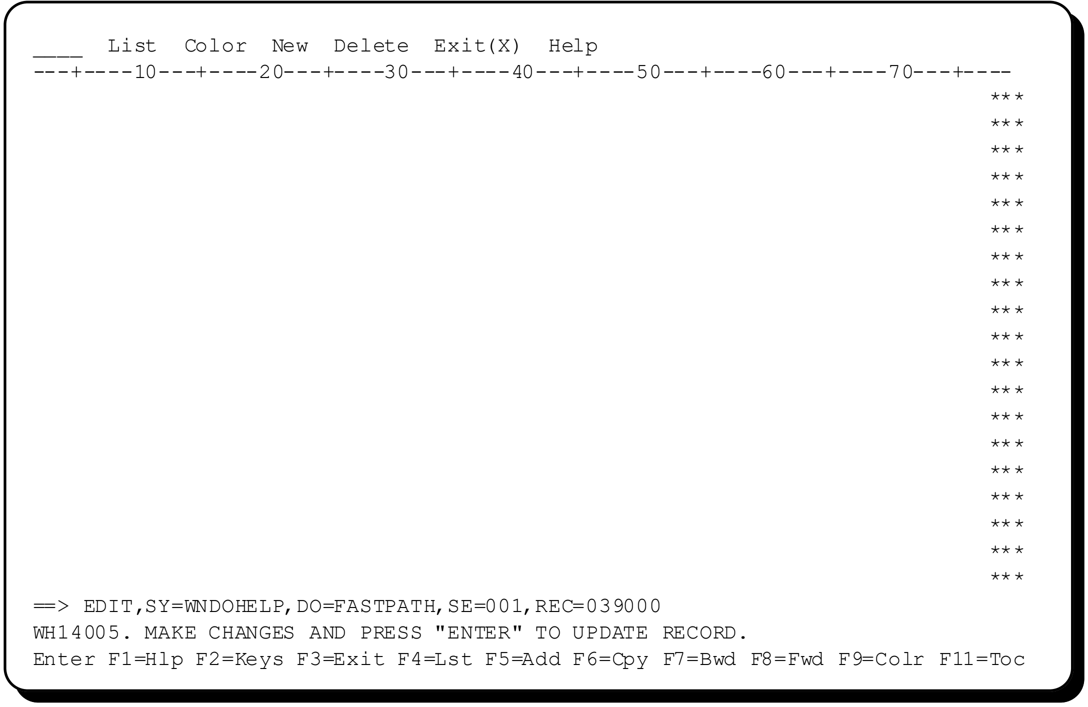
THE HELP TEXT EDITOR

The Help Text Editor is used for creating the display that the operator will see when the help key is pressed.

The help text is constructed in text records, each record containing 19 lines, one editor display page. When displayed in List form, or in a help display, the text is contiguous from record to record. Blank lines are always null, allowing full use of the INS key to shift data. Trailing blank lines at the end of a record are removed when the text is displayed in a help window.

When one text record is full, you must add another text record (PF5) in order to continue. Text records are automatically numbered, the record number displaying on the command line (==>) in the REC= keyword. You can move from one text record to another either by browsing forward or backward (PF8/PF7) or by changing the current record number.

This is used to 'paint the screen' the way you want the text in the help window to be displayed. This screen appears as follows:



```
____  List  Color  New  Delete  Exit(X)  Help
---+----10---+----20---+----30---+----40---+----50---+----60---+----70---+----
                                                                             ***
                                                                             ***
                                                                             ***
                                                                             ***
                                                                             ***
                                                                             ***
                                                                             ***
                                                                             ***
                                                                             ***
                                                                             ***
                                                                             ***
                                                                             ***
                                                                             ***
                                                                             ***
                                                                             ***
                                                                             ***
                                                                             ***
                                                                             ***
                                                                             ***
==> EDIT,SY=WNDOHELP,DO=FASTPATH,SE=001,REC=039000
WH14005. MAKE CHANGES AND PRESS "ENTER" TO UPDATE RECORD.
Enter F1=Hlp F2=Keys F3=Exit F4=Lst F5=Add F6=Cpy F7=Bwd F8=Fwd F9=Colr F11=Toc
```

To enter text, simply key lines of data, formatting it as desired so far as line spacing, indentation, etc. are concerned. If changes are needed after keying some text, you can simply move the cursor to the area to be changed and type over any existing data. You can use the DEL key to delete characters in a line. To insert characters on a line, position the cursor where the insertion is to be made, press the INS key and type the desired characters. The ERASE EOF key may be used to erase all or part of a line at any time.

When all text for this page has been keyed, press ENTER. This will write the updated text record to file. If the text being entered cannot fit on one screen, you may press the FORWARD key (or change the line number at the top of the display).

LINE DELETION

To delete an entire line, TAB to the COMMAND FIELD at the end of the line (where the asterisks are) and enter a 'D', then press ENTER. The entire line will be deleted and all following lines will shift up one line.

To delete more than one line at a time, enter a 'D' in more than one COMMAND FIELD before pressing ENTER or enter 'Dnn' on one or more COMMAND FIELDS, where nn is the number of consecutive lines (including the current line) to be deleted.

You may also perform block deletion by placing 'DD' on the first line to be deleted and another 'DD' on the last line to be deleted.

LINE INSERTION

To insert a line between two text lines, tab to the COMMAND FIELD of the preceding line and enter an 'I' over any one of the asterisks, then press ENTER. This will cause a blank line to be inserted following that line, and all subsequent lines will shift down one line. You may then key any desired text on that line.

To insert multiple lines, enter 'Inn' in the preceding COMMAND FIELD, where nn is the number of lines to insert, then press ENTER. This will cause nn blank lines to be inserted after the current line, and all subsequent lines will shift down nn lines.

MOVING AND COPYING TEXT LINES

One or more consecutive text lines can be moved or copied to another location. To move text lines, enter 'M' or 'Mnn' (nn = number of lines to move) in a COMMAND FIELD and press ENTER. This causes the specified line(s) to be deleted from that location. Now place an 'A' (insert-after) in the COMMAND FIELD on the line preceding the new location, or a 'B' (insert-before) on the line following the new location and press ENTER. The moved lines will be inserted following that line if the 'A' command is used, or before that line for the 'B' command.

To copy text lines, enter 'C' or 'Cnn' (nn = number of lines to copy) in a COMMAND FIELD and press ENTER. Now place an 'A' (insert-after) in the COMMAND FIELD on the line preceding the new location, or a 'B' (insert-before) on the line following the new location and press ENTER.

You may also perform block moves and copies by placing 'MM' or 'CC' on the first line to be moved or copied and another 'MM' or 'CC' on the last line to be moved/copied.

You may enter more than one 'M' or 'C' command on one edit screen, if desired. The result is that all selected lines are grouped together consecutively and inserted as one set wherever the 'A' or 'B' command is placed. For instance, if 'C2' is entered in the command area at line 3, then 'C4' is entered in the command area at line 16, all six text lines will be grouped together and inserted as one set following the line where the 'A' command is placed.

Note that you may use the 'A' or 'B' sub-commands as many times as desired, without performing another 'M' or 'C' sub-command. Each time it is used, the same data that was moved or copied last will be inserted at the new location.

TEXT RECORD SPLITS

When a text record is too full to contain more data, or any time you need to make room to insert additional text, you can perform a text record split.

To perform a split, place an S in the command field opposite the text line you want as the first line of the next record, then press ENTER.

A new text record will be added following this record. All text of the previous record from the line containing the 'S' to the end will be deleted from the previous record and added starting at line 1 of the new record.

INSERT STRTFLD/STOPFLD

The F command can be used to insert a STRTFLD or STOPFLD statement at any point in the text. To use it, do the following:

Place the 'F' in the command field of the line after which you want the STRTFLD or STOPFLD statement, and press ENTER.

If there is no STRTFLD in this text record, a formatted STRTFLD will be inserted with no field name. You must key the field name.

If a STRTFLD already exists prior to this point in the text record, a formatted STOPFLD will be inserted with the field name found in that STRTFLD. This avoids the necessity of keying the STOPFLD and ensures that the field name will be correct.

DISPLAY ONE OR MORE BLANK LINES - SPACE

The SPACE command is entered on a text line by itself (in position 1) in the form:

<SPACE nn>

where nn is the number of blank lines to display at this point. The nn value can be entered as a single digit.

SKIP THE DISPLAY TO THE NEXT PANEL - EJECT

The EJECT command is entered on a text line by itself (in position 1) in the form:

<EJECT>

The EJECT command forces a panel break at this point in the display. In other words, all text following the EJECT will display at line one of the next help panel when browsing forward in the help display.

EJECT also forces panel breaks when browsing backward. It is useful, therefore, for maintaining proper alignment of help panels during forward and backward browse operations.

DEFINE THE BEGINNING OF TEXT FOR ONE FIELD - STRTFLD

The STRTFLD command is entered on a text line by itself (in position 1) in the form:

<STRTFLD XXXXXXXXXXXXXXXXXXXXX>

where xxx...xxx is the FIELD ID to be assigned to this block of text. The text is terminated with a STOPFLD statement containing the same Field Id. Field ID can be up to twenty characters long and contain any data.

The presence of a set of STRTFLD/STOPFLD statements with the same field ID causes the automatic creation of a special record called a Cross-document record. This record is keyed by the field ID and contains the starting and ending text record numbers for this text. The Cross-Document record is the means Help-Windows locates text.

You can nest sets of text by inserting STRTFLD/STOPFLD pairs around texts which is itself included in another STRTFLD/STOPFLD pair. There is no limit to the level of nesting. The only rule is that the Field ID on each STRTFLD/STOPFLD pair be unique.

Blocks of text delimited and named by STRTFLD/STOPFLD pairs can be the target of direct help displays by referencing the STRTFLD name in the Field ID of the AUTODEF window.

In addition, they can be the target of indirect displays by referencing the STRTFLD name on an INCLUDE statement.

DEFINE THE END OF TEXT FOR ONE FIELD - STOPFLD

The STOPFLD command is entered on a text line by itself (in position 1) in the form:

<STOPFLD xxxxxxxxxxxxxxxxxxx>

The STOPFLD statement marks the end of a set of text, the beginning of which was marked with a STRTFLD statement with the same name.

If the STOPFLD statement is omitted, the Cross Document record may be incomplete. The starting text record number will be recorded from the STRTFLD statement but the ending number is left zero. This means that only the first text record will display, then the Field ID will be considered to be at the end.

You can add missing STOPFLD statements at any time, plus move them from one text record to another using the M line command.

COPY AND DISPLAY THE TEXT FROM ANOTHER FIELD ID - INCLUDE

The INCLUDE command is entered on a text line by itself (in position 1) in the form:

<INCLUDE xxxxxxxx,yyyyyyyyyyyyyy>

where xxxxxxxx is the text SYSTEM name and yyy...yyy is the FIELD ID.

INCLUDE will retrieve the referenced block of text and insert it in-line when this point is reached as the text is being displayed. There can be any number of INCLUDE commands with or without normal text, and INCLUDE commands can be nested. That is, text retrieved with INCLUDE can have another INCLUDE within it. INCLUDE commands can be nested up to five levels.

CREATING COLOR AND HIGHLIGHTING

When using text attribute characters, set the attribute code where you want it to appear. Terminate it by repeating the same code where you want to return to normal intensity protected.

Unless otherwise specified, all words on the menu will be displayed in low intensity protected. In order to emphasize certain words or lines, you can use either of the following two methods:

1) Bracket selected text with text attribute characters. For this method, place one of the following characters before the text that is to be highlighted. The attribute will remain in effect for the remainder of the line or until another attribute is encountered. If another attribute of the same type is encountered, highlighting will revert back to the default (protected, low intensity).

   There are three text attribute characters, as follows:

Logical 'not' sign (¬)	-	Protected, high intensity
Pound sign (#)	-	Unprotected, high intensity
'At' sign (@)	-	Unprotected, normal intensity

2) Use the Color pull-down menu or press PF9 while in the editor to invoke the attribute set function. This method is described on the following pages.

To invoke the Attribute function, press the ATTRIBUTES key (shown in the PF key prompt area). This places the terminal in SET COLOR mode. Note that upon invoking the attribute function, the function key area changes to include additional functions. These additional functions are as follows:

ENTER — If the cursor is inside the attribute window, pressing ENTER will move the cursor to the current attribute pointer position.

F3=EXIT Exit Set Attribute mode to the editor.

F4=SET Set an attribute (as specified in the attribute window) at the current cursor position.

F5=SELECT Select the attribute at the position of the cursor and display the type of attribute in the attribute window. (If the cursor is inside the attribute window, this will move the window out of the way.)

F6=REMOVE Remove (delete) the attribute at the position of the cursor.

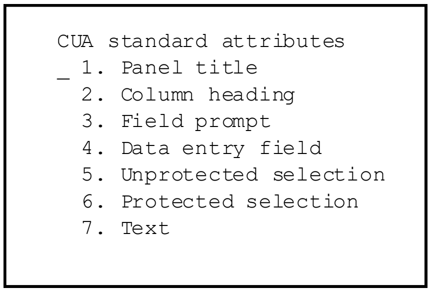
F7=BWD Move the current attribute pointer to the previous attribute.

F8=FWD Move the current attribute pointer to the next attribute.

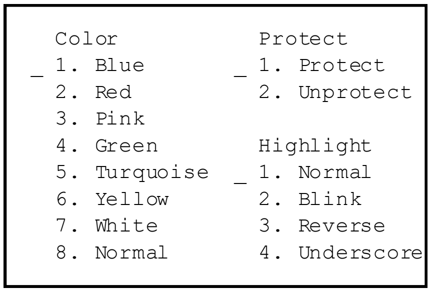
F9=EXTEND Toggle the attribute window between CUA and extended attributes.

F11=SHOW Show all attributes on the screen as an at-sign (@).

When the attributes function is invoked, a pop-up window appears. The attributes window can appear in two forms:



```
CUA standard attributes
_ 1. Panel title
  2. Column heading
  3. Field prompt
  4. Data entry field
  5. Unprotected selection
  6. Protected selection
  7. Text
```



```
  Color            Protect
_ 1. Blue        _ 1. Protect
  2. Red           2. Unprotect
  3. Pink
  4. Green         Highlight
  5. Turquoise   _ 1. Normal
  6. Yellow        2. Blink
  7. White         3. Reverse
  8. Normal        4. Underscore
```

- A CUA standards attribute window. This window contains the CUA attributes for creating a display. The following chart portrays the display attributes for the various selections:

Selection	Extended color	Intensity	Protect/unprotect
Panel title	Yellow	High	Protected
Column Heading	Turquoise	High	Protected
Field Prompt	Turquoise	Normal	Protected
Data Entry	Green	Normal	Unprotected
Unprotected Selection	White	Normal	Unprotected
Protected Selection	White	Normal	Protected
Text	Blue	Normal	Protected

- An extended attribute window. This window offers all attribute combinations.

If the CUA standard window is displayed and you wish to use the extended attribute window, press the EXTEND key (shown in the PF key area). This key works as a toggle between the two types of attribute windows.

Upon pressing the ATTRIBUTES key, either a percent sign or logical not sign may display somewhere on the screen. This is the Current Attribute Pointer. If an attribute is present at that position of the screen, it will appear as a percent sign, else it will appear as the not sign. You may press the FORWARD or BACKWARD keys to position the pointer to the next or previous attributes on the screen, or you may position the cursor to a suspected location of an attribute and press the SELECT key to move the attribute pointer directly to that position.

To set an attribute, simply select the choices within the attribute window by entering the number of the choice. Then position the cursor to a space before the word (or line) that is to contain the attribute features. Then press the SET key (shown in the function key area). The attribute will remain in effect until another attribute or the end of the line is encountered.

To remove an attribute, position the cursor to the attribute to be removed and press the REMOVE key (shown in the PF key area).

[Note] Setting color with either attributes windows may cause extended attributes to be generated in the screen display when viewed. You must be using a terminal that supports extended attributes and the COLOR and/or HILIGHT feature(s) must be set in the CICS terminal control table (TCT) entry (or EXTENDEDDS with RDO) in order to view them. If you view a display on a terminal that does not support extended data streams, the extended color and highlighting attributes will be removed (in favor of regular field attributes) when the screen is displayed.

FINDING ATTRIBUTES ON THE SCREEN

Since attributes are normally invisible it is sometimes difficult to locate them in order to change or remove them. To do this, use the Show attribute function, which can be invoked either from the Show pull-down menu or by pressing PF11 when the attribute window is up.

Show attributes will show the location of all attributes on the screen by placing an at-sign at the attribute position. You can then move the cursor over an attribute and remove it with PF6 or reset a different attribute with PF4.

You can also use PF7 and PF8 to locate attributes either before (PF7) the current cursor location or after (PF8). When pressed, a percent sign will appear at the attribute location and the cursor will stop there.

MARKING AN ATTRIBUTE LOCATION - THE SELECT KEY

When the cursor is positioned where you want to place an attribute but you need to change the attribute to be set, you can save time by first pressing the Select key (PF5). This will display a not sign at the cursor location. Now tab to the attribute window and change the number of the attribute to be set.

When you subsequently press Enter, the cursor will jump back to the not sign. Now press PF4 to set the attribute.

ENTERING COMMANDS AT THE TEXT EDITOR

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key.

ENTER Update the menu or message and/or apply any commands in the command field (to the right of the screen).

PF1 HELP Display a Help screen for information pertaining to the text editor.

PF3 EXIT Exit to the previous logical level.

PF4 LST Display the text in List form. Text will display as it will appear to the end user, though still in full screen mode. All record breaks are removed, the text is contiguous, and all imbedded text commands (SPACE, EJECT, etc.) are honored.

PF5 ADD Add a new, empty text record immediately following the one currently displayed. The new record is then presented, ready to receive additional text.

The record sequence number for the new record is computed by reading ahead to the next record and halving the distance between its number and the number of the current record. If a new record cannot be inserted (sequence numbers differ by less than two), the text records must be reorganized in order to add another at this point. This can be accomplished by copying the next record to a new number, then deleting it, thereby making room, or by performing a Renumber command. See Miscellaneous Commands and Functions

PF6 COPY Copy the entire contents of the current text record and add it wherever you choose. After pressing PF6, you will be asked to press ENTER to complete the copy. At that point, you must change something on the command line, usually only the record sequence number, but you could also direct this text record to a different SYSTEM, if desired.

The COPY command accomplishes the same effect as doing a line copy for all lines of a text record, then doing an Add at the appropriate point in the receiving text, followed by a line insert of the copied text.

PF7 BACKWARD Display the previous page of text.

PF8 FORWARD Display the next page of text.

PF9 COLR Set the editor to attribute mode. This will display a different function key area at the bottom of the screen. For more information, please refer to *CREATING COLOR AND HIGHLIGHTING* earlier in this section.

PF11 TOC Display the Table of Contents.

While in the editor, the table of contents produces a different display than that seen while in help mode. The help mode table of contents is only those text lines that are marked as a TOC entry. The editor table of contents includes those, if any, but also displays a listing of cross-document field names. This will include all field names that have been defined via STRTFLD statements.

When the table of contents is displayed, you can tab to any line and press PF9 to edit the text associated with that field ID.

The edit table of contents makes it easy to locate text, or find the field ID that was used to define it.

COMMANDS OF THE EDITOR COMMAND LINE

The command line of the editor is denoted by ==> at the bottom of the screen. All commands on the command line must be followed by the text record qualifiers, which are the keywords identifying the SYSTEM, DOCUMENT, SECTION and RECORD NUMBER.

These keywords are:

SY=system,DO=document,SE=section,REC=record number

The editor command itself must precede the first keyword. The following describes the commands available for managing the text editor. Note that all commands may be abbreviated down to two characters, and you can shift the command line left and right with the INS and DEL keys.

ADD Add an empty text record at the designated number. This is the same as pressing PF5 while in the editor.

BACKWARD Browse backward through the text file. Same as pressing PF7.

COLOR Invoke the attribute set window for assigning color or highlight attributes to text. Same as pressing PF9.

COPY Copy all text from this record and create a new record, adding it at a designated location. After entering the COPY command, you will be prompted to press Enter to add. At that point, the command line must specify the target, system, document, section and record number. Note that you cannot copy text records which contain a STRTFLD, as that would create a duplicate Field ID.

DELETE Delete the current text record. After entering the delete command, you will be prompted to press ENTER to complete it. At that time, the record number on the command line is be deleted.

EDIT Accept all input in edit mode, rewrite the text record with any changes and process all editor line commands. Also performs validity checks on imbedded text commands.

FORWARD Browse forward through the text file. Same as pressing PF8.

HELP Display a help screen for editor instructions. Same as PF1.

LIST Display this and following text records in list mode, as it will be viewed by the end-user. Trailing blank lines are dropped and imbedded text commands are interpreted. Same as pressing PF4.

KEYS Display the list of PF key assignments available while in the editor. This is the same as pressing PF2.

QUIT Exit the editor and return to previous level. Same as PF3.

RENUM Renumber all text records in this system. This command will assign new record numbers to all text records, incrementing each by 1000 or the value specified by the INCR=nnnn keyword, if present. For more information on renumbering text records, see *MISCELLANEOUS COMMANDS AND FUNCTIONS*, later in this section.

SEARCH This command will continue a text search for a previously entered character string.

T This command continues a search command in reverse order.

USING THE JHLP DIRECTORY

The directory is provided to quickly locate and optionally change or copy the various parts of a help definition.

There are three directories available. These are:

1) The transaction directory. A display of all TRANSACTION KEY records.
2) The field directory. A display of all CROSS-DOCUMENT records.
3) The system directory. This is a display of all text SYSTEM names.

The directories are invoked with the JHLP transaction code. If it is entered with no operands, the Transaction Directory will display, starting with the first transaction definition on file.

TRANSACTION DIRECTORY COMMANDS

JHLP	Start display with first trancode
JHLP,D	Start display with first trancode
JHLP,D,TR=xxxx	Start with trancode xxxx
JHLP,D,TR=xxxx,SCR=yyyyyyyy	Start at tran xxxx, Screen yyyyyyyy
JHLP,D,TR=xxxx,SCR=yyyyyyyy,LOC=zzz...	Tran x, Screen y, Location zzz..zzz

[Note] To position to a particular location, TR= and SCR= must be fully entered, but LOC= does not. The directory will start with the closest to whatever is entered on the LOC= keyword.

The transaction directory is a listing of Transaction Key records. These are automatically created when an Auto-Define function is performed.

Once the transaction directory is displayed, you can browse forward and backward with PF8/PF7. After tabbing down to an entry, you can perform any of the following functions on that entry by pressing a key

Enter - Bring that entry to the top of the display.
PF2 - Copy that transaction key record using new key information.
PF4 - Display the transaction key record for update.
PF6 - Delete that transaction key record
PF9 - Edit the text associated with that transaction key record.

FIELD DIRECTORY COMMANDS

JHLP,D,FI Start display with first field definition
JHLP,D,SY=xxxxxxxx,FI Start with first field of system xxxxxxxx
JHLP,D,SY=xxxxxxxx,FI=yyy...yyy Start at system xxxxxxxx, field yyy...yyy

[Note] To position to a particular field, System must be fully entered, but Field can be generic. The directory will start with the field closest to whatever is entered on the FI= keyword.

The Field directory is a listing of Cross Document records. These records are automatically created from the STRTFLD/STOPFLD statements in the text.

Once the field is displayed, you can browse forward and backward with PF8/PF7. After tabbing down to an entry, you can perform any of the following functions on that entry by pressing a key.

Enter - Bring that entry to the top of the display.
PF2 - Copy that cross-document record using new key information.
PF4 - Display the cross-document record for update.
PF6 - Delete that cross-document record
PF9 - Edit the text associated with that cross-document record.

SYSTEM DIRECTORY COMMANDS

JHLP,D,SY Start display with first system record
JHLP,D,SY=xxxxxxxx Start display with system xxxxxxxx

[Note] To position to a particular system, the SY= value can be generic. The directory will start with the system closest to whatever is entered on the SY= keyword.

The System directory is a listing of SYSTEM records. These records are automatically created when a new text system is entered in auto-define.

Once the directory is displayed, you can browse forward and backward with PF8/PF7. After tabbing down to an entry, you can perform any of the following functions on that entry by pressing a key

Enter - Bring that entry to the top of the display.
PF9 - Edit the first text associated with that system record.

[Note] The remaining functions on the prompt line (Copy, Update, Add Delete) cannot be performed on system records. You can not apply direct maintenance to system records.

MISCELLANEOUS COMMANDS AND FUNCTIONS

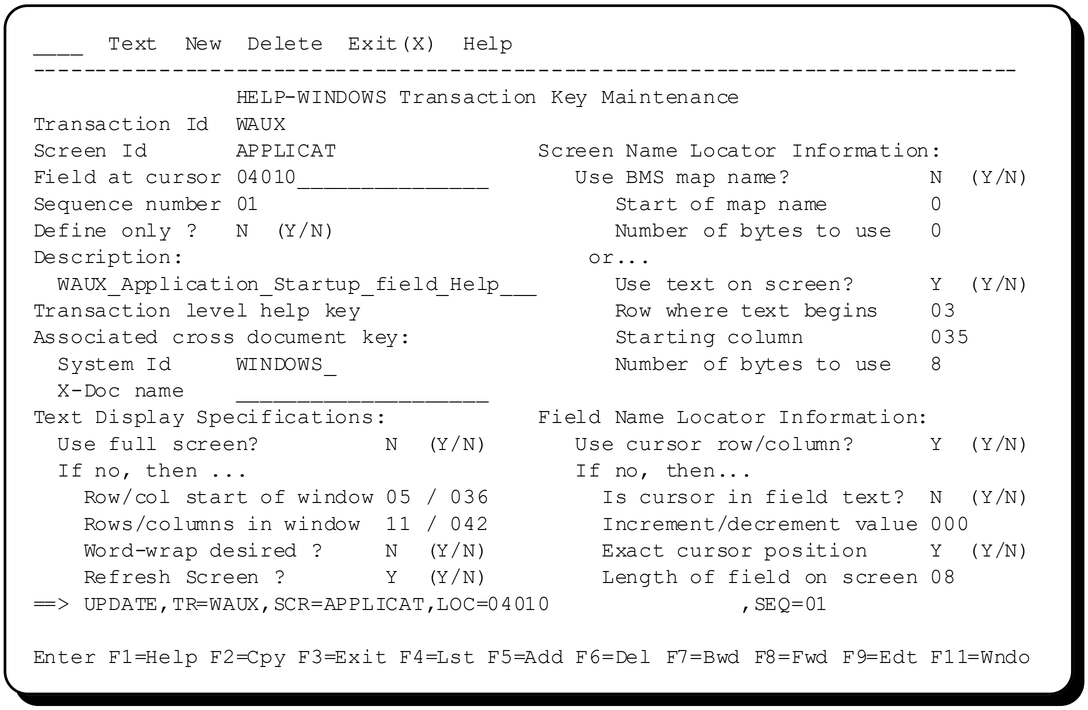
Additional functions and features of the Help-Windows feature of CICS-JUGGLER appear below.

TRANSACTION KEY RECORD DIRECT MAINTENANCE

The transaction key record is the "bridge" between the screen display and the correct help text. It is defined automatically as you move through the various windows of the Auto-define process, and therefore never needs direct maintenance.

You can, however, fully define help definitions or modify existing definitions by locating the correct transaction key record and modifying it, or adding an all new transaction key record.

To retrieve a transaction key record for direct maintenance, display the transaction directory by entering JHLP,TR=xxxx, where xxxx is the transaction code desired. Then locate the correct record, either by the description field or the SCRN ID / FIELD NAME information. Position the cursor by that entry and press PF4 (update). The transaction key record will appear generally as follows:



```
____  Text  New  Delete  Exit(X)  Help
------------------------------------------------------------------------------
                HELP-WINDOWS Transaction Key Maintenance
Transaction Id  WAUX
Screen Id       APPLICAT                Screen Name Locator Information:
Field at cursor 04010_______________       Use BMS map name?           N  (Y/N)
Sequence number 01                            Start of map name        0
Define only ?   N  (Y/N)                      Number of bytes to use   0
Description:                                or...
  WAUX_Application_Startup_field_Help___      Use text on screen?      Y  (Y/N)
Transaction level help key                    Row where text begins    03
Associated cross document key:                Starting column          035
  System Id     WINDOWS_                      Number of bytes to use   8
  X-Doc name    ____________________
Text Display Specifications:            Field Name Locator Information:
  Use full screen?          N  (Y/N)       Use cursor row/column?      Y  (Y/N)
  If no, then ...                          If no, then...
    Row/col start of window 05 / 036         Is cursor in field text?  N  (Y/N)
    Rows/columns in window  11 / 042         Increment/decrement value 000
    Word-wrap desired ?     N  (Y/N)         Exact cursor position     Y  (Y/N)
    Refresh Screen ?        Y  (Y/N)         Length of field on screen 08
==> UPDATE,TR=WAUX,SCR=APPLICAT,LOC=04010               ,SEQ=01

Enter F1=Help F2=Cpy F3=Exit F4=Lst F5=Add F6=Del F7=Bwd F8=Fwd F9=Edt F11=Wndo
```

The fields of the transaction key record will be familiar to you once you have performed Auto-define. Each of the set of fields from the various AUTODEF windows appear in four quadrants of the transaction key record.

All of the same rules apply to the fields of the transaction key record as for the Auto-define process. For help on individual fields, position the cursor to the desired field and press PF1.

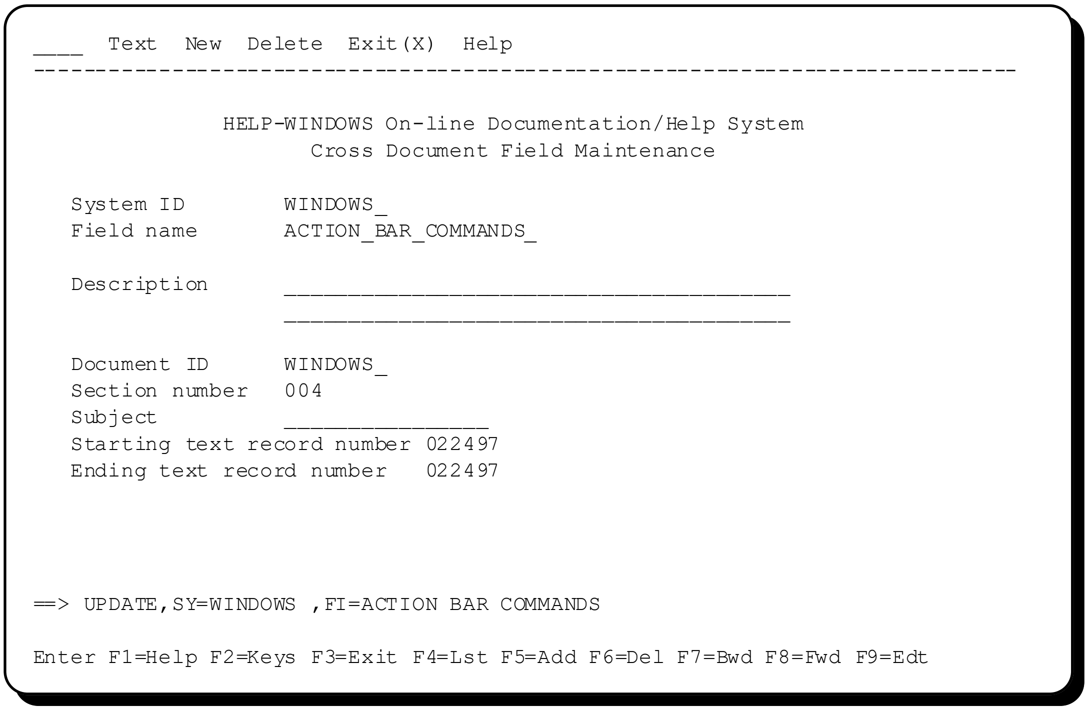
The functions available by PF keys and pull-down menus are as follows:

PF2 (Cpy)	-	Copy this transaction key record with new key information.
PF4 (Lst)	-	Display the associated help text in list mode.
PF5 (Add)	-	Add a new transaction key record.
PF6 (Del)	-	Delete this transaction key record.
PF7 (Bwd)	-	Browse backward
PF8 (Fwd)	-	Browse forward
PF9 (Edt)	-	Display the associated help text in edit mode.
PF11(Jugl)	-	Display the window as it will appear.

CROSS DOCUMENT RECORD DIRECT MAINTENANCE

The Field name is the same as the twenty-byte Field ID from the AUTODEF window and the name on the STRTFLD/STOPFLD statements in the text.

To retrieve a cross-document record for direct maintenance, display the Field directory by entering JHLP,FI or JHLP,SY=xxxxxxxx,FI=yyy...yyy, where xxxxxxxx is the SYSTEM name and yyy...yyy is from one to twenty characters of the Field ID. Locate the desired record and position the cursor by that entry, then press PF4 (update). The cross-document record will display.



```
____  Text  New  Delete  Exit(X)  Help
------------------------------------------------------------------------------

               HELP-WINDOWS On-line Documentation/Help System
                      Cross Document Field Maintenance

   System ID        WINDOWS_
   Field name       ACTION_BAR_COMMANDS_

   Description      ________________________________________
                    ________________________________________

   Document ID      WINDOWS_
   Section number   004
   Subject          ________________
   Starting text record number 022497
   Ending text record number   022497




==> UPDATE,SY=WINDOWS ,FI=ACTION BAR COMMANDS

Enter F1=Help F2=Keys F3=Exit F4=Lst F5=Add F6=Del F7=Bwd F8=Fwd F9=Edt
```

A Cross-document record is created automatically when a STRTFLD statement is entered in the text, using the identifier on the STRTFLD as the Field name. You cannot change this value.

The DESCRIPTION field is maintained for compatibility with the full HELP-WINDOWS product. You can enter a description here if you like, which will display on the Field directory.

DOCUMENT ID is maintained for compatibility with the full HELP-WINDOWS product. For cross-document records created by auto-define, the document ID must always be FASTPATH.

SECTION NUMBER is maintained for compatibility with the full HELP-WINDOWS product. For cross-document records created by auto-define, the section number must always be 001.

SUBJECT is maintained for compatibility with the full HELP-WINDOWS product. For cross-document records created by auto-define, the subject number must always be blank.

The STARTING TEXT NUMBER is the text record number of the first text record containing the STRTFLD that has the same name as Field Name in the cross-document record. You should not make changes to this field. The correct way to change the starting text number is to move the STRTFLD statement to another text record. By manipulating the STRTFLD, Help-Windows will always keep the cross-document records in sync. If for any reason the starting text number is incorrect, it can be directly changed here.

The ENDING TEXT NUMBER is the text record number of the last text record containing the STOPFLD that has the same name as Field Name in the cross-document record. You should not make changes to this field. The correct way to change the ending text number is to move the STOPFLD statement to another text record. By manipulating the STOPFLD, Help-Windows will always keep the cross-document records in sync. If for any reason the ending text number is incorrect, it can be directly changed here.

The Cross-document record is an automatic record that is created when a STRTFLD statement is first entered in text. It is the record that records the beginning and ending text record numbers of the help text for a given Field ID.

Maintenance of the Cross-document record (sometimes called XDOC), is fully automatic so that it is never necessary to display them, but when there is a question, or if for any reason the starting and ending numbers are incorrect, the record can be directly updated.

The functions available by PF keys and pull-down menus are as follows:

PF4 (Lst)	-	Display the associated help text in list mode.
PF5 (Add)	-	Add a new transaction key record.
PF6 (Del)	-	Delete this transaction key record.
PF7 (Bwd)	-	Browse backward
PF8 (Fwd)	-	Browse forward
PF9 (Edt)	-	Display the associated help text in edit mode.

RENUMBERING TEXT RECORDS

When enough text records have been added in one spot so that the increment between the last text record number and the one following is 'one', it is necessary to perform some reorganization to the text file, assuming more records need to be added at that spot.

There are two ways to make room for more text records:

1) Manually renumber one or more records.
   a) Add a new text record after the record which is blocking the way.
   b) Use Line Move commands to move all text from the previous record (the blocking record).
   c) Insert the moved text into the new record just added.
   d) The blocking record is now empty and can be used for new text.

2) Use the Renumber command.
   a) At any text record of the current SYSTEM, over type the EDIT command in the command line (==>) with the Renumber command. It can be abbreviated as RE, REN, RENU or RENUM. There must be a keyword SY=, DO= and SE= present with correct values.
   b) You will be prompted to press ENTER to perform a renumber, or exit with the Quit key (PF3).
   c) Upon pressing ENTER, the renumber operation will begin, which makes two passes of all the text in that system. The first pass counts the text records and scans the cross-document records for matching references. The second pass performs the reorganization of the text, which involves deleting and adding many records. A running pass counter and record counter appears at the bottom of the screen to report the progress of the renumber operation.

[Note] Be sure Dynamic Transaction Backout is active for the JHLP transaction. If the system comes down during a renumber, the text file can be corrupted.

USING THE WORD WRAP FEATURE

Word-wrap provides the ability to display help text which is wider than the window, but avoid requiring that the operator pan right and left to read it all. Word-wrap causes the text to break on word boundaries as it approaches the right edge of the window.

This is often useful, but sometimes the end result is undesirable for the following reasons:

1) Text wraps 'up' as well as 'down'. Thus, text from one paragraph may merge onto the end of a previous paragraph when that is not desired.

2) Text indentations are lost, so text that is formatted in columnar displays, for instance, loses its formatting when it displays.

These two problems can be overcome in most situations by using the word-wrap control commands in the editor. There are two commands which can be entered in the sub-command field (***) to the right of the text line to which they apply.

P (Paragraph end) Marks this text line as the last line of a paragraph so that subsequent text will not merge onto the line.

N (No wrap) Marks this text line as non-wrappable so that it will display exactly as it is keyed. This is often used when blank lines are left between paragraphs instead of a SPACE command. It also resolves the columnar data problem previously mentioned.

The word-wrap control commands will remain in the sub-command area unless they are removed by spacing over them. You may still enter other editor line commands (M, C, I, D, A, B) in the same field.

RESETTING UPPER CASE TRANSLATION

Upper case translation for the terminal is automatically set off any time the editor is entered so that help text may be keyed in lower case. It is restored when the JHLP transaction ends normally.

If an abend occurs, or a terminal error, or if for any reason the translation mode is not restored, a command is available which will reset it.

On a clear screen, hold the shift key down while you enter

```
JHLP,UC=ON
```

This will cause the upper case translation feature of the terminal to be restored so that normal operation can continue.

CREATING AND USING A TABLE OF CONTENTS

You can create a table of contents for an entire text system, which will display on-line from any help display. The table of contents is useful as a quick look-up for the operator to find the desired text, without constructing a lot of help menus. When the table of contents is displayed the operator can tab to the desired topic, and press ENTER to retrieve the help text for that subject.

Table of contents entries are keyed in the text as imbedded text commands, similar to STRTFLD/STOPFLD statements. The format is as follows:

```
<TOCx Text you want to display as a table of contents entry>
```

where 'x' is a 1, 2 or 3, designating the level of indention desired. TOC1 statements indent two positions, TOC2 statements indent by four, TOC3 will indent six positions. This allows you to specify topics and sub-topics.

The text of the TOCx command can be as long as the available space on the line, but remember to allow for window size and indentation. The table of contents will display in whatever window is in use when requested.

To display the table of contents in help mode, press PF11. The display will appear generally as follows:

```
            This line would be a TOC1 table of contents entry
            This line would be a TOC2 table of contents entry
            This line would be a TOC2 table of contents entry
            This line could be a TOC3 table of contents entry
            This line could be a TOC3 table of contents entry
            This line would be a TOC2 table of contents entry

    This line would be a TOC1 table of contents entry
```

Once the table of contents is displayed, the operator can press the TAB or RETURN key to position the cursor to the desired topic and press ENTER. The help text for that table of contents entry will display in the current window.

Table of Contents in Edit mode

In the editor, pressing PF11 will also display the table of contents, but in addition to any TOCx entries found, it will also display the Cross-Document Field names. These records are marked in the display as XDOC= to distinguish them from TOCx entries.

You can position the cursor to an XDOC entry and press ENTER to quickly move to the starting text for that field.

SEARCHING FOR TEXT STRINGS IN THE EDITOR

It is often desirable to locate the position of one or all occurrences of a text character string, which may occur in any text record of the system.

This can be done with the text search command, which is appended to the end of the EDIT command on the command line as follows:

```
EDIT,SY=system,DO=document,SE=section,REC=nnnnnn S=xxxx...xxxxxx
```

where all characters following the S= keyword become the search string. Do not enclose the string in quotes. If imbedded spaces are present, they will be included in the search criteria. Upper and lower case characters are considered the same in text searches. The maximum length of the search string is the number of characters from S= to the end of the line.

If the search string is found, the text record where it is found is displayed with all text lines highlighted which contain the search string. The command now changes to:

```
SEARCH,SY=system,DO=document,SE=section,REC=nnnnnn
```

and a message is displayed indicating to press ENTER to continue. The search will continue until the end of the document is reached, stopping at each text record where the search string is found, as long as ENTER is pressed.

To stop the search on any text record, change the SEARCH command back to EDIT. You can then make any desired changes to the text record.

To search backward in the text file, enter the search string as T=xx..xx rather than S=xx..xx. The search will then proceed from the current record number in a reverse direction until the beginning of the document is found.

IMPORTING HELP TEXT FROM ANOTHER SOURCE

If you have text in some other form that you wish to load into the text file for use by the Help-Windows feature of CICS-JUGGLER, that can be performed with the batch reformat program, JUGLREFM.

Use whatever means are available to get your text records into 80-byte card image form. For MVS, this can be a PDS member or a sequential file. For VSE it must be an unblocked SYSIPT file.

JUGLREFM supports a series of control statements that must appear at the beginning of your document. You can load several different documents in one pass, if desired, by placing the critical control statements at the appropriate point in the text record file.

The following describes the control statements and JCL for JUGLREFM.

Control statements input to JUGLREFM

&SYSTEM=xxxxxxxx
Denotes the following text as a new SYSTEM and establishes the system name. This may optionally be followed by a 79-byte system description, which must contain an ampersand in the first position.

&DOCUMENT=xxxxxxxx
Denotes the following text as a new DOCUMENT and establishes the document name. This may optionally be followed by a 79-byte description, which must contain an ampersand in the first position.

&SECTION=nnn
Denotes a new SECTION and sets the section number. May optionally be followed by a description record containing an ampersand in position one.

[Note] DOCUMENT and SECTION default to FASTPATH and 001 when text is added online. You may load text with different identifiers, however.

&CASE=UPPER|LOWER|MIXED
Sets the translation mode to upper case, lower case or mixed case. CASE=UPPER will force all text to upper. CASE=LOWER will force it to lower, CASE=MIXED leaves it untouched. Default is MIXED.

&REPLACE=YES|NO
Specifies if duplicate records already in the text file are to be deleted before loading this text. If NO, and a duplicate is found, a message is printed and the text record is skipped. REPLACE =YES should be used only to reload corrected data.

&SHIFTLEFT=FNB|nn|OFF
Shift all incoming text to the left to the first non-blank character or a specified number of positions. Can be used to eliminate text indentions.
&SHIFTLEFT=FNB justifies all text to position 1.
&SHIFTLEFT=nn shifts left nn positions.
&SHIFTLEFT=OFF performs no shifting.
The default is &SHIFTLEFT=FNB.

&START=nnnnn|1000
Sets the starting text record number. May be any free-form number from 1 to 99999. Default is 1000.

&INCRE=nnnnn|1000
Sets the text record increment factor, the value to be added to each record number. Default is 1000.

[Notes]

1) All parameters stay in effect until replaced by another parameter with the exception of &REPLACE, which resets to NO when a new &SYSTEM, &DOCUMENT or &SECTION keyword is found.

2) All parameters except &SYSTEM, &DOCUMENT and &SECTION may appear at any point in the text and take effect at that point.

The 80-byte text records may contain imbedded text commands. These are

<STRTFLD xxxxxxxxxxxxxxxxxxx>	Define start of FIELD xxx...xxx
<STOPFLD xxxxxxxxxxxxxxxxxxx>	Define end of FIELD xxx...xxx
<SPACE nn>	Insert nn blank lines.
<EJECT>	Force help panel break.

JCL required for the JUGLREFM program

MVS example:

```
//TEXTLOAD    JOB     N,ACCOUNT-ID,MSGCLASS=X
//STEP1       EXEC    PGM=JUGLREFM
//STEPLIB     DD      DSN=XXX.XXX.XXX,DISP=SHR
//HWN$FIL     DD      DSN=XXX.XXX.XXX,DISP=SHR          CICS-JUGGLER CONTROL FILE
//SYSPRINT    DD      SYSOUT=*
//SYSUDUMP    DD      SYSOUT=*
//SYSIN       DD      *
&SYSTEM=MYHELP
&This is a description of my system, which will appear in the directory.
&DOCUMENT=MYDOC
&SECTION=001
(80-BYTE CARD-IMAGE TEXT RECORDS)
/*
//
```

DOS/VSE example:

```
// JOB TEXTLOAD
// LIBDEF PHASE,SEARCH=LIBRARY.SUBLIBRARY
// DLBL HWN$FIL,'CICS-JUGGLER CONTROL FILE',,VSAM
// OPTION PARTDUMP
// EXEC JUGLREFM,SIZE=AUTO
&SYSTEM=MYHELP
&This is a description of my system, which will appear in the directory.
&DOCUMENT=MYDOC
&SECTION=001
(80-BYTE CARD-IMAGE TEXT RECORDS)
/*
/&
```

METHODS OF CREATING A HELP MENU

There are various ways to create menus of help displays. You can use any existing programming technique to build a menu or you can use HELP-WINDOWS to do it.

If you have an application development system or fourth-generation language package you could create a menu panel using the "paint-the-screen" technique available in most of these packages, then define a transaction key for a word which appears in each of the selection prompts of the menu screen.

You could accomplish the same thing with an application program using BMS mapping or any other screen-building technique.

For both of these techniques you should set an unprotected attribute code somewhere in the text of the selection prompt to allow the operator to tab the cursor to that position and press the help key. You then define a field-level transaction key for that position using either row/column specification or cursor-in-text specification.

You can also use HELP-WINDOWS to build a help menu through the use of nested help displays.

In general, you create a text record which will be your menu display, then define help definitions which are associated with a word in each of the selection prompts. You can use auto-define for these definitions but you will need to supply the transaction code of the dummy transaction (see *USING DUMMY TRANSACTIONS*, on the next page) by typing it at the main AUTODEF display. Each transaction key either references structured text or fast-path text to be associated with this menu selection. The text may be displayed in full-screen or window mode, at your discretion.

Now you must provide a method for the operator to get to this.

You could associate the help menu with some other user menu that the operator always sees, or you could associate it with a transaction screen. This would be accomplished through the normal means, that is, defining a help definition to be recognized on a user transaction or screen, with the text pointing to the help menu text record.

USING DUMMY TRANSACTIONS

Still another method is to use "dummy" transactions. This method provides a way to get directly to a help menu or directly to the text from a clear screen.

Since HELP-WINDOWS will recognize any text on the screen and can associate that with a help display, it is not necessary that the operator be in a real transaction in order to retrieve a help display. You can define a help definition and say that the transaction ID is anything, whether it is a real transaction or not.

For instance, you could define a transaction key with the transaction ID of HELP. HELP is not necessarily defined to CICS as a valid transaction code. From a clear screen, if you enter HELP, then press the help key (don't press ENTER), HELP-WINDOWS will retrieve and display the first page of text associated with that transaction. HELP-WINDOWS does not know or care whether it is a real transaction or not.

Your dummy transaction does not need to point to a menu, it could point directly to the help text. Thus, you could define a series of dummy transactions to invoke different on-line documentation displays. All the end-user needs to know in order to view the displays is what transaction to key.

HELP ACCESS MODE - USING ON-LINE HELP DISPLAYS

This topic deals with the use of the HELP-WINDOWS feature by the end user. That is, the retrieval of help displays while in a user transaction. This is called Help Access Mode, or simply 'help mode'.

This section should be used by the text-writer for a deeper understanding of the kinds of specifications that can be made when defining help displays that will produce the various results discussed herein.

Command entry in help mode, screen chaining, multiple window displays, nested help and the cut/paste feature are discussed.

ACTIVATION OF HELP ACCESS MODE

Before a help display can be retrieved and displayed on the screen, help access mode must be "activated" for the terminal. Help access mode is activated when CICS-JUGGLER is activated at the terminal, if the CICS-JUGGLER user profile has a HELP KEY defined.

If CICS-JUGGLER is not on, or if the profile does not establish a Help key, the end user will not be able to retrieve help displays.

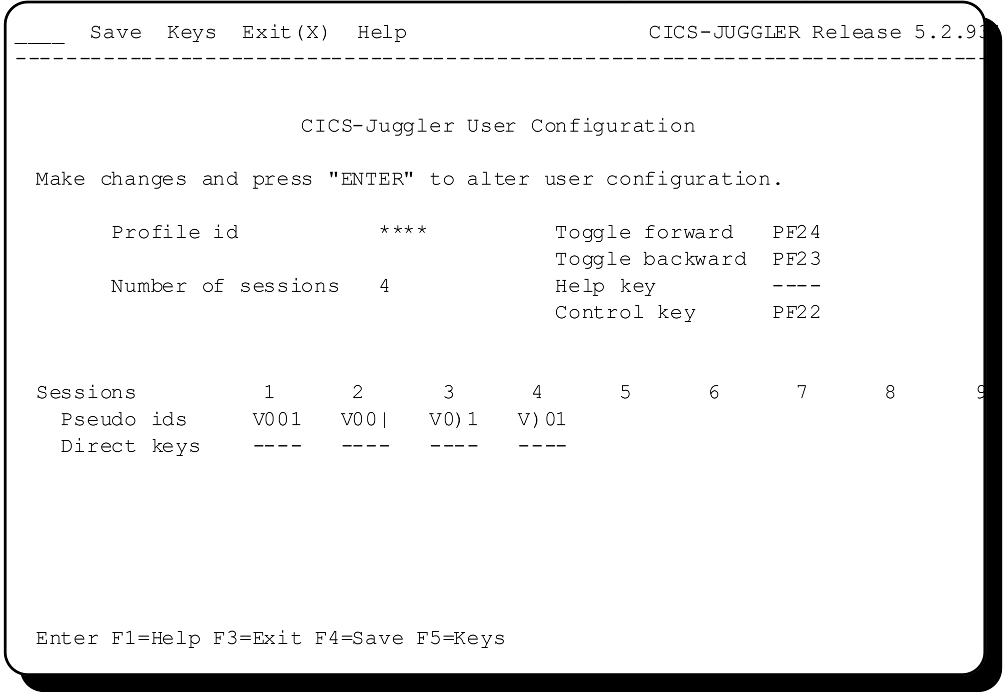
Activation of CICS-JUGGLER is performed with a JUGL,ON transaction, either entered directly at the terminal, or executed automatically using the automatic start-up feature of CICS-JUGGLER.

RETRIEVING HELP DISPLAYS

To retrieve a help display, first initiate a user transaction for which a help display has been defined. Then proceed to the appropriate screen (if not already there) and press your help key.

We will use the JNIT transaction, as it is available to all CICS-JUGGLER users.

First we enter the JNIT transaction with no operands, resulting in the following display:



```
____  Save  Keys  Exit(X)  Help                  CICS-JUGGLER Release 5.2.9
-----------------------------------------------------------------------------


                       CICS-Juggler User Configuration

 Make changes and press "ENTER" to alter user configuration.

       Profile id           ****          Toggle forward   PF24
                                          Toggle backward  PF23
       Number of sessions   4             Help key         ----
                                          Control key      PF22


 Sessions          1      2      3      4      5      6      7      8      9
   Pseudo ids     V001   V00|   V0)1   V)01
   Direct keys    ----   ----   ----   ----






 Enter F1=Help F3=Exit F4=Save F5=Keys
```

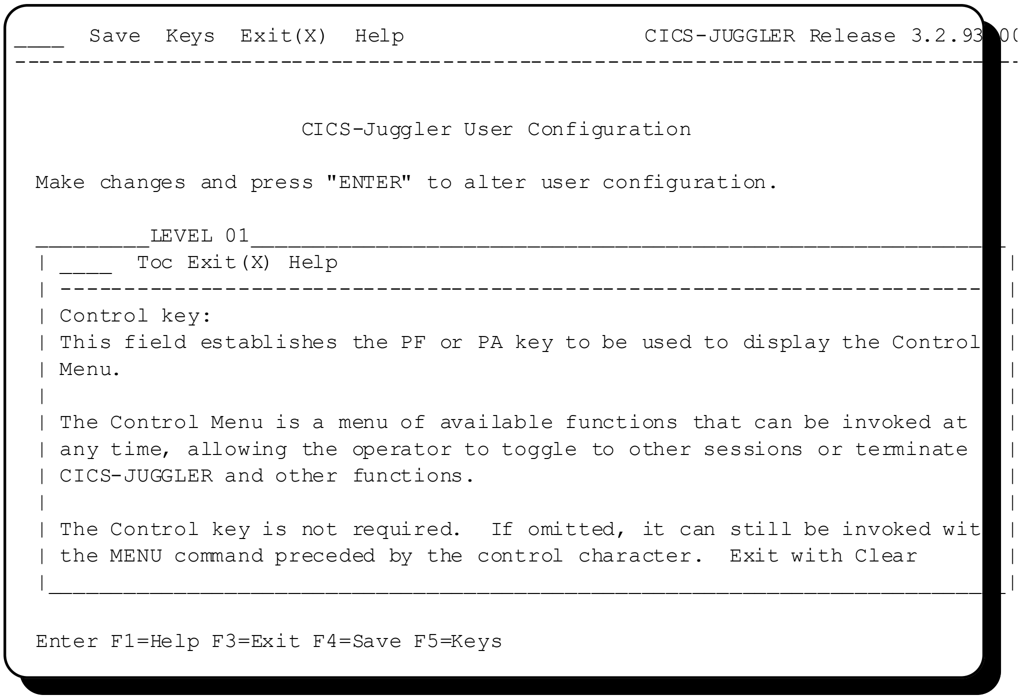
As distributed, this transaction has three types of help:

Field level help, which applies only to a certain field on the screen. This type of help is accessed by tabbing to the field in question and pressing the Help Access key.

Screen level help, which applies to the entire screen. This type of help is accessed by moving the cursor to an area on the screen for which no field level help is defined, (the title for instance). Screen level help may only be accessed if field level help is not available at the position of the cursor.

Transaction level help, which applies to the entire transaction. This type of help may only be accessed when the cursor is positioned to a field for which no help is defined on a screen for which no help is defined. This type is used for either of two reasons, for a fail-safe to ensure that help is always available, or for the only help display for very simple transactions.

Upon displaying this screen, tab to the Control Key entry field and press the Help Access key, the help text will display in a window as it was defined, as follows:



```
____  Save  Keys  Exit(X)  Help                   CICS-JUGGLER Release 3.2.93 0(
-------------------------------------------------------------------------------


                      CICS-Juggler User Configuration

 Make changes and press "ENTER" to alter user configuration.

 _________LEVEL 01____________________________________________________________
 | ____  Toc Exit(X) Help                                                     |
 | -------------------------------------------------------------------------- |
 | Control key:                                                               |
 | This field establishes the PF or PA key to be used to display the Control  |
 | Menu.                                                                      |
 |                                                                            |
 | The Control Menu is a menu of available functions that can be invoked at   |
 | any time, allowing the operator to toggle to other sessions or terminate   |
 | CICS-JUGGLER and other functions.                                          |
 |                                                                            |
 | The Control key is not required.  If omitted, it can still be invoked wit  |
 | the MENU command preceded by the control character.  Exit with Clear       |
 |____________________________________________________________________________|

 Enter F1=Help F3=Exit F4=Save F5=Keys
```

DIRECTIONAL OPTIONS AT A HELP DISPLAY

The last line of the window is used to display the directional options available at this point. You can invoke one of these options by pressing the corresponding PF key. The available options are:

PF2 - (KEYS) PF key prompts are displayed at the bottom of the window, however if the window is to narrow to display all available prompts, you may press PF2 to temporarily display the prompts in a vertical manner within the current window. Press Enter or PF3 to exit the Keys display, any other PF key will exit the keys display then perform the function shown in the Keys window.

PF3 - (EXIT) Use this key to exit from the current help display to the previous logical level. (This may be an earlier help display or the transaction is progress.)

PF4 - (CUT/PASTE) To use the cut/paste function, place the cursor anywhere in one of the words in the help display and press PF6. Upon pressing PF6, the help display will disappear and the application screen is re-displayed. The word where the cursor was placed is copied from the help display and placed into the field of the application display where the cursor was positioned when the help key was originally pressed.

PF5 - (LEFT) Move the display to the left one or more positions. Pressing PF5 will pan left one position. To pan more than one position, enter LE=nn (or LEFT=nn) in the command area, where nn is the number of positions to pan.

Note that panning left means that you logically move the window to the left, over the text. It will appear as though the text moves to the right.

PF6 - (RIGHT) Move the display to the right one or more positions. Pressing PF6 will pan right one position. To pan more than one position, enter RI=nn (or RIGHT=nn) in the command area, where nn is the number of positions to pan.

Note that panning right means that you logically move the window to the right, over the text. It will appear as though the text moves to the left.

[Note] If word-wrap is in effect for this display, there is no need to perform any panning. Word-wrap will cause sentences of text to break on word boundaries so that all of the text will display, no matter what size the window. With word-wrap, you only need to page forward to read all of the text.

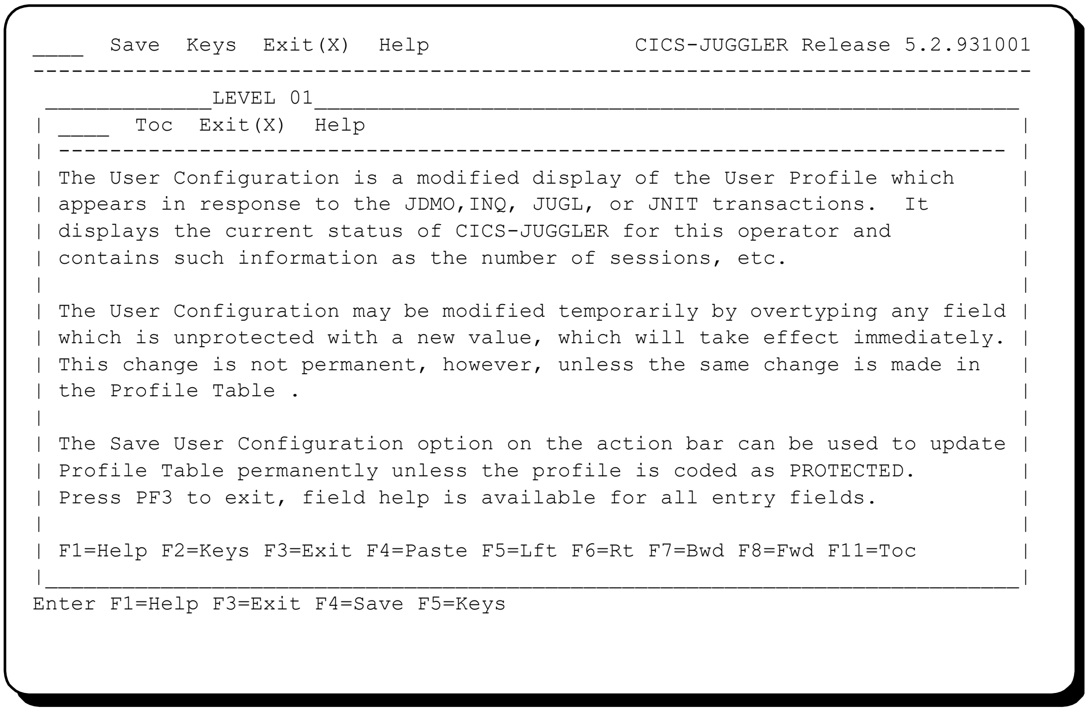
PF7 - (PREVIOUS) Display the previous page of text before this one. The display will move backward in the text for the depth of the window, displaying as much text as can be displayed in the window.

PF8 - (NEXT) Display the next page of text after this one. The display will move forward in the text for the depth of the window, displaying as much text as can be displayed in the window.

PF11 - (TOC) Display the Help Table of Contents.

RETRIEVING A SCREEN-LEVEL HELP DISPLAY

The previous example illustrated the retrieval of a field-level help display. That particular display was retrieved because there was a field-level display defined. If the cursor was not positioned to a 'tabbable' field at the time the Help Key was pressed (the screen title for instance), the following screen-level help would display:



```
____  Save  Keys  Exit(X)  Help                CICS-JUGGLER Release 5.2.931001
-----------------------------------------------------------------------------
 ____________LEVEL 01___________________________________________________________
| ____  Toc  Exit(X)  Help                                                     |
| -------------------------------------------------------------------------- |
| The User Configuration is a modified display of the User Profile which     |
| appears in response to the JDMO,INQ, JUGL, or JNIT transactions.  It       |
| displays the current status of CICS-JUGGLER for this operator and          |
| contains such information as the number of sessions, etc.                  |
|                                                                             |
| The User Configuration may be modified temporarily by overtyping any field |
| which is unprotected with a new value, which will take effect immediately. |
| This change is not permanent, however, unless the same change is made in   |
| the Profile Table .                                                         |
|                                                                             |
| The Save User Configuration option on the action bar can be used to update |
| Profile Table permanently unless the profile is coded as PROTECTED.        |
| Press PF3 to exit, field help is available for all entry fields.           |
|                                                                             |
| F1=Help F2=Keys F3=Exit F4=Paste F5=Lft F6=Rt F7=Bwd F8=Fwd F11=Toc        |
|_____________________________________________________________________________|
Enter F1=Help F3=Exit F4=Save F5=Keys
```

When the help key is pressed, HELP-WINDOWS will determine that there is not a field-level help display defined for the current cursor position, however there is a screen-level display for a screen with the JNIT transaction code and the word "CONFIGUR" at row 3, column 40. The screen-level text will be retrieved and displayed.

The screen-level help is displayed in the window size and position that was defined. All of the same directional options previously explained are available at this point.

RETRIEVING A TRANSACTION-LEVEL HELP DISPLAY

As previously stated, transaction level help (if defined) will display only if no field or screen level help is defined. As distributed, the screens of the JNIT transaction have both field and screen level help, so transaction level help should never display. However, the directional options are the same for transaction level help.

CHAINED HELP WINDOWS

Chained help windows occur when multiple transaction key records are defined with the same transaction code, screen identifier and location, with incrementing sequence numbers.

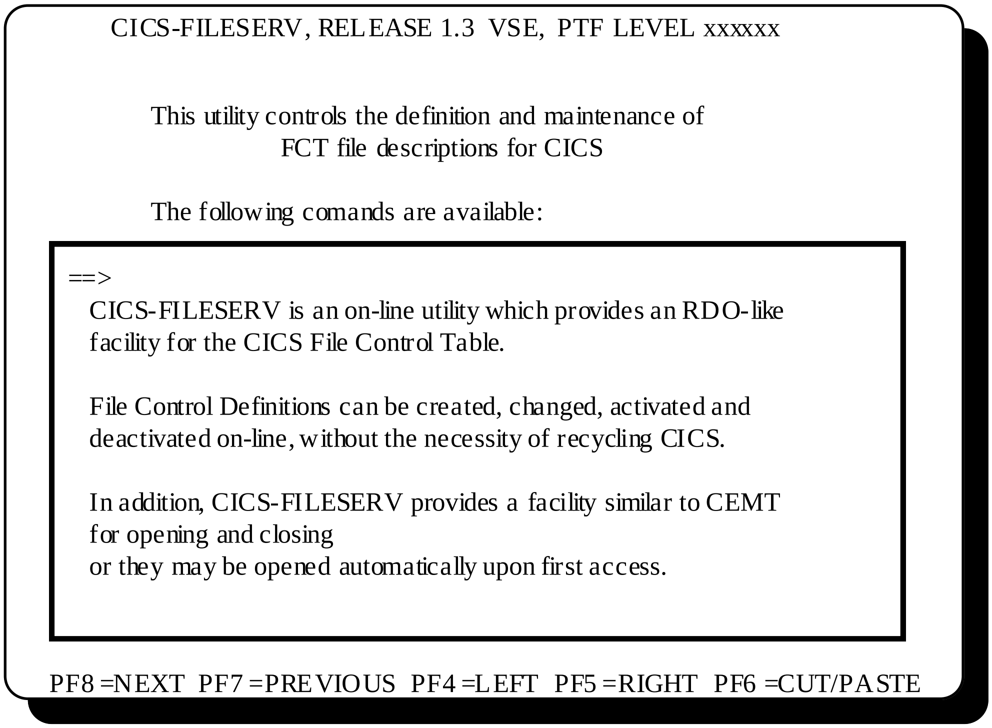
In this situation, each transaction key references a different section of text to be displayed, and may define a different window configuration.

In help access mode, the text associated with the first transaction key will be retrieved when the help key is pressed. Upon browsing forward with the NEXT command or key, all of the text associated with the first transaction key is displayed. When the end of text is reached for the first definition, another NEXT command will cause the second transaction key record to be retrieved, whereupon the text and window configuration for that definition is displayed.

If the window configuration of the second transaction key is the same as the first, you cannot tell that chaining has occurred. If however, the second key has a different size and/or position for the window, you will see a different display.

One of the options for this type of operation is the REFRESH SCREEN option of the transaction key record. If this field is specified as YES, the window of the previous transaction key will disappear when chaining occurs and the new window is displayed. If REFRESH SCREEN is specified as NO, the first window remains on the screen and the second window is displayed in addition to, and sometimes on top of it.

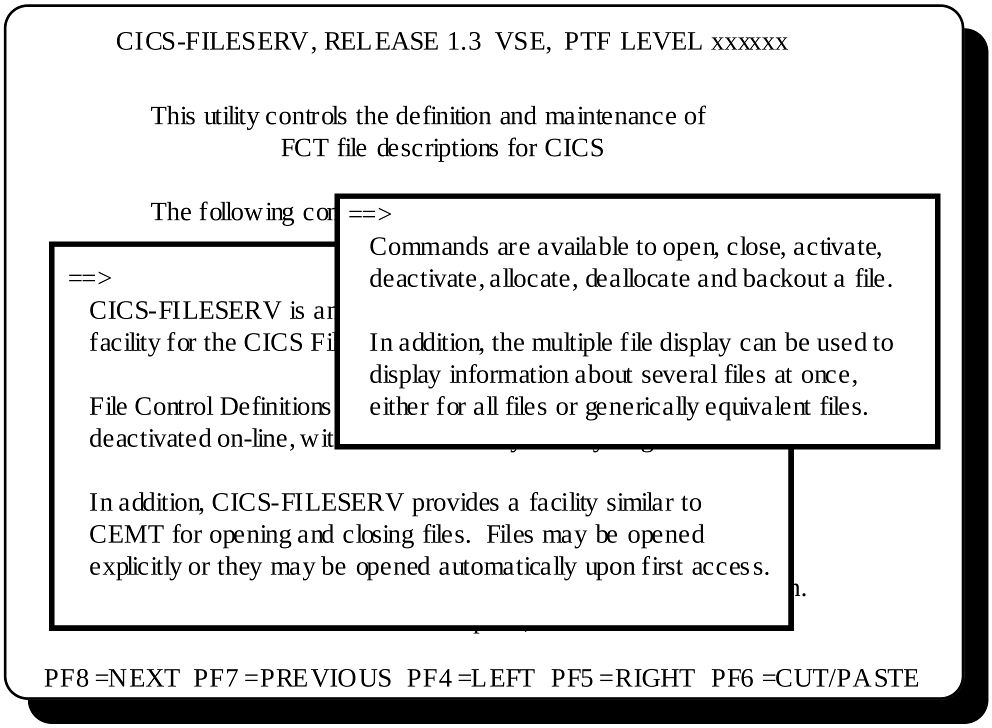
The result is that multiple windows can appear on the screen at one time, each with its associated text. An example of multiple window chaining could appear as follows.
First we have the application screen with a single window displayed:



```
        CICS-FILESERV, RELEASE 1.3  VSE,  PTF LEVEL xxxxxx


           This utility controls the definition and maintenance of
                      FCT file descriptions for CICS

           The following comands are available:

  ==>
    CICS-FILESERV is an on-line utility which provides an RDO-like
    facility for the CICS File Control Table.

    File Control Definitions can be created, changed, activated and
    deactivated on-line, without the necessity of recycling CICS.

    In addition, CICS-FILESERV provides a facility similar to CEMT
    for opening and closing
    or they may be opened automatically upon first access.

  PF8 =NEXT  PF7 =PREVIOUS  PF4 =LEFT  PF5 =RIGHT  PF6 =CUT/PASTE
```

Now we browse forward with PF8. If this is the end of text for the first transaction key, and there is another record with the same transaction code, screen ID and location but with a different

sequence number, you would see still a different display. The next display might appear as follows:



CICS-FILESERV, RELEASE 1.3 VSE, PTF LEVEL xxxxxx

This utility controls the definition and maintenance of
FCT file descriptions for CICS

The following co

==>
Commands are available to open, close, activate,
deactivate, allocate, deallocate and backout a file.

In addition, the multiple file display can be used to
display information about several files at once,
either for all files or generically equivalent files.

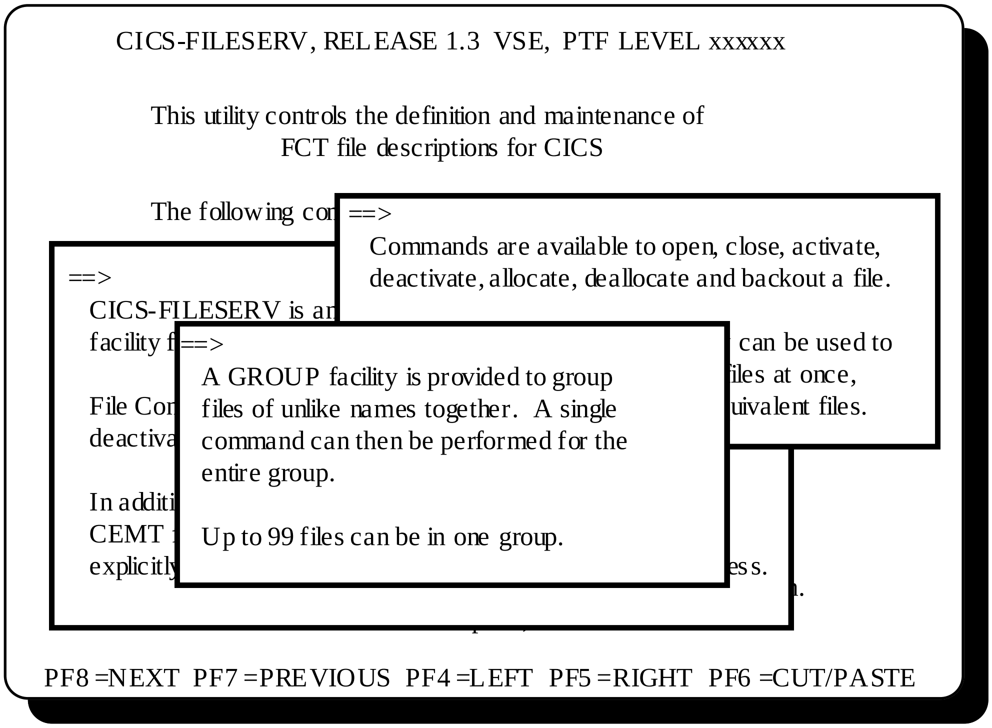
==>
CICS-FILESERV is an
facility for the CICS Fi

File Control Definitions
deactivated on-line, wit

In addition, CICS-FILESERV provides a facility similar to
CEMT for opening and closing files. Files may be opened
explicitly or they may be opened automatically upon first access.

PF8 =NEXT PF7 =PREVIOUS PF4 =LEFT PF5 =RIGHT PF6 =CUT/PASTE

Continuing forward, we might encounter still another transaction key which configures the display as follows:



CICS-FILESERV, RELEASE 1.3 VSE, PTF LEVEL xxxxxx

This utility controls the definition and maintenance of
FCT file descriptions for CICS

The following co

==>
Commands are available to open, close, activate,
deactivate, allocate, deallocate and backout a file.

can be used to
iles at once,
uivalent files.

==>
CICS-FILESERV is an
facility f

File Con
deactiva

In additi
CEMT
explicitly

==>
A GROUP facility is provided to group
files of unlike names together. A single
command can then be performed for the
entire group.

Up to 99 files can be in one group.

PF8 =NEXT PF7 =PREVIOUS PF4 =LEFT PF5 =RIGHT PF6 =CUT/PASTE

The window which contains the cursor is the current active window and is the one which responds to any commands or PF key directional options.

Browsing backward from this point with the PREVIOUS command or key will remove the windows one at a time as the beginning of text for each window is passed.

Also, pressing PF3 will revert back to the previous window. Continued pressing of the help key will take you all the way back to the application display with no windows. Pressing the CLEAR key will take you directly back to the application screen, no matter how many windows are on the screen.

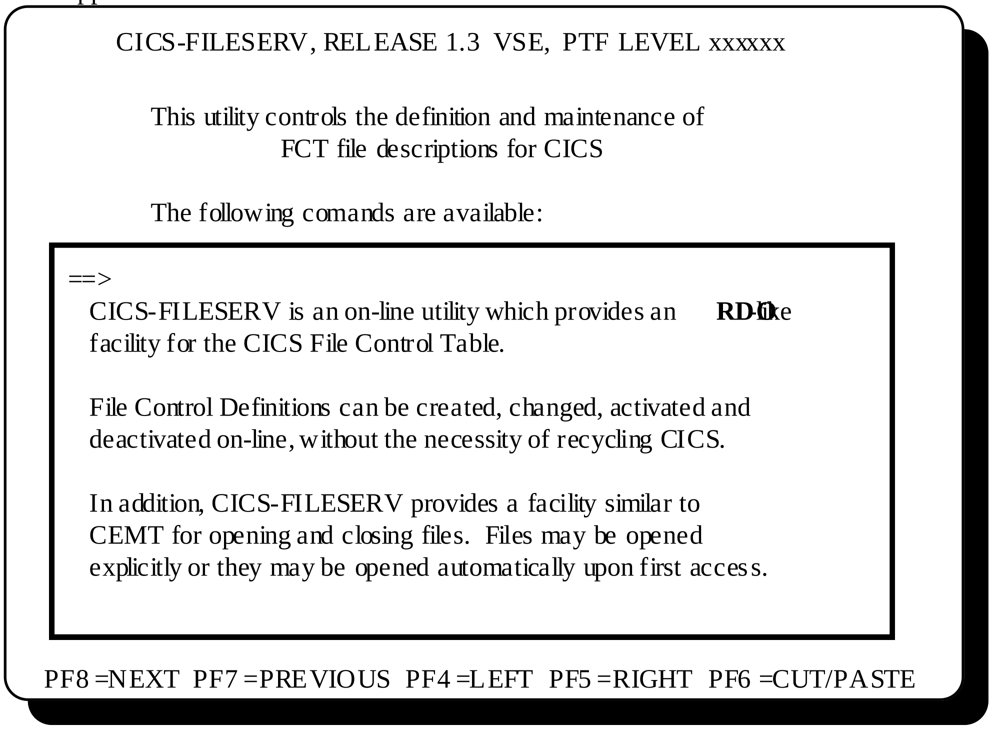
HELP DISPLAYS WITHIN HELP TEXT (NESTED HELP)

It is possible to define a help screen or help window, then define another help display to be keyed to one of the words of text in the first display. This is called 'nested' help.

The definition for this is accomplished by coding the word in the first display with an unprotected attribute so that the operator can tab to it (it should probably also be high intensity or a different color so as to be noticed), then defining a transaction key using that word as the field text identifier. For this definition, CURSOR IN FIELD TEXT should be specified as YES.

To retrieve the nested help display while in help mode, tab to the field, then press the ENTER key instead of the help key.
An example of this would be as follows. On the first display, the word RDO might be set up with a transaction key using 'RDO' as a field text identifier. RDO is coded to be high-intensity, unprotected.

The appears as follows:



CICS-FILESERV, RELEASE 1.3 VSE, PTF LEVEL xxxxxx

This utility controls the definition and maintenance of
FCT file descriptions for CICS

The following comands are available:

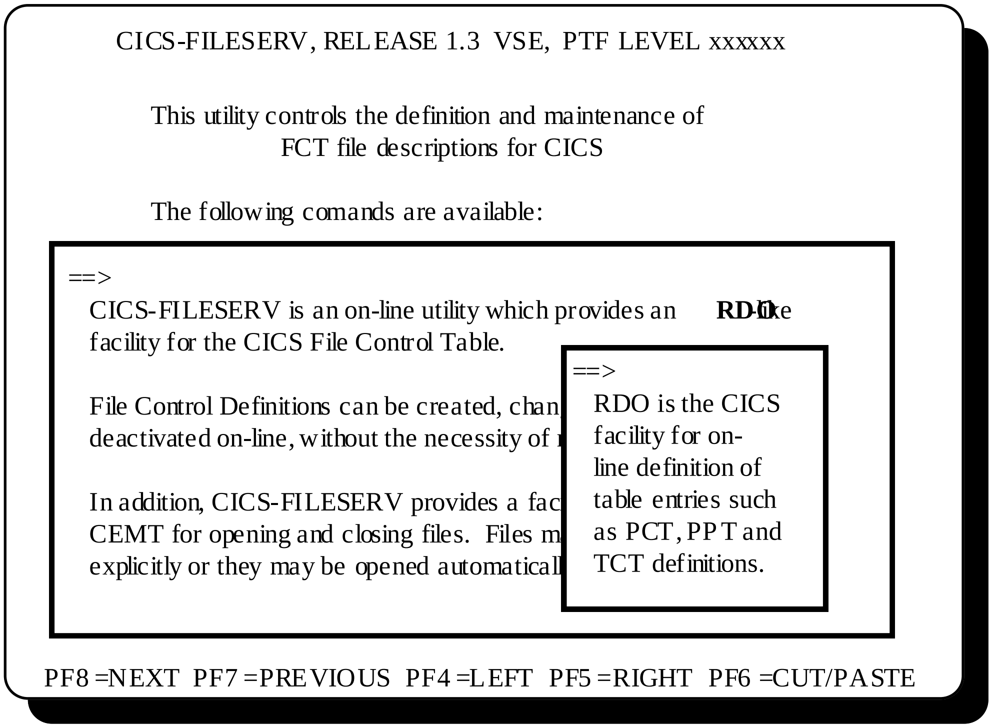
==>
CICS-FILESERV is an on-line utility which provides an **RDO**-like
facility for the CICS File Control Table.

File Control Definitions can be created, changed, activated and
deactivated on-line, without the necessity of recycling CICS.

In addition, CICS-FILESERV provides a facility similar to
CEMT for opening and closing files. Files may be opened
explicitly or they may be opened automatically upon first access.

PF8 =NEXT PF7 =PREVIOUS PF4 =LEFT PF5 =RIGHT PF6 =CUT/PASTE

If the operator tabs to the RDO field and presses ENTER, the next display could appear as:



CICS-FILESERV, RELEASE 1.3 VSE, PTF LEVEL xxxxxx

This utility controls the definition and maintenance of
FCT file descriptions for CICS

The following comands are available:

==>
CICS-FILESERV is an on-line utility which provides an **RDO**-like
facility for the CICS File Control Table.

File Control Definitions can be created, chan
deactivated on-line, without the necessity of

In addition, CICS-FILESERV provides a fac
CEMT for opening and closing files. Files m
explicitly or they may be opened automaticall

==>
RDO is the CICS
facility for on-
line definition of
table entries such
as PCT, PPT and
TCT definitions.

PF8 =NEXT PF7 =PREVIOUS PF4 =LEFT PF5 =RIGHT PF6 =CUT/PASTE

The nested window could, of course appear anywhere on the screen, or it could be a full-screen display.

Nested help windows work in the same manner as chained windows. That is, pressing PF3 causes the current window to disappear and reverts back to the previous window in the nest.

EXITING FROM A HELP DISPLAY

When you are ready to exit a help display and return to transaction operation, there are two methods of doing so:

1) Pressing PF3 will remove the current window or help screen and return to the previous level. If only one window or screen is present, this will return you to the application screen.

   If multiple help displays are present, either because of screen chaining or nested help displays, pressing PF3 takes you back to the previous display. Pressing it again takes you back another level, etc.

2) Pressing the CLEAR key always returns you to the application screen, regardless of how many help displays are concurrently active.

Once you return to the application screen you may continue with the next input to that application, just as if you had not interrupted it. The cursor is replaced in whatever position it was in when you first pressed the help key and all attributes are restored to their original value.

USING THE CUT/PASTE FEATURE

The cut and paste function is a special feature of HELP-WINDOWS which allows the operator to copy text from the help display into the application screen.

A good use of this feature is when the text in the help window describes all of the valid codes for a given field on the application screen. The operator can place the cursor in the field, press the help key to see all of the valid codes, then, rather than exiting help mode and keying the code in the field, simply place the cursor on the desired code and press the cut/paste key. This will cause the word where the cursor was placed to be copied from the help text and placed into the application field where the cursor was positioned when the help key was originally pressed.

When the word is pasted into the application field, the modified-data-tag for the field is turned on, as it would be if the operator had keyed the word. Thus, when ENTER or a PF key is pressed to the application, the pasted word will transmit to the application program just as if it were keyed.

When defining a help display, you do not need to do anything to enable the cut/paste feature to be used. It will work with any help display. It may be helpful, however, to place unprotected attributes before each word or value which would logically be eligible for a cut/paste operation, thereby allowing the operator to use the TAB key to easily position to the desired entry.

The following limitations and rules apply to the cut/paste feature of HELP-WINDOWS:

1) You can only paste into an unprotected field of the application display.

2) Only one word will be copied when the cut/paste key is pressed. The cursor may be anywhere within that word. If multiple-word copy is desired you must connect the words with a hyphen or underscore or some other character so that HELP-WINDOWS will consider it to be all one word.

3) If the word being pasted is longer than the receiving field, characters will be truncated on the right until the word fits the field.

4) If the word being pasted is shorter than the receiving field, no additional characters are padded on the end.

THE MENU GENERATION FEATURE

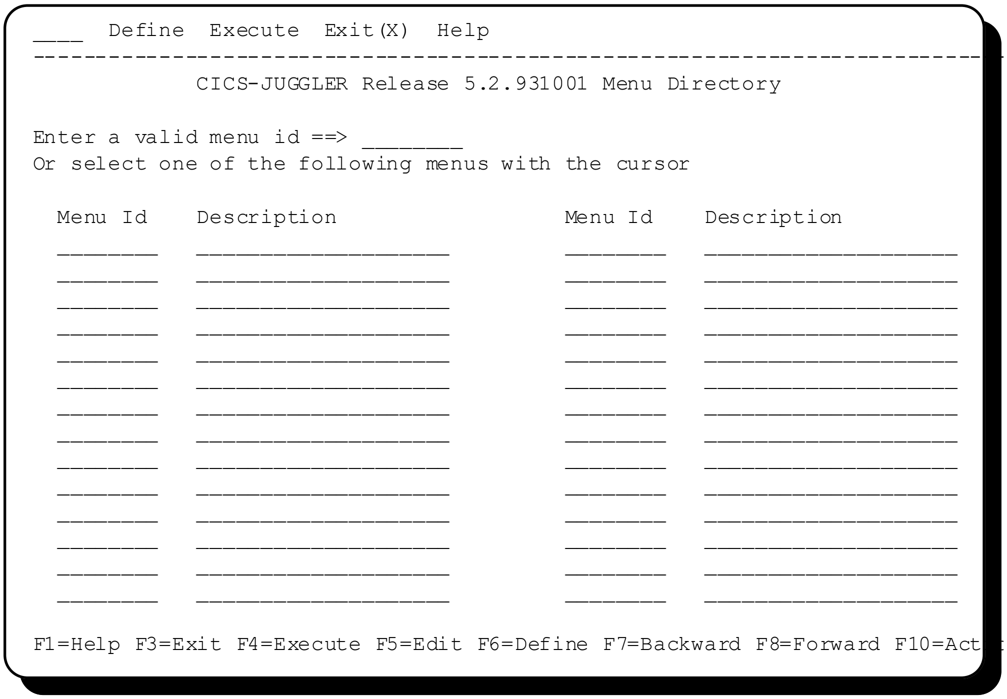
An optionally licensed feature of CICS-JUGGLER is a facility that enables users to create menus to be used by operators. A menu can be configured such that it can invoke transactions, programs and/or other menus.

A demonstration menu system is distributed on the installation tape. This system provides examples of methods of coding menus and operation of menus. To invoke the demonstration menu, from a clear screen enter **JMNU,DEMOMENU**.

Menu generation is accessed through the transaction code - **JMNU**, which is explained in more detail in the following discussion.

THE MENU DIRECTORY DISPLAY

To invoke menu generation, enter the transaction code **JMNU**. Upon pressing ENTER, the Menu Directory display will appear as follows:



```
____  Define  Execute  Exit(X)  Help
-----------------------------------------------------------------------------
             CICS-JUGGLER Release 5.2.931001 Menu Directory

Enter a valid menu id ==> ________
Or select one of the following menus with the cursor

  Menu Id    Description                   Menu Id    Description
  ________   ____________________          ________   ____________________
  ________   ____________________          ________   ____________________
  ________   ____________________          ________   ____________________
  ________   ____________________          ________   ____________________
  ________   ____________________          ________   ____________________
  ________   ____________________          ________   ____________________
  ________   ____________________          ________   ____________________
  ________   ____________________          ________   ____________________
  ________   ____________________          ________   ____________________
  ________   ____________________          ________   ____________________
  ________   ____________________          ________   ____________________
  ________   ____________________          ________   ____________________
  ________   ____________________          ________   ____________________
  ________   ____________________          ________   ____________________

F1=Help F3=Exit F4=Execute F5=Edit F6=Define F7=Backward F8=Forward F10=Act
```

FIELDS OF THE MENU DIRECTORY DISPLAY

==> This is the menu ID field and is used to access a menu that may or may not be displayed on this screen. This field is used by keying the menu ID, then pressing the desired PF key; or tabbing the cursor to the function in the function key area, and pressing ENTER.

The columns of fields below this point make up the directory of available menus. All menus on file are displayed in alphabetic order. Each column entry contains two fields:

DESCRIPTION This is a brief description of the menu.

MENU ID This is the name of the menu. Names are assigned to each menu definition for ease of identification. These fields are unprotected so that you may tab to the desired menu then press the PF key of the function that you wish to invoke. You can not change a menu name with this field.

ENTERING COMMANDS AT THE MENU DIRECTORY DISPLAY

At the bottom of the screen is a list of available functions that may be invoked. You may select the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

PF1 HELP Display a Help screen for information pertaining to the directory.

PF3 EXIT Exit the Menu Generation transaction.

PF4 EXECUTE This is used to execute (invoke) the specified menu.

PF5 EDIT This is used to view and optionally modify the actual text of the specified menu.

PF6 DEFINE Display the Menu Definition of the specified menu.

PF7 BACKWARD	Browse backward to the previous page of menu entries (if any).
PF8 FORWARD	Browse forward to the next page of menu entries (if any).

DEFINING AND MAINTAINING MENUS

ADDING A NEW MENU

To add a new menu, from the Menu Definition screen, activate the NEW pull-down menu, then select either ADD or COPY. ADD will create a new menu with all fields blank. COPY will create a new menu with the same specifications and menu text as the currently displayed menu.

Once ADD or COPY is selected, the screen will display the new menu definition, requesting the menu ID that is to be assigned to the new definition. Key a one to eight digit Menu ID and press ENTER to create the new menu definition.

Once created, you may code the fields as described in *THE MENU DEFINITION DISPLAY*, later in this section.

CHANGING AN EXISTING MENU

To change an already existing menu definition, from the Menu Directory display, select the menu definition that you wish to change, and press the DEFINE key.

The menu definition will then be displayed. To alter the definition, key the changes and press ENTER.

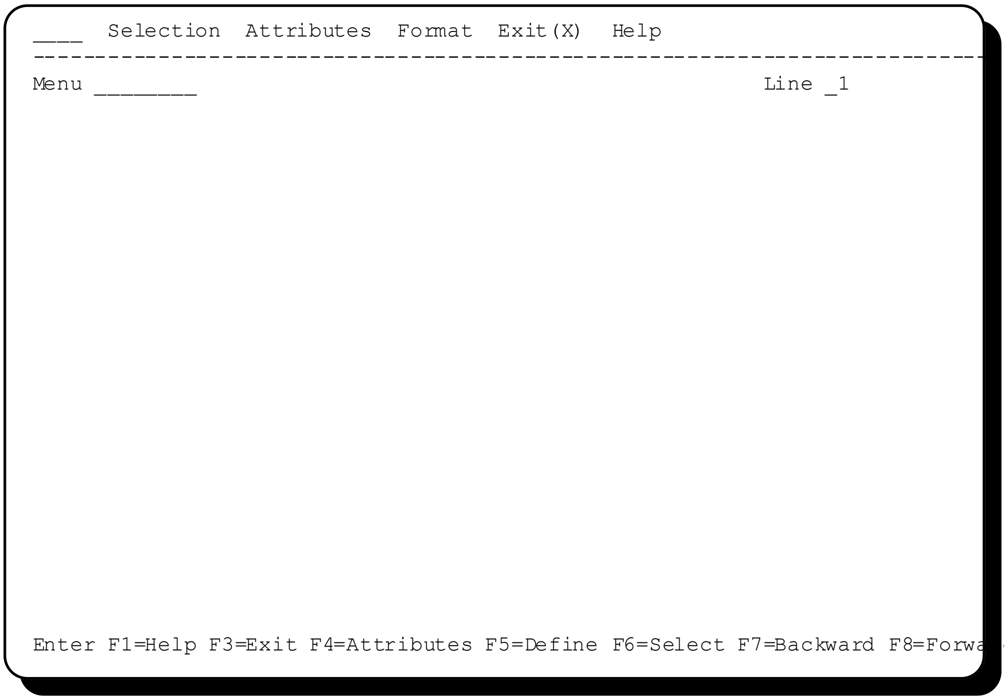
DELETING A MENU

To delete a menu definition, from the Menu Directory display, select the menu definition that you wish to delete, and press the DEFINE key.

The menu definition will then be displayed. To delete the definition, activate the DELETE pull-down menu, and select DELETE.

THE MENU TEXT EDITOR

This is used to 'paint the screen' the way you want the menu to be displayed. This screen appears when attempting to define or modify a menu by pressing the EDIT key from the Menu Directory or Menu Definition displays. This screen appears as follows:



```
____  Selection  Attributes  Format  Exit(X)  Help
---------------------------------------------------------------------------
Menu ________                                          Line _1












Enter F1=Help F3=Exit F4=Attributes F5=Define F6=Select F7=Backward F8=Forwa
```

USING THE MENU TEXT EDITOR

To enter text, simply key lines of data, formatting it as desired so far as line spacing, indentation, etc. are concerned. If changes are needed after keying some text, you can simply move the cursor to the area to be changed and type over any existing data. You can use the DEL key to delete characters in a line. To insert characters on a line, position the cursor where the insertion is to be made, press the INS key and type the desired characters (note that you may need to turn NULLS on, which is explained later in this topic). The ERASE EOF key may be used to erase all or part of a line at any time.

When all text for this page has been keyed, press ENTER. This will write the updated text record to file. If the text being entered cannot fit on one screen, you may press the FORWARD key (or change the line number at the top of the display).

LINE DELETION

To delete an entire line, TAB to the COMMAND FIELD at the end of the line (where the asterisks are) and enter a 'D', then press ENTER. The entire line will be deleted and all following lines will shift up one line.

To delete more than one line at a time, enter a 'D' in more than one COMMAND FIELD before pressing ENTER or enter 'Dnn' on one or more COMMAND FIELDS, where nn is the number of consecutive lines (including the current line) to be deleted.

You may also perform block deletion by placing 'DD' on the first line to be deleted and another 'DD' on the last line to be deleted.

LINE INSERTION

To insert a line between two text lines, tab to the COMMAND FIELD of the preceding line and enter an 'I' over any one of the asterisks, then press ENTER. This will cause a blank line to be inserted following that line, and all subsequent lines will shift down one line. You may then key any desired text on that line.

To insert multiple lines, enter 'Inn' in the preceding COMMAND FIELD, where nn is the number of lines to insert, then press ENTER. This will cause nn blank lines to be inserted after the current line, and all subsequent lines will shift down nn lines.

MOVING AND COPYING TEXT LINES

One or more consecutive text lines can be moved or copied to another location. To move text lines, enter 'M' or 'Mnn' (nn = number of lines to move) in a COMMAND FIELD and press ENTER. This causes the specified line(s) to be deleted from that location. Now place an 'A' (insert-after) in the COMMAND FIELD on the line preceding the new location, or a 'B' (insert-before) on the line following the new location and press ENTER. The moved lines will be inserted following that line if the 'A' command is used, or before that line for the 'B' command.

To copy text lines, enter 'C' or 'Cnn' (nn = number of lines to copy) in a COMMAND FIELD and press ENTER. Now place an 'A' (insert-after) in the COMMAND FIELD on the line preceding the new location, or a 'B' (insert-before) on the line following the new location and press ENTER.

You may also perform block moves and copies by placing 'MM' or 'CC' on the first line to be moved or copied and another 'MM' or 'CC' on the last line to be moved/copied.

You may enter more than one 'M' or 'C' command on one edit screen, if desired. The result is that all selected lines are grouped together consecutively and inserted as one set wherever the 'A' or 'B' command is placed. For instance, if 'C2' is entered in the command area at line 3, then 'C4' is entered in the command area at line 16, all six text lines will be grouped together and inserted as one set following the line where the 'A' command is placed.

Note that you may use the 'A' or 'B' sub-commands as many times as desired, without performing another 'M' or 'C' sub-command. Each time it is used, the same data that was moved or copied last will be inserted at the new location.

CREATING COLOR AND HIGHLIGHTING

Unless otherwise specified, all words on the menu will be displayed in low intensity protected. In order to emphasize certain words or lines, you can use the ATTRIBUTES function.

To invoke the Attribute function, press the ATTRIBUTES key (shown in the PF key prompt area). This places the terminal in SET COLOR mode. Note that upon invoking the attribute function, the function key area changes to include additional functions. These additional functions are as follows:

ENTER — If the cursor is inside the attribute window, pressing ENTER will move the cursor to the current attribute pointer position.

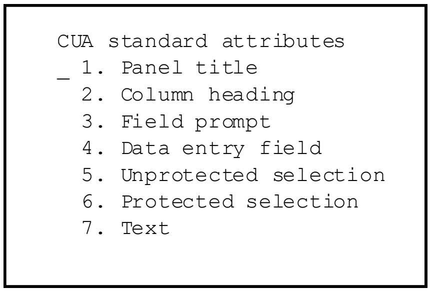
F2=SHOW — Show all attributes on the screen as an at-sign (@).

F3=EXIT — Exit Set Attribute mode to the editor.

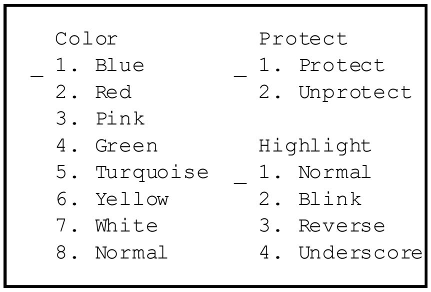
F4=SET — Set an attribute (as specified in the attribute window) at the current cursor position.

F5=SELECT Select the attribute at the position of the cursor and display the type of attribute in the attribute window. (If the cursor is inside the attribute window, this will move the window out of the way.)
F6=REMOVE Remove (delete) the attribute at the position of the cursor.
F7=BWD Move the current attribute pointer to the previous attribute.
F8=FWD Move the current attribute pointer to the next attribute.
F9=EXTEND Toggle the attribute window between CUA and extended attributes.

When the attributes function is invoked, a pop-up window appears. The attributes window can appear in two forms:



```
  CUA standard attributes
  _ 1. Panel title
    2. Column heading
    3. Field prompt
    4. Data entry field
    5. Unprotected selection
    6. Protected selection
    7. Text
```



```
    Color              Protect
  _ 1. Blue          _ 1. Protect
    2. Red             2. Unprotect
    3. Pink
    4. Green           Highlight
    5. Turquoise     _ 1. Normal
    6. Yellow          2. Blink
    7. White           3. Reverse
    8. Normal          4. Underscore
```

- A CUA standards attribute window. This window contains the CUA attributes for creating a display. The following chart portrays the display attributes for the various selections:

Selection	Extended color	Intensity	Protect/unprotect
Panel title	Yellow	High	Protected
Column Heading	Turquoise	High	Protected
Field Prompt	Turquoise	Normal	Protected
Data Entry	Green	Normal	Unprotected
Unprotected Selection	White	Normal	Unprotected
Protected Selection	White	Normal	Protected
Text	Blue	Normal	Protected

- An extended attribute window. This window offers all attribute combinations.

If the CUA standard window is displayed and you wish to use the extended attribute window, press the EXTEND key (shown in the PF key area). This key works as a toggle between the two types of attribute windows.

Upon pressing the ATTRIBUTES key, either a field mark or an asterisk with an overscore will display somewhere on the screen. This is the Current Attribute Pointer. If an attribute is present at that position of the screen, it will appear as a field mark, else it will appear as the overscored asterisk. You may press the FORWARD or BACKWARD keys to position the pointer to the next or previous attributes on the screen, or you may position the cursor to a suspected location of an attribute and press the SELECT key to move the attribute pointer directly to that position.

To set an attribute, simply select the choices within the attribute window by entering the number of the choice. Then position the cursor to a space before the word (or line) that is to contain the attribute features. Then press the SET key (shown in the function key area). The attribute will remain in effect until another attribute or the end of the line is encountered.

To remove an attribute, position the cursor to the attribute to be removed and press the REMOVE key (shown in the PF key area).

[Note] Setting color with either attributes windows may cause extended attributes to be generated in the screen display when viewed. You must be using a terminal that supports extended attributes and the COLOR and/or HILIGHT feature(s) must be set in the CICS terminal control table (TCT) entry (or EXTENDEDDS with RDO) in order to view them. If you view a display on a terminal that does not support extended datastreams, the extended color and highlighting attributes will be removed (in favor of regular field attributes) when the screen is displayed.

THE SELECT WINDOW

While in the Menu Editor, you may view and alter the selection information of a Menu field on the screen by moving the cursor to the field and pressing the SELECT key (shown in the PF key prompt area).

Upon pressing the SELECT key, a popup window will appear showing the selection information of the field. To view the selection information of another field, you may press the FORWARD or BACKWARD keys, or change the SELECTION NUMBER in the popup window and press ENTER.

The information shown in the popup window directly corresponds to the information shown on the Menu Definition Display. For a more detailed description of these fields, see *THE MENU DEFINITION DISPLAY*, earlier in this section.

ENTERING COMMANDS AT THE MENU TEXT EDITOR

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key.

ENTER Update the menu or message and/or apply any commands in the command field (to the right of the screen).

PF1 HELP Display a Help screen for information pertaining to the text editor.

PF3 EXIT Exit to the Menu Directory display.

PF4 ATTRIBUTES
Set the editor to attribute mode. This will display a different function key area at the bottom of the screen. For more information, please refer to CREATING COLOR AND HIGHLIGHTING under THE MENU TEXT EDITOR, earlier in this section.

PF5 DEFINE Display the Message Definition screen.

PF6 SELECT This is used in the Menu Editor to display a popup window with the menu definition selection information about the row that the cursor was in when the SELECT key was pressed. For more information, see *THE SELECT WINDOW*, earlier in this section.

PF7 BACKWARD Display the previous page of text.

PF8 FORWARD Display the next page of text.

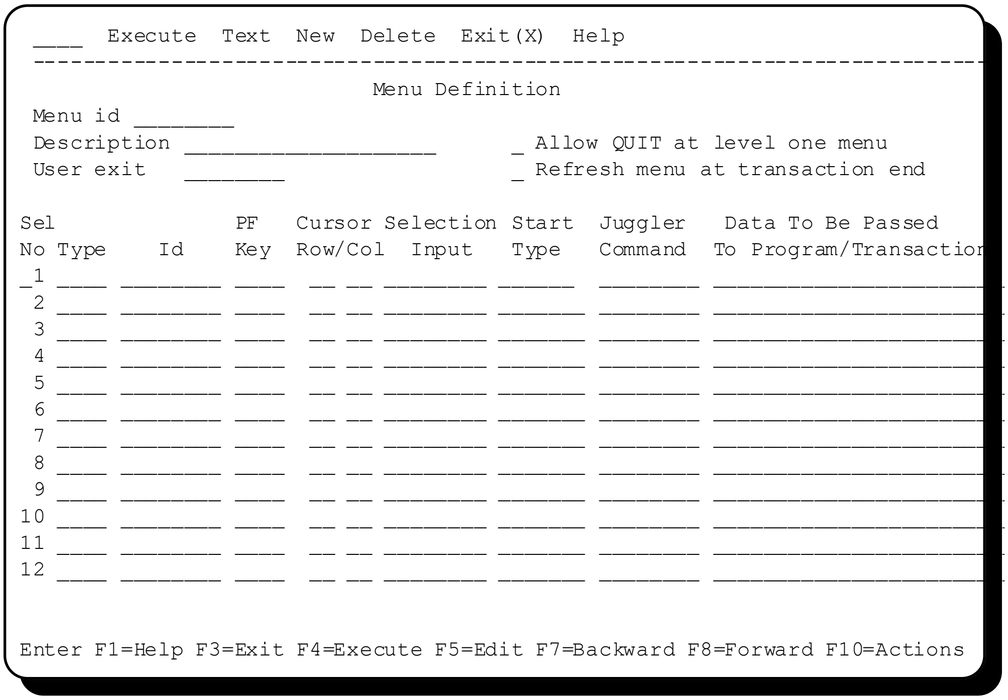
PF9 NULL ON/OFF
The editor can display trailing spaces as spaces or nulls. If spaces are used, you may type text in any column of the editor and upon pressing ENTER, the text will remain where it was typed, however, since all characters in a line are full, the INSERT key will not operate. If nulls are used, the INSERT key will

operate properly, but if keyed text is preceded with nulls, when ENTER is pressed the text will shift left one column for every preceding null. This key works as a toggle switch. Pressed once turns nulls on; pressed again turns nulls off.

PF11 SCALE ON/OFF
This is used to display a scale along the top of the editor. This works as a toggle switch to display or remove the scale.

THE MENU DEFINITION DISPLAY

This screen is used to define each item on the menu and specify what is to happen when that selection is made. This screen appears when defining or modifying parameters of a menu, (by pressing the DEFINE key from the Menu Editor display). Portions of this screen will appear when the SELECT key is pressed in the text editor. This screen appears as follows:



```
 ____  Execute  Text  New  Delete  Exit(X)  Help
 ------------------------------------------------------------------------------
                            Menu Definition
 Menu id ________
 Description ____________________      _ Allow QUIT at level one menu
 User exit   ________                  _ Refresh menu at transaction end

Sel              PF   Cursor Selection Start  Juggler   Data To Be Passed
No Type    Id    Key  Row/Col  Input   Type   Command  To Program/Transaction
_1 ____ ________ ____  __ __ ________ ______  ________ _______________________
 2 ____ ________ ____  __ __ ________ _______ ________ _______________________
 3 ____ ________ ____  __ __ ________ _______ ________ _______________________
 4 ____ ________ ____  __ __ ________ _______ ________ _______________________
 5 ____ ________ ____  __ __ ________ _______ ________ _______________________
 6 ____ ________ ____  __ __ ________ _______ ________ _______________________
 7 ____ ________ ____  __ __ ________ _______ ________ _______________________
 8 ____ ________ ____  __ __ ________ _______ ________ _______________________
 9 ____ ________ ____  __ __ ________ _______ ________ _______________________
10 ____ ________ ____  __ __ ________ _______ ________ _______________________
11 ____ ________ ____  __ __ ________ _______ ________ _______________________
12 ____ ________ ____  __ __ ________ _______ ________ _______________________


Enter F1=Help F3=Exit F4=Execute F5=Edit F7=Backward F8=Forward F10=Actions
```

FIELDS OF THE MENU DEFINITION DISPLAY

ALLOW QUIT AT LEVEL ONE MENU
If selected, an operator will be allowed to exit from this menu to CICS, if this menu was not entered from another menu.

CURSOR ROW/COL
These two fields are used to specify a position of the cursor to which the operator can tab and press ENTER, to invoke this function, if cursor selection of menu items is desired.

If desired, the column field can be coded as '00', to specify a generic column. In other words, to select this option, the cursor may appear anywhere within the specified row when ENTER is pressed. If column is coded, the cursor must be in exactly that position for selection to occur.

DATA TO BE PASSED TO PROGRAM/TRANSACTION

This field defines the data to be passed to the application program in the terminal I/O area or Interval Control Element. For ATTACH, the data defined is the data that would be entered immediately following the transaction code if this transaction were entered on a terminal screen. Thus, if only the trancode is required as input, this field should be left blank. For START, the data should be in the format expected by the target program.

Example To automatically start a transaction that would normally be entered on the screen in the form:

```
FSRV I  ACCTFIL  A*
```

the following coding would be required in the statement (in addition to the method(s) of selection of this option from the menu):

TYPE	TRAN
ID	FSRV
START TYPE	ATTACH
DATA	_I_ACCTFIL_A*

DESCRIPTION This is a short description of this menu definition. This field is not mandatory but is recommended for ease of recognition since it will appear on the Menu Directory Display.

IDThis is the actual ID of the function that is to be invoked. Valid entries are:

For TYPE=ESC
This field should be left blank.

For TYPE=MENU or SUBM
Code the Menu ID of the CICS-JUGGLER menu to be invoked.

For TYPE=PROG
Code the name of the program to be invoked, as it is known to CICS.

For TYPE=TRAN
Code the name of the transaction to be invoked, as it is known to CICS.

For TYPE=QUIT
This field should be left blank.

JUGGLER COMMAND
This field is used to specify any CICS-JUGGLER API commands to be processed upon selection of this menu. For more information on API commands, see *THE APPLICATION PROGRAM INTERFACE,* in section 14 - *USER EXITS AND PROGRAM INTERFACES.*

MENU ID This is the name of this menu. This name will appear in the Menu Directory display. It is a mandatory field.

PF KEY This field is used to specify a PF key that the operator can press to invoke this selection. This field is <u>not</u> mandatory. Entries are:

CLR This is the mnemonic for the CLEAR key.

ENTR This is the mnemonic for the ENTER key.

PFxx Any valid PF or PA key, coded as PFx, PFxx or PAx.

REFRESH MENU AT TRANSACTION END
If selected and the operator entered a transaction through this menu, display this menu when that transaction is terminated.

[Note]: Transaction end is recognized when there is no Return Transid and the task is not conversational. If selected, CICS-JUGGLER **must** be active on the terminal, for this feature to operate properly.

SEL NO This is the selection number. This is used only for displaying records beyond the ninth entry. The top most SEL NO field is unprotected. This enables you to change the starting selection number of the display. For instance, to display the next nine entries, key a "10" in this field and press ENTER.

SELECTION INPUT
This field is used to specify data that the operator may key (in an entry field) in order to invoke this menu option. (i.e. A, B, C... or 1, 2, 3...)

START TYPE This field is used to specify the technique for CICS-JUGGLER to use when starting a program or transaction. Valid entries are:

For TYPE=PROG

LINK This specifies that this program will be started and upon termination, control will return immediately to this menu.

XCTL This specifies that control will be transferred entirely to this program. Upon completion, CICS-JUGGLER will respond according to the coding of the REFRESH MENU AT TRANSACTION END field.

For TYPE=TRAN

ATTACH This means that the transaction will be started in the same way that CICS starts a transaction when the transaction code is entered from the terminal. That is, an I/O area is passed to the application containing the transaction code in the first position and any associated data following.

[Note]: ATTACH must be used if data is to be passed to the application in a terminal I/O area, or if the transaction being started runs in a remote region through MRO or ISC.

START START means that the transaction will be initiated using an EXEC CICS START. When using START, no terminal I/O area is passed to the application program. Data is passed in an interval control record, which the program can retrieve.

[Note]: There is no "preferred" method here. If a transaction does not need incoming data, either method will work. ATTACH will be slightly quicker since it avoids the CICS Automatic Transaction Initiation mechanism. If the program expects to receive data with an EXEC CICS RETRIEVE command, START must be used.

TYPE This field is used to specify the type of selection that this menu option will invoke. Valid entries are:

ESC When selected, this option will exit this menu to a clear screen no matter what level of menu nesting is in effect.

MENU When selected, this option will invoke another CICS-JUGGLER menu.

PROG When selected, this option will invoke a program.

SUBM When selected, this option will invoke a sub-menu. A sub-menu is a menu that appears within a window. Instead of displaying a completely different menu, the sub-menu window simply overlaps a portion of the screen.

TRAN When selected, this option will invoke a transaction.

QUIT When selected, this option will exit this menu to the next higher level (Either a clear screen or the menu that called this menu, depending on the method in which this menu was invoked.)

spaces This field may also be used to delete a menu selection line simply by erasing the TYPE field with the ERASE EOF key or space bar.

USER EXIT This field is used to specify a user exit program that will be called anytime this menu is displayed. The exit program may temporarily alter the way the menu displays and operates. For instance, if an operator is not authorized to used a certain transaction, the exit program can temporarily delete the option from the menu, so that the operator never even sees the option. For more information on this user exit, see *THE MENU GENERATION FEATURE USER EXIT*, in section 14 - *USER EXITS AND PROGRAM INTERFACES.*

ENTERING COMMANDS AT THE MENU DEFINITION DISPLAY

At the bottom of the screen is a list of available functions that may be invoked. You may activate the desired option by pressing the associated PF key or by tabbing to the option in the function key area and pressing ENTER.

ENTER Apply any changes or additions made to this screen; or effect the function in the function key area specified by the option to which the cursor pointed when ENTER was pressed.

PF1 HELP Display a Help screen for information pertaining to this screen or to the field that the cursor pointed when the HELP key was pressed.

PF3 EXIT Exit to the Menu Directory display.

PF4 EXECUTE Execute this menu. This will exercise the menu as the operator will see it.

PF5 EDIT Display the menu editor to view or change the actual menu.

PF7 BACKWARD Browse backward to the previous menu definition.

PF8 FORWARD Browse forward to the next menu definition.

In addition to the PF key functions, the action bar may be used to create a new menu, copy this menu, or delete this menu.

INITIATION AND OPERATION OF MENUS

Two methods for initiating a menu are available. One method is to press the EXECUTE key (shown in the PF key prompt area) from the Menu Directory or Menu Definition displays. The second method is to ENTER the JMNU,xxxxxxxx command from a clear screen (where xxxxxxxx is the Menu ID of the menu to be executed).

Once a menu is operating, the method(s) of selecting menu options is controlled entirely by the menu definition. Any of the following methods may be available:

- Tab to the option and press ENTER.
- Enter a selection in an entry field.
- Press a PF or PA key.

- Press the CLEAR key.

AUTOMATIC INITIATION OF MENUS

Menus can be defined to automatically initiate when a session is first entered or by toggling into it.

To do this, use the Application Startup Table of the CICS-JUGGLER customization. Code JMNU as the Transaction ID. ATTACH as the Start Type, and a comma followed by the menu ID in the Data field. You may code the same menu to be started in all sessions, or have different menus, as desired.

See *THE APPLICATION STARTUP TABLE*, in section 10 - *CUSTOMIZATION*, for more information.

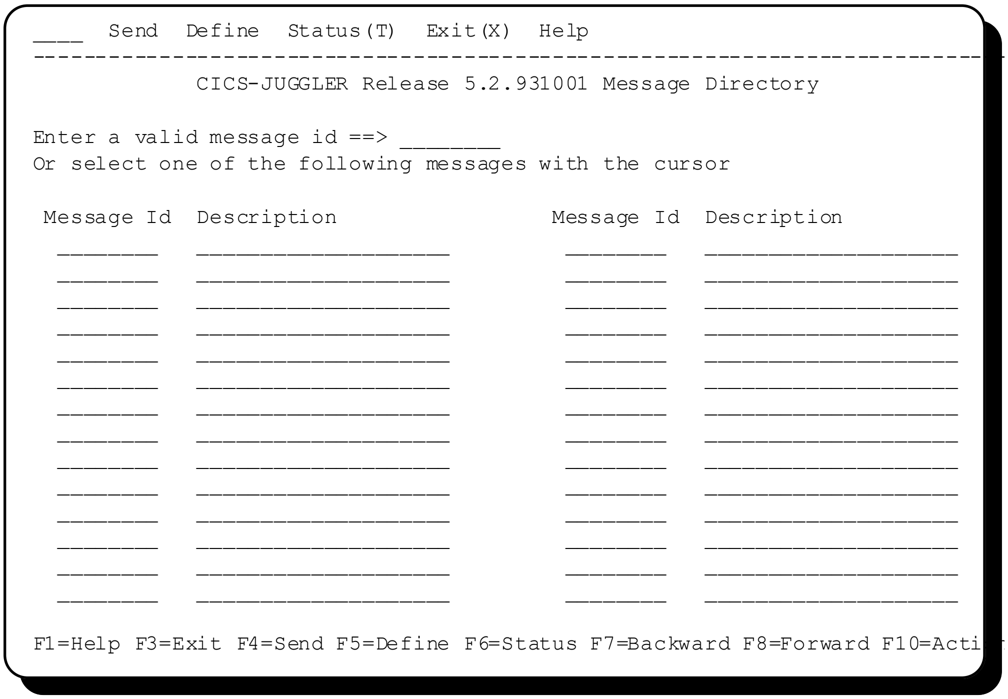
THE MESSAGE BROADCASTING FEATURE

An optionally licensed feature of CICS-JUGGLER is the flexible message broadcasting facility that may be used to transmit messages, news, inquiries etc. The message may be transmitted as a full-screen image or a pop-up window. In addition, the sender can specify one of four ways that the message is to be retrieved by the target terminal: immediately, upon response to a prompt, upon pressing the toggle key or upon pressing the CLEAR key. Also, the message may be broadcasted to a specific terminal, a specific user, a list of terminals and/or users, or to all terminals in this CICS.

The message broadcasting facility is accessed through the transaction code - **JMSG**, which is explained in detail in the following discussion.

THE MESSAGE DIRECTORY DISPLAY

To invoke the message broadcasting transaction, enter the transaction code **JMSG**. Upon pressing ENTER, the Message Directory display will appear generally as follows:



```
____  Send  Define  Status(T)  Exit(X)  Help
------------------------------------------------------------------------------
             CICS-JUGGLER Release 5.2.931001 Message Directory

Enter a valid message id ==> ________
Or select one of the following messages with the cursor

 Message Id  Description                   Message Id  Description
  ________   ____________________           ________   ____________________
  ________   ____________________           ________   ____________________
  ________   ____________________           ________   ____________________
  ________   ____________________           ________   ____________________
  ________   ____________________           ________   ____________________
  ________   ____________________           ________   ____________________
  ________   ____________________           ________   ____________________
  ________   ____________________           ________   ____________________
  ________   ____________________           ________   ____________________
  ________   ____________________           ________   ____________________
  ________   ____________________           ________   ____________________
  ________   ____________________           ________   ____________________
  ________   ____________________           ________   ____________________
  ________   ____________________           ________   ____________________

F1=Help F3=Exit F4=Send F5=Define F6=Status F7=Backward F8=Forward F10=Acti
```

FIELDS OF THE MESSAGE DIRECTORY DISPLAY

==> This is the message ID field and is used to access a message that may or may not be displayed on this screen. This field is used by keying the message ID, then pressing the desired PF key; or tabbing the cursor to the function in the function key area, and pressing ENTER.

The columns of fields below this point make up the directory of available messages. All messages on file are displayed in alphabetic order. Each column entry contains two fields:

DESCRIPTION This is a brief description of the message.

MESSAGE ID This is the name of the message. Since CICS-JUGGLER can retain messages along with all associated parameters, names are assigned to each message definition for ease of identification. These fields are unprotected so that you may tab to the desired message then press the PF key of the function that you wish to invoke. You can not change a message name with this field.

ENTERING COMMANDS AT THE MESSAGE DIRECTORY DISPLAY

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

PF1 HELP Display a Help screen for information pertaining to the directory.

PF3 EXIT Exit the Message Broadcasting facility.

PF4 SEND This is used to broadcast the message specified by the field that the cursor is in at the time PF4 is pressed or by the name keyed in the ==> field.

PF5 DEFINE This is used to view and optionally modify the message or the parameters of the message specified by the field that the cursor is in at the time PF5 is pressed or by the name keyed in the ==> field.

PF6 STATUS This is used to display the Active Message Status display for the message specified by the field that the cursor is in at the time PF6 is pressed or by the name keyed in the ==> field.

PF7 BACKWARD Browse backward to the previous entry of the message directory.

PF8 FORWARD Browse forward to the next entry of the message directory.

PF10 ACTIONS Remember the cursor position, and place the cursor in the action bar. This would be used for action bar function that are cursor position dependant.

DEFINING AND MAINTAINING MESSAGES

ADDING A NEW MESSAGE

To add a new message, from the Message Definition screen, activate the NEW pull-down menu, then select either ADD or COPY. ADD will create a new message with all fields blank. COPY will create a new message with the same specifications and message text as the currently displayed message .

Once ADD or COPY is selected, the screen will display the new message definition, requesting the message ID that is to be assigned to the new definition. Key a one to eight digit name and press ENTER to create the new message definition.

Once created, you may code the fields as described in *THE MESSAGE DEFINITION DISPLAY*, later in this section.

CHANGING AN EXISTING MESSAGE

To change an already existing message definition, from the Message Directory display, select the message definition that you wish to change, and press the DEFINE key.

The message definition will then be displayed. To alter the definition, key the changes and press ENTER.

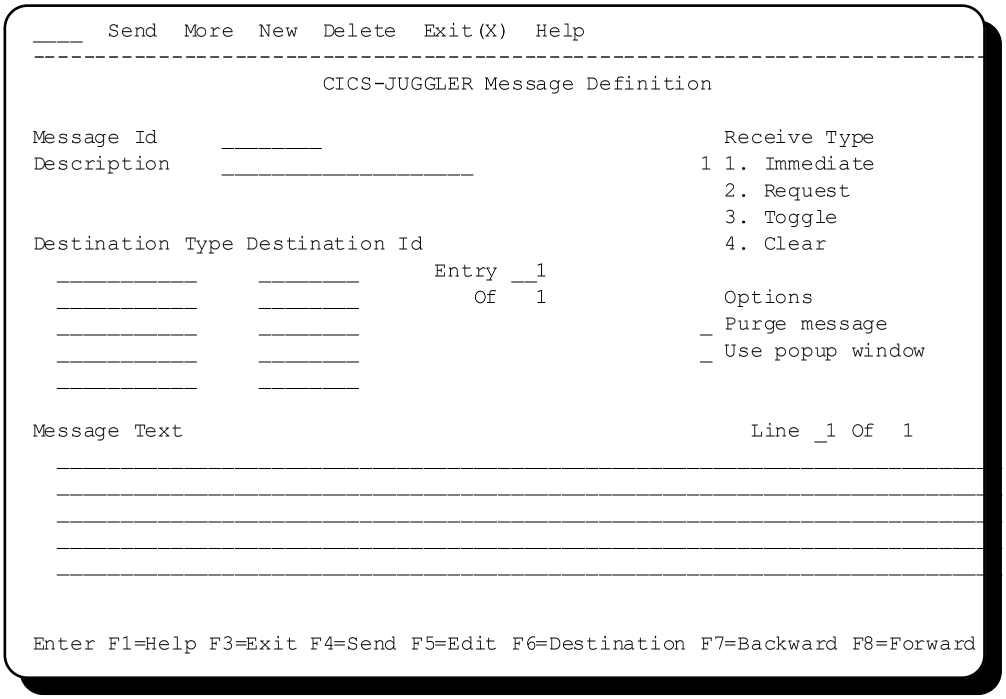
DELETING A MESSAGE

To delete a message definition, from the Message Directory display, select the message definition that you wish to delete, and press the DEFINE key.

The message definition will then be displayed. To delete the definition, activate the DELETE pull-down menu, and select DELETE.

THE MESSAGE DEFINITION DISPLAY

This screen is used to specify the broadcast parameters of a message. This screen is accessed by pressing the DEFINE key from the Message Directory display and appears generally as follows:



```
____  Send  More  New  Delete  Exit(X)  Help
----------------------------------------------------------------------------
                       CICS-JUGGLER Message Definition

Message Id     ________                                 Receive Type
Description    ____________________                   1 1. Immediate
                                                        2. Request
                                                        3. Toggle
Destination Type Destination Id                         4. Clear
  ___________     ________      Entry __1
  ___________     ________         Of   1               Options
  ___________     ________                            _ Purge message
  ___________     ________                            _ Use popup window
  ___________     ________

Message Text                                              Line _1 Of  1
  ___________________________________________________________________________
  ___________________________________________________________________________
  ___________________________________________________________________________
  ___________________________________________________________________________
  ___________________________________________________________________________


Enter F1=Help F3=Exit F4=Send F5=Edit F6=Destination F7=Backward F8=Forward
```

FIELDS OF THE MESSAGE DEFINITION DISPLAY

DESCRIPTION This is a short description of this message definition. This field is not mandatory but is recommended for ease of recognition.

DESTINATION ID This specifies the terminal or operator ID of the target.

For DESTINATION TYPE=TERMINAL

Code the terminal ID, as it is known to CICS. This may be coded in any of the following ways:

1) Code an individual statement for each terminal to be defined, coding this field as the four-character terminal ID of the terminal to be configured.

2) Define a generic definition using "wild-card" characters. Code this operand with question marks (?) in one or more positions of the terminal ID. With this method, the message is broadcast to any terminal that all positions in the terminal ID match the corresponding positions in this field where question marks are not coded.

3) Define a single statement which applies to all terminals. Code this field as ????. With this method, this statement applies to every terminal in this CICS.

For DESTINATION TYPE=OPERATOR ID

Code the operator name, as it appears in the TCT. This is the three-character CICS operator ID of the operator. This may be coded in any of the following ways:

1) Code an individual statement for each operator, coding the three-character operator ID of the user to receive the message.

2) Define a generic definition using "wild-card" characters. Code question marks (?) in one or more positions of the operator ID. With this method, the message is broadcast to any terminal that all positions in the operator ID match the corresponding positions where question marks are <u>not</u> coded.

3) Define a single statement which applies to all users. Code the field as ????. With this method, the message is broadcast to every operator in this CICS.

For DESTINATION TYPE=USER ID

Code the user ID, as it appears in the SNT. This is the eight-character user ID of the operator. This may be coded in any of the following ways:

1) Code an individual statement for each operator, coding the eight-character user ID of the user to receive the message.

2) Define a generic definition using "wild-card" characters. Code question marks (?) in one or more positions of the user ID. With this method, the message is broadcast to any terminal that all positions in the user ID match the corresponding positions where question marks are <u>not</u> coded.

3) Define a single statement that applies to all users. Code the field as ????????. With this method, the message is broadcast to every operator in this CICS.

DESTINATION TYPE
This specifies the targeting method to be used. Valid entries are:

TERMINAL Code 'TERMINAL' or 'T' if this entry is to specify a terminal as a target.

OPERATOR ID
Code 'OPERATOR ID' or 'O' if this entry is to specify an operator as a target.

USER ID Code 'USER ID' or 'U' if this entry is to specify a user as a target. Note that this is not valid for CICS releases prior to 1.7.

[Note]: If multiple entries point to the same physical terminal, all but the first entry will be ignored. (i.e. If a TERMINAL and OPERATOR ID entry is coded, and both entries point to the same terminal, CICS-JUGGLER will only honor the first entry.)

ENTRY xxx OF yyy
If the all message destination entries will not display on five lines, the xxx in this field may be altered to reposition the text to display starting with the specified line.

Note that you may also press the DESTINATION key (shown in the PF key prompt area). This will cause the display to change so that the entire screen is used to display the destination entries.

LINE xx OF yy If the entire message does not display on five lines, the xx in this field may be altered to reposition the text to display starting with the specified line.

MESSAGE ID This is the name of this message. This name will appear in the Message Directory display and is mandatory.

MESSAGE TEXT This is the actual text of the message. If desired, you may use these lines for creating or altering the message (instead of the message editor).

OPTIONS These are special options that apply to all message and receive types. Valid entries are:

PURGE MESSAGE
If selected, this specifies that this message definition is to be deleted, once all targeted terminals have viewed the message. If this option is not selected, this message definition will not be deleted, and will be available for future use.

USE POPUP WINDOW
If selected, a pop-up window will appear on the screen of the target terminal(s) to display the message, instead of using the full screen to display the message.

RECEIVE TYPE This is the method of message retrieval for the target terminal. Valid selections are:

IMMEDIATE The current display on the terminal is saved and the message is displayed. When the operator presses CLEAR, the original display is restored.

REQUEST The operator must enter the JMSG transaction to retrieve any messages. This makes the message broadcaster function similar to an electronic mail package. The message may also be requested from the Control Window.

TOGGLE The message will be displayed the next time the operator presses any toggle key or performs any function that causes a session switch. When CLEAR is subsequently pressed, the session switch will be completed.

CLEAR When the operator of the target terminal presses the CLEAR key, the message will display. When CLEAR is pressed again, CICS-JUGGLER will honor the clear, and act accordingly.

ENTERING COMMANDS AT THE MESSAGE DEFINITION DISPLAY

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

ENTER Apply any changes or additions made to this screen, pull down the specified menu from the the Action Bar, or effect the specified function in the PF key prompt line. Note that when pulling-down a menu or effecting a function, selection is specified by the position of the cursor when ENTER is pressed.

PF1 HELP Display a Help screen for information pertaining to this screen.

PF3 EXIT Exit to the Message Directory display.

PF4 SEND This is used to broadcast the currently displayed message.

PF5 EDIT This is used to view and optionally modify the message text of this message definition.

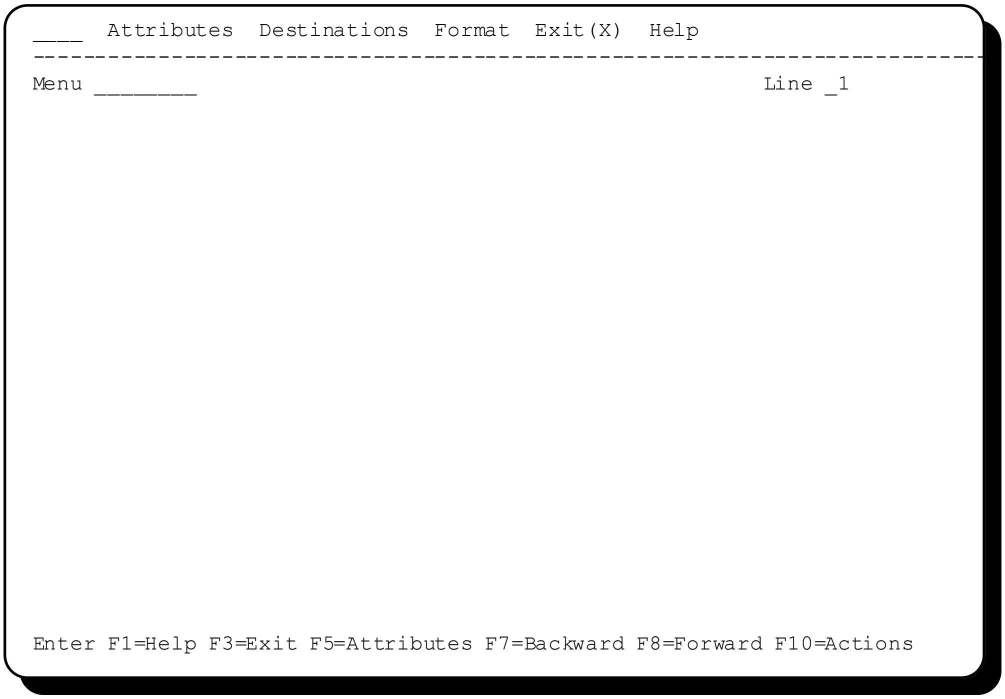
PF6 DESTINATION This displays a screen to edit as many as 56 destination specifications.

PF7 BACKWARD Browse backward to the previous message definition.

PF8 FORWARD Browse forward to the next message definition.

THE MESSAGE TEXT EDITOR

This is used to 'paint the screen' the way you want the message to be displayed. This screen appears when attempting to define or modify a message by pressing the EDIT key from the Message Directory or Message Definition displays. This screen appears as follows:



```
____  Attributes  Destinations  Format  Exit(X)  Help
----------------------------------------------------------------------------
Menu ________                                           Line _1












Enter F1=Help F3=Exit F5=Attributes F7=Backward F8=Forward F10=Actions
```

USING THE MESSAGE TEXT EDITOR

To enter text, simply key lines of data, formatting it as desired so far as line spacing, indentation, etc. are concerned. If changes are needed after keying some text, you can simply move the cursor to the area to be changed and type over any existing data. You can use the DEL key to delete characters in a line. To insert characters on a line, position the cursor where the insertion is to be made, press the INS key and type the desired characters (note that you may need to turn NULLS on, which is explained later in this topic). The ERASE EOF key may be used to erase all or part of a line at any time.

When all text for this page has been keyed, press ENTER. This will write the updated text record to file. If the text being entered cannot fit on one screen, you may press the FORWARD key (or change the line number at the top of the display).

LINE DELETION

To delete an entire line, TAB to the COMMAND FIELD at the end of the line (where the asterisks are) and enter a 'D', then press ENTER. The entire line will be deleted and all following lines will shift up one line.

To delete more than one line at a time, enter a 'D' in more than one COMMAND FIELD before pressing ENTER or enter 'Dnn' on one or more COMMAND FIELDS, where nn is the number of consecutive lines (including the current line) to be deleted.

You may also perform block deletion by placing 'DD' on the first line to be deleted and another 'DD' on the last line to be deleted.

LINE INSERTION

To insert a line between two text lines, tab to the COMMAND FIELD of the preceding line and enter an 'I' over any one of the asterisks, then press ENTER. This will cause a blank line to be inserted following that line, and all subsequent lines will shift down one line. You may then key any desired text on that line.

To insert multiple lines, enter 'Inn' in the preceding COMMAND FIELD, where nn is the number of lines to insert, then press ENTER. This will cause nn blank lines to be inserted after the current line, and all subsequent lines will shift down nn lines.

MOVING AND COPYING TEXT LINES

One or more consecutive text lines can be moved or copied to another location. To move text lines, enter 'M' or 'Mnn' (nn = number of lines to move) in a COMMAND FIELD and press ENTER. This causes the specified line(s) to be deleted from that location. Now place an 'A' (insert-after) in the COMMAND FIELD on the line preceding the new location, or a 'B' (insert-before) on the line following the new location and press ENTER. The moved lines will be inserted following that line if the 'A' command is used, or before that line for the 'B' command.

To copy text lines, enter 'C' or 'Cnn' (nn = number of lines to copy) in a COMMAND FIELD and press ENTER. Now place an 'A' (insert-after) in the COMMAND FIELD on the line preceding the new location, or a 'B' (insert-before) on the line following the new location and press ENTER.

You may also perform block moves and copies by placing 'MM' or 'CC' on the first line to be moved or copied and another 'MM' or 'CC' on the last line to be moved/copied.

You may enter more than one 'M' or 'C' command on one edit screen, if desired. The result is that all selected lines are grouped together consecutively and inserted as one set wherever the 'A' or 'B' command is placed. For instance, if 'C2' is entered in the command area at line 3, then 'C4' is entered in the command area at line 16, all six text lines will be grouped together and inserted as one set following the line where the 'A' command is placed.

Note that you may use the 'A' or 'B' sub-commands as many times as desired, without performing another 'M' or 'C' sub-command. Each time it is used, the same data that was moved or copied last will be inserted at the new location.

CREATING COLOR AND HIGHLIGHTING

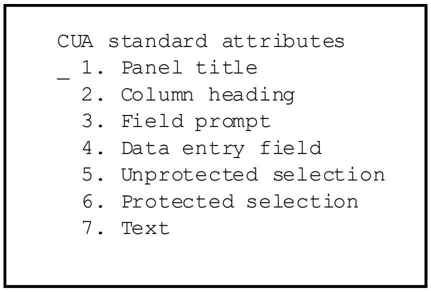
Unless otherwise specified, all words in the message will be displayed in low intensity protected. In order to emphasize certain words or lines, you can use the ATTRIBUTES function.

To invoke the Attribute function, press the ATTRIBUTES key (shown in the PF key prompt area). This places the terminal in SET COLOR mode. Note that upon invoking the attribute function, the function key area changes to include additional functions. These additional functions are as follows:

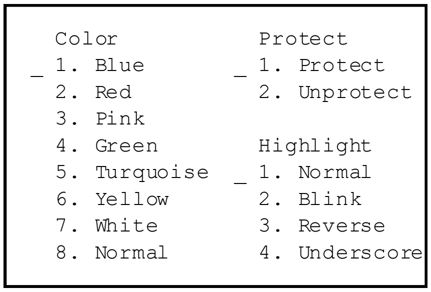
ENTER	If the cursor is inside the attribute window, pressing ENTER will move the cursor to the current attribute pointer position.

F2=SHOW Show all attributes on the screen as an at-sign (@).
F3=EXIT Exit Set Attribute mode to the editor.
F4=SET Set an attribute (as specified in the attribute window) at the current cursor position.
F5=SELECT Select the attribute at the position of the cursor and display the type of attribute in the attribute window. (If the cursor is inside the attribute window, this will move the window out of the way.)
F6=REMOVE Remove (delete) the attribute at the position of the cursor.
F7=BWD Move the current attribute pointer to the previous attribute.
F8=FWD Move the current attribute pointer to the next attribute.
F9=EXTEND Toggle the attribute window between CUA and extended attributes.

When the attributes function is invoked, a pop-up window appears. The attributes window can appear in two forms:



```
CUA standard attributes
_ 1. Panel title
  2. Column heading
  3. Field prompt
  4. Data entry field
  5. Unprotected selection
  6. Protected selection
  7. Text
```



```
  Color           Protect
_ 1. Blue       _ 1. Protect
  2. Red          2. Unprotect
  3. Pink
  4. Green        Highlight
  5. Turquoise  _ 1. Normal
  6. Yellow       2. Blink
  7. White        3. Reverse
  8. Normal       4. Underscore
```

- A CUA standards attribute window. This window contains the CUA attributes for creating a display. The following chart portrays the display attributes for the various selections:

Selection	Extended color	Intensity	Protect/unprotect
Panel title	Yellow	High	Protected
Column Heading	Turquoise	High	Protected
Field Prompt	Turquoise	Normal	Protected
Data Entry	Green	Normal	Unprotected
Unprotected Selection	White	Normal	Unprotected
Protected Selection	White	Normal	Protected
Text	Blue	Normal	Protected

- An extended attribute window. This window offers all attribute combinations.

If the CUA standard window is displayed and you wish to use the extended attribute window, press the EXTEND key (shown in the PF key area). This key works as a toggle between the two types of attribute windows.

Upon pressing the ATTRIBUTES key, either a field mark or an asterisk with an overscore will display somewhere on the screen. This is the Current Attribute Pointer. If an attribute is present at that position of the screen, it will appear as a field mark, else it will appear as the overscored asterisk. You may press the FORWARD or BACKWARD keys to position the pointer to the next or previous attributes on the screen, or you may position the cursor to a suspected location of an attribute and press the SELECT key to move the attribute pointer directly to that position.

To set an attribute, simply select the choices within the attribute window by entering the number of the choice. Then position the cursor to a space before the word (or line) that is to contain the attribute features. Then press the SET key (shown in the function key area). The attribute will remain in effect until another attribute or the end of the line is encountered.

To remove an attribute, position the cursor to the attribute to be removed and press the REMOVE key (shown in the PF key area).

[Note]: Setting color with either attributes windows may cause extended attributes to be generated in the screen display when viewed. You must be using a terminal that supports extended attributes and the COLOR and/or HILIGHT feature(s) must be set in the CICS terminal control table (TCT) entry (or EXTENDEDDS with RDO) in order to view them. If you view a display on a terminal that does not support extended datastreams, the extended color and highlighting attributes will be removed (in favor of regular field attributes) when the screen is displayed.

ENTERING COMMANDS AT THE MESSAGE TEXT EDITOR

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key.

ENTER Update the message or message and/or apply any commands in the command field (to the right of the screen).

PF1 HELP Display a Help screen for information pertaining to the text editor.

PF3 EXIT Exit to the Message Directory display.

PF5 ATTRIBUTES
Set the editor to attribute mode. This will display a different function key area at the bottom of the screen. For more information, please refer to *CREATING COLOR AND HIGHLIGHTING* under *USING THE MESSAGE AND MESSAGE TEXT EDITOR*, earlier in this section.

PF7 BACKWARD Display the previous page of text.

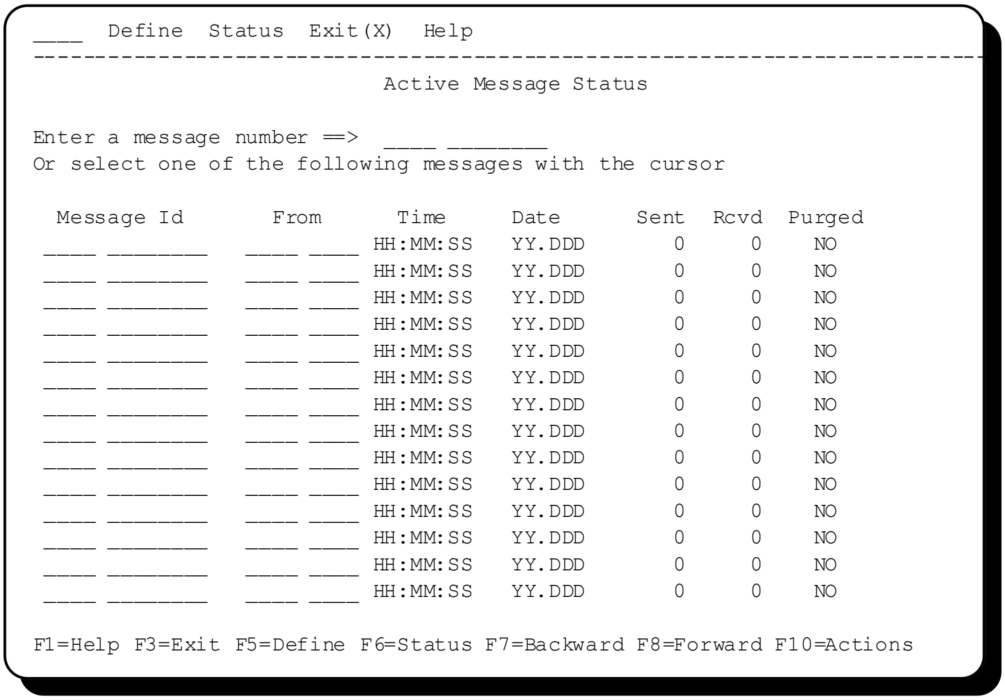
PF8 FORWARD Display the next page of text.

PF9 NULL ON/OFF
The editor can display trailing spaces as spaces or nulls. If spaces are used, you may type text in any column of the editor and upon pressing ENTER, the text will remain where it was typed, however, since all characters in a line are full, the INSERT key will not operate. If nulls are used, the INSERT key will operate properly, but if keyed text is preceded with nulls, when ENTER is pressed the text will shift left one column for every preceding null. This key works as a toggle switch. Pressed once turns nulls on; pressed again turns nulls off.

PF11 SCALE ON/OFF
This is used to display a scale along the top of the editor. This works as a toggle switch to display or remove the scale.

THE ACTIVE MESSAGE STATUS DISPLAY

The Active Message Status screen displays a summary of the active messages (messages that have been sent during the period that CICS has been active). The Active Message Status screen is displayed by pressing the STATUS key from the Message Directory display. It appears generally as follows:



```
____  Define  Status  Exit(X)  Help
---------------------------------------------------------------------------
                            Active Message Status

Enter a message number ==>  ____ ________
Or select one of the following messages with the cursor

  Message Id       From      Time      Date      Sent  Rcvd  Purged
 ____ ________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ ________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ ________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ ________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ ________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ ________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ ________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ ________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ ________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ ________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ ________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ ________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ ________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ ________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO

F1=Help F3=Exit F5=Define F6=Status F7=Backward F8=Forward F10=Actions
```

FIELDS OF THE ACTIVE MESSAGE STATUS DISPLAY

DATE This is the date that this message was sent.

FROM This is the terminal ID and operator ID of the terminal from which this message was sent.

MESSAGE ID This is the name of the message.

PURGED This indicates whether this message definition has been deleted.

RCVD This is the number of targeted terminals that have received and responded to this message.

SENT This is the number of terminals that have been targeted to receive this message.

TIME This is the time that the message was sent.

ENTERING COMMANDS AT THE ACTIVE MESSAGE STATUS DISPLAY

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

PF1 HELP Display a Help screen for information pertaining to this screen.

PF3 EXIT Exit to the Message Directory display.

PF5 DEFINE This is used to view and optionally modify the message or the parameters of the message to which the cursor pointed at the time PF5 is pressed or by the name keyed in the ==> field.

PF6 STATUS Display the Terminal Status screen for the message to which the cursor pointed at the time PF6 is pressed or by the ==> field.

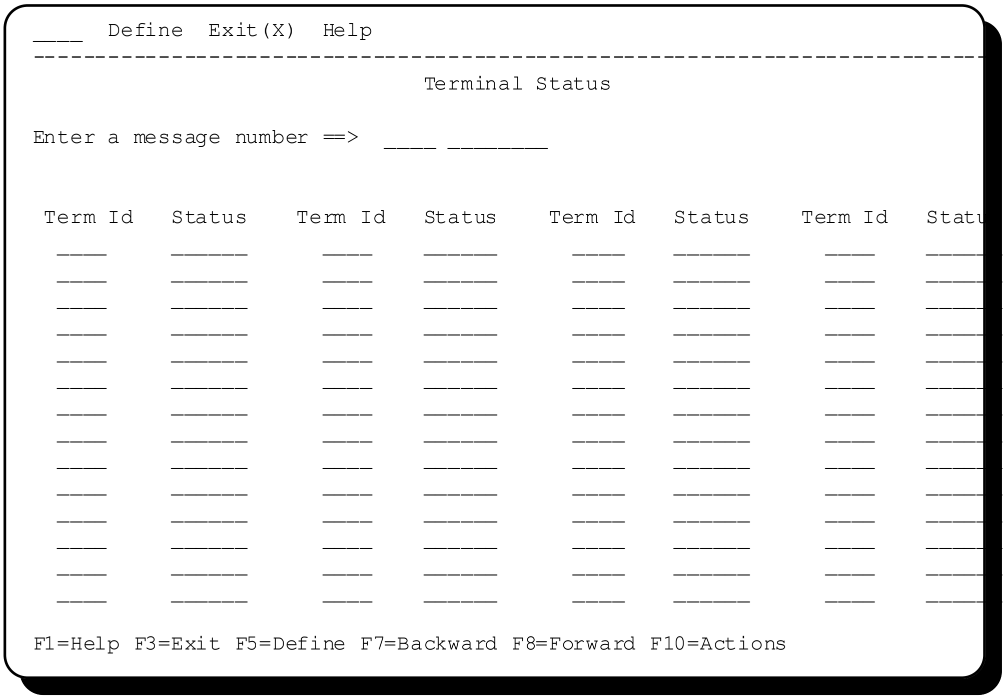
PF7 BACKWARD Browse backward to the previous page.

PF8 FORWARD Browse forward to the next page.

PF10 ACTION Remember the cursor position, and place the cursor in the action bar. This would be used for action bar function that are cursor position dependant.

THE TERMINAL STATUS DISPLAY

The Terminal Status screen displays a summary of the destinations that are specified by the message shown in the ==> field. The Terminal Status screen is displayed by pressing PF6 from the Active Message Status display. It appears as follows:



```
____  Define  Exit(X)  Help
--------------------------------------------------------------------------
                                Terminal Status

Enter a message number ==>  ____ ________


 Term Id   Status    Term Id  Status    Term Id  Status    Term Id  Statu
  ____     ______     ____    ______     ____    ______     ____    _____
  ____     ______     ____    ______     ____    ______     ____    _____
  ____     ______     ____    ______     ____    ______     ____    _____
  ____     ______     ____    ______     ____    ______     ____    _____
  ____     ______     ____    ______     ____    ______     ____    _____
  ____     ______     ____    ______     ____    ______     ____    _____
  ____     ______     ____    ______     ____    ______     ____    _____
  ____     ______     ____    ______     ____    ______     ____    _____
  ____     ______     ____    ______     ____    ______     ____    _____
  ____     ______     ____    ______     ____    ______     ____    _____
  ____     ______     ____    ______     ____    ______     ____    _____
  ____     ______     ____    ______     ____    ______     ____    _____
  ____     ______     ____    ______     ____    ______     ____    _____
  ____     ______     ____    ______     ____    ______     ____    _____

F1=Help F3=Exit F5=Define F7=Backward F8=Forward F10=Actions
```

FIELDS OF THE TERMINAL STATUS DISPLAY

STATUS This is the status of the terminal. Possible values to be displayed in this field are:

WAITING The terminal has not yet received the message.

IN PROGRESS The terminal is currently displaying the message.

TERM ID This is the ID of the terminal that has been targeted to receive the message. Once a terminal has received the message, the terminal will no longer appear on this list.

ENTERING COMMANDS AT THE TERMINAL STATUS DISPLAY

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

PF3 EXIT Exit to the Active Message Status display.

PF5 DEFINE This is used to view and optionally modify the message or the parameters of the message to which the cursor pointed at the time PF5 is pressed or by the name keyed in the ==> field.

PF7 BACKWARD Browse backward to the previous page of terminal entries.

PF8 FORWARD Browse forward to the next page of terminal entries.

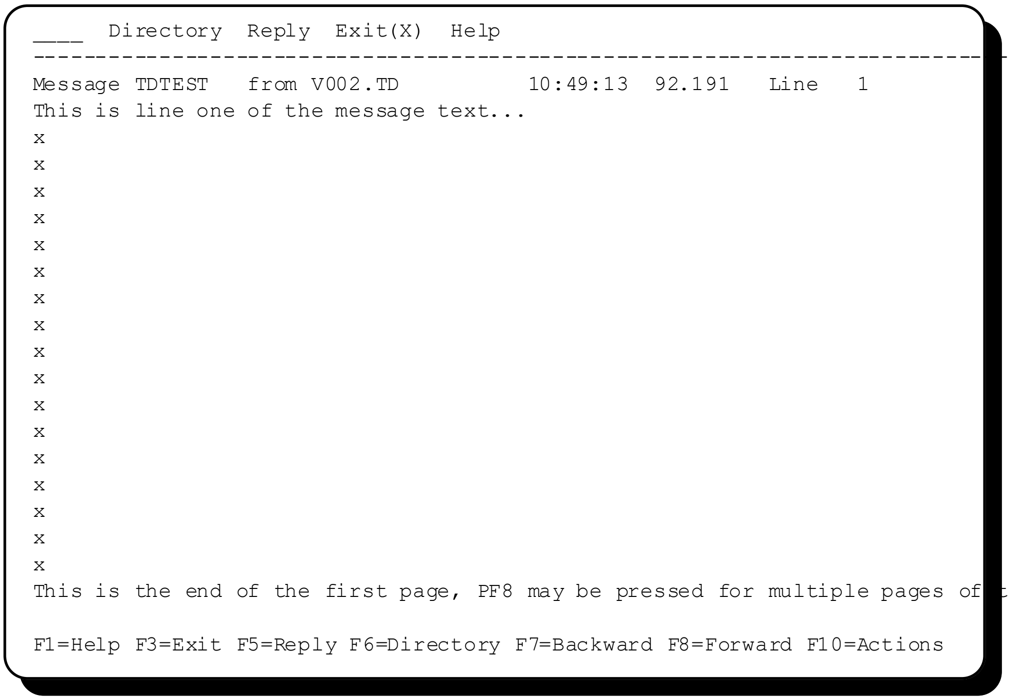
PF10 ACTIONS Remember the cursor position, and place the cursor in the action bar. This would be used for action bar function that are cursor position dependant.

THE MESSAGE RECEPTION DISPLAY

When a message has been transmitted, either the message will be immediately received, or it will be queued and will wait until the operator requests the message to be displayed, depending on the coding of the Receive Type options of the message definition.

Once the message is displayed by either receive type, the message will be displayed in the phantom session. The Phantom Session is a temporary session for CICS-JUGGLER internal use only. This session is in addition to the number of sessions that are present on this terminal. It is made available when needed by CICS-JUGGLER and disappears once the need for this session is resolved.

The message is displayed on the Message Reception Display, which appears generally as follows:



```
____  Directory  Reply  Exit(X)  Help
-------------------------------------------------------------------------------
Message TDTEST   from V002.TD          10:49:13  92.191   Line   1
This is line one of the message text...
x
x
x
x
x
x
x
x
x
x
x
x
x
x
x
x
x
This is the end of the first page, PF8 may be pressed for multiple pages of

F1=Help F3=Exit F5=Reply F6=Directory F7=Backward F8=Forward F10=Actions
```

Along with the message text, the Message Reception screen displays the message ID, the terminal and operator IDs of the terminal that sent the message and the time and date the message was sent.

ENTERING COMMANDS AT THE MESSAGE RECEPTION DISPLAY

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

PF1 HELP Display a Help screen for information pertaining to this screen.

PF3 EXIT This will exit the display and return to the transaction that was executing prior to receiving the message.

PF5 REPLY This will display the message editor, in order to allow a response to be sent to the terminal that transmitted the message. Fore more information about the editor, see *THE MESSAGE TEXT EDITOR*, earlier in this section.

PF6 DIRECTORY This will display the Message Directory Display, just as if the message broadcasting facility was entered through the JMSG transaction code. The only difference being, when PF3 is pressed from the Message Directory display (to exit the transaction), the current screen will be displayed instead of returning to CICS.

PF7 BACKWARD Browse backward to the previous page of message text.

PF8 FORWARD Browse forward to the next page of message text.

PF10 ACTION Remember the cursor position, and place the cursor in the action bar. This would be used for action bar function that are cursor position dependant.

INSTALLATION OF THE CICS-JUGGLER PRODUCT

This section describes the procedures for installing CICS-JUGGLER in your CICS environment. Procedures for both DOS/VSE and OS/MVS are described.

CICS REQUIREMENTS (*** MUST READ ***)

1). The parameter EXITS=YES must be specified in the System Initialization Table or as a SIT override.

2). The module DFHPCPxx must run in the same partition or region as CICS. That is, for MVS, these modules can not reside in the LPA and for DOS, they can not reside in the SVA. (xx is the CICS Module suffix).

CICS 3.3 users must either unprotect the extended read-only DSA (RENTPGM=NOPROTECT in DFHSIT) or remove the reentrant flag from the DFHPCP load module, which will cause it to load in storage protect key 8, rather than zero.

3). If your CICS contains a Nucleus Load Table (NLT) with PROTECT=YES, the DFHPCPxx must not be included in the NLT (xx is the CICS Module suffix).

4). For VTAM systems, if the READ BUFFER keyword of the CICS-JUGGLER customization table is coded YES, the maximum TIOAL parameter of the CICS Terminal Control Table may need to be increased. If you experience ATNI abends (particularly on remote terminals), or if your terminal hangs when the Toggle key is pressed, increasing this value will usually correct the problem.

5). CICS-JUGGLER will run on CICS versions 1.7 through 3.3, inclusive.

INSTALLATION STEPS

Installation consists of the following steps:

1). Link-edit the STSINST program from file 1 of the tape.
2). Run the STSINST program to print the full installation instructions and create the installation JCL.
3). Tailor the installation JCL if required.
4). Run the installation JCL to load all programs and files.
5). Define the required CICS table entries.
6). Define the DFHZNEP interface (if desired).
7). Run the installation verification transaction JIVP, which should find most common installation errors.

THE INSTALLATION TAPE

The installation tape consists of two files.

File 1. The first file is the object records for four program modules which are used both to finish the installation process and to provide the temporary expiration password. These three programs are:

1). STSINST - The installation program.
2). STS0100 - The password processor program.
3). STSPASS - The password table. This module is pre-loaded with a 30-day password.
4). STSCORE -A general purpose CICS memory display/alter utility.

File 2. The second file contains the records for seven of the products available from Unicom Systems. You must use the STSINST program to extract and deblock the product to be installed. The products present in file 2 are:

CICS-WINDOWS
CICS-JUGGLER
VTAM-EXPRESS
CICS-FILESERV
HELP-WINDOWS
VTAM-WINDOWS

Feel free to install any of the other products on the tape. Each product has a documentation member with the installation procedure and a brief overview of the product. A technical reference guide for that product can be obtained from your sales representative at Unicom Systems.

DOS/VSE INSTALLATION

Step 1. Load the installation programs and password table

The following JCL can be used to install the first file of the tape for DOS/VSE users:

```
// JOB LOAD STSINST
// ASSGN SYSIPT,xxx                                          [Note 1]
// MTC REW,SYSIPT
// LIBDEF PHASE,CATALOG=L.S                                  [Note 2]
// OPTION CATAL
   INCLUDE
// EXEC LNKEDT
// RESET SYSIPT
/*
```

[Notes]:

1. Assign SYSIPT to a tape drive that can read the BPI of your installation tape.
2. This LIBDEF is for DOS/VSE, where
   L.S = Library, Sub-library

   The library for the install program, password processor and password table must be a CICS load library.

Step 2. Print installation instructions and punch installation JCL

The following JCL executes the STSINST program to print the installation documentation and punch the install JCL:

```
// ASSGN SYS011,xxx                                    [Note 1]
// LIBDEF *,SEARCH=L.S                                 [Note 2]
// EXEC STSINST
PRODUCT=JUGGLER                                        [Note 3]
MODE=PRINT                                             [Note 3]
OPSYS=DOS                                              [Note 3]
CICS=x.x                                               [Note 3]
LINES=56                                               [Note 3]
JCL=INSTALL                                            [Note 3]
/*
/&
```

[Notes]:

1. Assign SYS011 to a tape drive that can read the BPI of your installation tape.

2. The LIBDEF must identify the library where STSINST was link-edited as a search library. L.S = Library, Sub-library.

3. See the discussion entitled *KEYWORDS OF THE UNICOM INSTALLATION PROGRAM*, later in this section, for the meaning of these keywords.

At this point, the full installation instructions will be printed. The installation JCL will be punched into the POWER punch queue. Retrieve or view the printed installation documentation and follow the instructions provided there to complete the installation of CICS-JUGGLER.

MVS INSTALLATION

For MVS, you must prepare initial JCL for two steps:

Step 1. Load the installation programs and password table

The following JCL can be used to install the first file of the tape for MVS users:

```
//LOADSTS     JOB    1,'ACCOUNT-ID',MSGCLASS=x
//STEP1       EXEC   PGM=IEWL,PARM='LIST,LET,XREF'
//SYSPRINT    DD     SYSOUT=*
//SYSLIB      DD     DSN=CICS.STEPLIB,                              [Note 1]
//                   DISP=SHR
//SYSLIN      DD     DSN=MASTER.TAPE,
//                   UNIT=TAPE,
//                   VOL=(,RETAIN,,,SER=INPUT),
//                   LABEL=(1,NL),
//                   DCB=(RECFM=FB,LRECL=80,
//                   BLKSIZE=80),
//                   DISP=(SHR,PASS)
//SYSUT1      DD     UNIT=SYSDA,
//                   SPACE=(1024,(20,20))
//SYSLMOD     DD     DSN=CICS.STEPLIB,                              [Note 1]
//                   DISP=SHR
```

[Notes]:

1. The receiving library must be a CICS load library.

Step 2. Print installation instructions and create installation JCL

The following JCL executes the STSINST program to print the installation documentation and create the install JCL:

```
//STEP2       EXEC   PGM=STSINST
//SYSPRINT    DD     SYSOUT=*
//STEPLIB     DD     DSN=CICS.STEPLIB,                              [Note 1]
//                   DISP=SHR
//MPRDIN      DD     DSN=MASTER.TAPE,UNIT=TAPE,
//                   LABEL=(2,NL),VOL=SER=INPUT,
//                   DCB=BLKSIZE=32000,
//                   DISP=(SHR,PASS)
//SYSPCH      DD     DSN=????.????(STSJCL),                         [Note 2]
//                   DCB=BLKSIZE=????,DISP=SHR
//SYSIN       DD     *
PRODUCT=JUGGLER                                                     [Note 3]
MODE=PRINT                                                          [Note 3]
OPSYS=MVS                                                           [Note 3]
CICS=xxx                                                            [Note 3]
LINES=56                                                            [Note 3]
JCL=INSTALL                                                         [Note 3]
CSDLIB=??????.??????                           (Optional)           [Note 4]
DFHCSD=??????.DFHCSD                           (Optional)           [Note 4]
INSTLIB=??????.??????                          (Optional)           [Note 4]
SYSLIB=??????.??????                           (Optional)           [Note 4]
SYSLMOD=??????.??????                          (Optional)           [Note 4]
SYSUT1=??????.??????                           (Optional)           [Note 4]
```

[Notes]:

1. The STEPLIB must identify the library where STSINST was link-edited in step 1.

2. This is any PDS where you want to load the installation JCL. You must specify a member name for your JCL and you must specify the blocksize of the PDS on the DCB parameter.

3. See the discussion entitled *KEYWORDS OF THE UNICOM INSTALLATION PROGRAM*, later in this section, for the meaning of these keywords.

4. These are optional parameters. If they are not present, the JCL punched from this job will contain question marks (?) at the applicable points, which will need to be tailored to your installation. If you specify these parms, the JCL will be punched with the designated parm rather than '????????'. These parameters are explained in more detail in *KEYWORDS OF THE UNICOM INSTALLATION PROGRAM*, later in this section.

At this point, the full installation instructions will be printed. The installation JCL will be loaded into the member name of the PDS specified by SYSPCH. Retrieve or view the printed installation documentation and follow the instructions provided there to complete the installation.

DEFINING THE DFHZNEP INTERFACE

This step does not have to be performed in order to operate CICS-JUGGLER; however, prior to using CICS-JUGGLER in a production environment, you should add some instructions to the node error program to purge CICS-JUGGLER from the terminal when an error or time-out occurs.

For details of the DFHZNEP interface, refer to section 14 - *DEACTIVATING CICS-JUGGLER WHEN A NODE ERROR OCCURS*.

ADDITIONAL INSTALLATION NOTES, MVS AND DOS/VSE

1). If you use RDO to define the programs, be sure to specify ASSEMBLER as the language.

2). If you wish to use a different transaction code than JUGL, you may make the TRANSID value any desired transaction code. If more than one transaction code is needed (for separate activity accounting or security reasons), you may provide as many PCT entries as desired, all pointing to program JUGGLER.

3). Slightly improved response time when in window mode may result if the JUGLMAIN program is made resident. It requires about 50K.

4). For MVS users only, if CICS-JUGGLER has been previously installed using the same source PDS, you must first delete the following members from the dataset since IEBUPDTE will not replace the existing members:

```
JUGLACF2      JUGLCSSF      JUGLCSSN
JUGLINT3      JUGLINT4      JUGLINT5
JUGLNEPC      JUGLNEPM      JUGLPURG
JUGLRSDM      JUGLRSDD      JUGLTBL
```

SUMMARY

At this point, the basic CICS-JUGGLER product is installed and ready to operate.

You may want to tailor the product to fit your environmental needs, as described in section 10 - *CUSTOMIZATION*, however the product is operational without doing so.

KEYWORDS OF THE UNICOM INSTALLATION PROGRAM

The installation program, STSINST, which is provided on the first file of the installation tape can be used to print the documentation or perform a complete install for any of the products on the master installation tape.

The individual installation instructions of each product provide the necessary JCL and keyword parameters that are specific to that product.

This section describes all of the available keywords of the STSINST program, plus the coding format used.

FORMAT OF THE STSINST KEYWORDS

All keywords and operands of the STSINST program are coded in the SYSIN (MVS) or SYSIPT (DOS) dataset. Each keyword and operand occupies one statement. You cannot concatenate multiple keywords on one line. Each keyword must begin in position one.

Following are all of the possible keywords and operands of the STSINST program, listed in alphabetical order. A detailed explanation of each keyword and operand follows.

Underlined values listed are the default values if the associated operand is omitted. Values separated by a vertical bar (|) indicate mutually exclusive values which may be coded.

KEYWORD DESCRIPTIONS

DIRECTORY The DIRECTORY keyword requires no operands. If present, it will print a directory of the tape contents.

CICS=160 | 170 | 210 | 211 | 212 | 311 | 321 | 330
This is the CICS release level. Valid entries are 160, 170 and 210 for DOS/VSE. For MVS: 170, 210, 211, 212, 311, 321, and 330. This keyword provides two functions. The primary purpose is to ensure that the correct version of CICS-JUGGLER is installed for your CICS release. The second function is to automatically adjust the JCL for the default dataset names. (i.e. for CICS=330, when referencing the install library, the punched JCL will reference CICS330.INSTLIB.)

CSDLIB=??????.??????
This is an optional keyword and is intended for MVS use only. This is not applicable to CICS releases prior to and including CICS 2.1.2. CSDLIB is used to specify the steplib for DFHCSDUP modules. If used, this keyword **must** be specified in the CREATE install step.

DFHCSD=??????.DFHCSD
This is an optional keyword and is intended for MVS use only. This is not applicable to CICS releases prior to and including CICS 2.1.2. DFHCSD is used to specify the DSN for the DFHCSD file. If used, this keyword **must** be specified in the CREATE install step.

FSRVFILE=xxxxxx
The FSRVFILE keyword identifies the name of the DLBL or DD statement in the job control statements for this execution of STSINST that names the VSAM file for CICS-FILESERV. The xxxxxx is the file ID of the DLBL statement (DOS) or DDNAME of the DD statement (MVS).

HELPSCRN=xxxxx

The HELPSCRN keyword identifies the name of the DLBL or DD statement in the job control statements for this execution of STSINST that names the VSAM screen file for HELP-WINDOWS. The xxxxxx is the file ID of the DLBL statement (DOS) or DDNAME of the DD statement (MVS).

HELPTEXT=xxxxxx
The HELPTEXT keyword identifies the name of the DLBL or DD statement in the job control statements for this execution of STSINST that names the VSAM text file for HELP-WINDOWS. The xxxxxx is the file ID of the DLBL statement (DOS) or DDNAME of the DD statement (MVS).

INSTLIB=??????.??????
This is an optional keyword and is intended for MVS use only. INSTLIB is used to specify the steplib for the STSINST module. If used, this keyword **must** be specified in the CREATE install step.

JCL=INSTALL|REINSTALL
The JCL keyword defines the action to be taken for the installation job control statements. Valid operands are:

INSTALL Create JCL which will delete and redefine the VSAM file(s).

REINSTALL Create JCL which will not delete nor redefine the VSAM file(s).

If the JCL keyword is omitted, the installation JCL will be printed only. If either INSTALL or REINSTALL is specified, the JCL will be punched to the POWER punch queue (DOS) or written to the PDS member identified by the SYSPCH DD statement (MVS).

JUGLFILE=xxxxxx
The JUGLFILE keyword identifies the name of the DLBL or DD statement in the job control statements for this execution of STSINST that names the VSAM file for CICS-JUGGLER. The xxxxxx is the 1-7 byte file ID of the DLBL statement (DOS) or 1-8 byte DDNAME of the DD statement (MVS).

LINES=56|nn The LINES keyword defines the number of lines to print on the system printer before skipping to a new page. If omitted, 56 is the default.

MODE=PRINT|CREATE|INSTALL|REINSTALL|DOCUMENT
The MODE keyword defines the action to be taken by STSINST. Valid mode operands are:

PRINT Print the installation instructions for the specified product.

CREATE Create an installation file on the work tape for the specified product.

INSTALL Load the VSAM files associated with the specified product. Coding MODE=INSTALL indicates that this is a new installation, loading to an empty VSAM file.

REINSTALL Load the VSAM files associated with the specified product. MODE=REINSTALL indicates that this is a subsequent installation, reloading the system records to an existing VSAM file that contains user data.

DOCUMENT Print the overview documentation for the specified product. The overview document is a short preview of the product, telling you how to get started using it. If you need the complete reference guide, that can be obtained from your sales representative.

OPSYS=DOS|MVS|MVS/XA|MVS/SP
The OPSYS keyword defines the operating system where the specified product is to reside. For all products except VTAM-EXPRESS, you need only to specify DOS or MVS. For the MVS version of EXPRESS you must specify either MVS/XA or MVS/SP. FOR MVS/ESA, specify MVS/XA.

PRODUCT=WINDOWS|JUGGLER|SVT|HELP|FILESERV|VWINDOWS|EXPRESS

The PRODUCT keyword identifies the product to be installed, or for which documentation is to be printed. A brief description of each product is contained in Appendix D.

The full product name corresponding to each keyword is:

WINDOWS	-	CICS-WINDOWS
JUGGLER	-	CICS-JUGGLER
HELP	-	HELP-WINDOWS
FILESERV	-	CICS-FILESERV
VWINDOWS	-	VTAM-WINDOWS
EXPRESS	-	VTAM-EXPRESS

SYSLIB=??????.??????

This is an optional keyword and is intended for MVS use only. SYSLIB is used to specify the SYSLIB for the IEWL execution. If used, this keyword **must** be specified in the CREATE install step.

SYSLMOD=??????.??????

This is an optional keyword and is intended for MVS use only. SYSLMOD is used to specify the SYSLMOD for the link edit of the CICS-JUGGLER programs. If used, this keyword **must** be specified in the CREATE install step.

SYSUT1=??????.??????

This is an optional keyword and is intended for MVS use only. SYSUT1 is used to specify the PDS to place the source and macro catalogs. If used, this keyword **must** be specified in the CREATE install step.

JUGLFILE=xxxxx

The JUGLFILE keyword identifies the name of the DLBL or DD statement in the job control statements for this execution of STSINST that names the VSAM file for CICS-WINDOWS. The xxxxxx is the file ID of the DLBL statement (DOS) or DDNAME of the DD statement (MVS).

VTWOFILE=xxxxxx

The VTWOFILE keyword identifies the name of the DLBL or DD statement in the job control statements for this execution of STSINST that names the VSAM file for VTAM-WINDOWS. The xxxxxx is the file ID of the DLBL statement (DOS) or DDNAME of the DD statement (MVS).

INSTALLING THE PRODUCT CONTROL PASSWORD

The product control password is a special code which controls the authorized use of any product from Unicom Systems. During the trial evaluation period, the product control password defines the date at which the trial version of the product will expire and can no longer be used.

For permanent licensed users, the product password defines all of the CPUs where the product may be used. Passwords for lease or lease-purchase licenses contain an expiration date and CPU ID.

When you initially receive an installation tape, whether for a trial evaluation or not, the tape contains a temporary password that will keep the product from expiring for approximately 30 days. If you are a licensed user, part of the re-install procedure is to assemble your permanent password table after installing the new tape.

If you are evaluating the product and have received a new temporary password from your sales representative, you must assemble the password table with the new password to keep the product from expiring.

The following job control is used to assemble and catalog the password control table, STSPASS.

DOS/VSE USERS

```
// JOB PASSWORD ASSEMBLY
// LIBDEF PHASE,CATALOG=L.S                                   [Note 1]
// OPTION CATAL
  PHASE STSPASS,*                                             [Note 2]
// EXEC ASSEMBLY
                DC     CL6'nnnnnn'                             CPU ID
                DC     CL4'xxxx'                           Product ID
                DC     CL6'mmddyy'                    Expiration Date
                DC     CL16'xxxxxxxxxxxxxxx'                 Password
                .
                .
                .
                DC     CL6'nnnnnn'                             CPU ID
                DC     CL4'xxxx'                           Product ID
                DC     CL6'mmddyy'                    Expiration Date
                DC     CL16'xxxxxxxxxxxxxxx'                 Password
                END
/*
// EXEC LNKEDT
/&
```

MVS USERS

```
//PASSTBL             JOB     1.'ACCOUNT DATA',MSGCLASS=x
//ASM                 EXEC    PGM=IEV90,PARM='DECK'
//SYSPRINT            DD      SYSOUT=*
//SYSLIN              DD      DUMMY
//SYSUT1              DD      UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSPUNCH DD         UNIT=SYSDA,DISP=(,PASS),
//                            SPACE=(CYL,(1,1)),
//                            DSN=&&TEMP1,
//                            DCB=BLKSIZE=3200
//SYSIN               DD      *
                  DC  CL6'nnnnnn'                              CPU ID
                  DC  CL4'xxxx'                            Product ID
                  DC  CL6'mmddyy'                     Expiration Date
                  DC  CL16'xxxxxxxxxxxxx    '                Password
                      .
                      .
                      .
                  DC  CL6'nnnnnn'                              CPU ID
                  DC  CL4'xxxx'                            Product ID
                  DC  CL6'mmddyy'                     Expiration Date
                  DC  CL16'xxxxxxxxxxxxx    '                Password
                      END
                      /*
//LINK                EXEC    PGM=IEWL,PARM=(LIST,XREF),
//                            COND=(4,LT)
//SYSLIB              DD      DUMMY
//SYSPRINT            DD      SYSOUT=*
//SYSLIN              DD      DSN=&&TEMP1,
//                            DISP=(OLD,DELETE)
```

```
//SYSLMOD      DD   DSN=CICS.STEPLIB(STSPASS),        [Note 1]
//                  DISP=SHR                          [Note 2]
//SYSUT1       DD   UNIT=SYSDA,SPACE=(CYL,(1,1))
```

[Notes]:

1. The library containing the link-edited module must be a CICS load library.
2. The link-edited module name must be STSPASS

DATA FIELDS OF THE PASSWORD CONTROL TABLE

The password table is a series of define constant instructions. The same password table is used for all products from Unicom Systems.

There are four DC statements required for each product, or for each occurrence of the same product on multiple CPUs. There is no limit to the number of product entries that may be present. The four DC instructions define the following values:

1). The first entry in the table is a 6-position CPU ID. For temporary passwords, code this as all zeros. For licensed users, code the full 6-position CPU ID where the product will operate.

If you are a VM user and have multiple guest machines on the same CPU, you can get by with a single entry if the last five positions of the CPU ID on each guest machine is the same. If this is the case, code zero as the first digit, followed by the five identifying digits. If each guest CPU ID is different, you must make an entry for each unique CPU ID.

2). The second entry is a 4-position product ID. Codes are:

WNDO	-	CICS-WINDOWS
JUGL	-	CICS-JUGGLER
HELP	-	HELP-WINDOWS
FSRV	-	CICS-FILESERV
VTWO	-	VTAM-WINDOWS
EXPR	-	VTAM-EXPRESS

3). The third entry is the expiration date. For temporary passwords or for lease and lease-purchase licenses, you will be given an expiration date along with your password. That date must be coded here in MMDDYY format. For perpetual licenses, code this as all zeros.

4). The last entry is the password itself. You must define it as a 16-byte field, although the total number of characters in the password may not reach 16. Code the password that you have been given.

CUSTOMIZATION

Certain default options in CICS-JUGGLER may be modified by the CICS system programmer. This tailoring could be performed for security reasons or for ease of operation. The following list provides most of the options along with the name of the table with which they are associated:

1) Limit the maximum number of CICS-JUGGLER users at one time. (User Option table)

2) Limit the maximum virtual terminals per physical terminal to less than nine. (User Option Table)

3) Exclude certain terminals or users from being able to use **CICS-JUGGLER**. (Terminal Exclusion Table)

4) Exclude certain transactions from recognizing the "Hot" keys. (Transaction Exclusion Table)

5) Force **CICS-JUGGLER** termination at CICS sign-on and sign-off. (User Option Table)

6) Eliminate the Pseudo Terminal ID option. (User Option Table)

7) Control the specification of Pseudo Terminal IDs. (User Option Table, Auto-Init Table, PSGNXIT statement in the JUGLTBL macro)

8) Define certain programs or transactions as "single-occurring", meaning that they can only be active in one virtual terminal per physical terminal at a time. (Single Occuring Transactions Table, Single Occuring Programs Table)

9) Prevent operators from terminating **CICS-JUGGLER** without first terminating the transactions in all virtual terminals. (User Option Table)

10) Define transaction codes other than CSSF and CSSN as sign-off or sign-on transactions. (Signoff Transactions Table, Signon Transactions Table)

11) Establish security for certain JUGL commands. (User Option Table)

12) Pre-define any or all start-up options of CICS-JUGGLER for each terminal. (Profile Table)

13) Provide support for multiple Interactive Interface sessions for DOS/VSE users. (Profile Table)

14) Suppress the terminal READ BUFFER command. (User Options Table, Profile Table)

15) Suppress the User Configuration display at JUGL,ON time. (User Options Table)

16) Maintain the same terminal user area in all virtual terminals. (User Options Table)

17) Specify a time interval, after which all or selected conversational tasks which have had no activity will be automatically purged. (User Options Table)

18) Define a user transaction to be automatically started at a terminal following CICS-JUGGLER activation and when first toggling into an empty session. (Application Startup Table)

19) Define a table of transactions which can not be run if CICS-JUGGLER is active on the terminal. (Stopped Transactions Table)

20) Designate a message log Transient Data queue for CICS-JUGGLER to use. (User Options Table)

21) Disable the Temporary Storage key modification feature. (User Options Table)

22) Define a user exit program for generating Pseudo Terminal IDs. (PSGNXIT statement in the JUGLTBL macro)

23) Associate a different VSAM file name with an alternate transaction code. (File Table)

CONVERTING THE JUGLTBL MACRO TO THE ONLINE VERSION

In past releases, all customizational options were specified by the use of a User Option Table which had to be coded by means of the JUGLTBL macro statements, assembled, link-edited and included in CICS by means of a PPT entry. Now, nearly all of the customization options can be performed interactively, on-line. The JUGLTBL macro is still provided with the package and must be used for three options, which are the name of the file where the on-line customization tables are to be kept and to enable the use of two user exits, explained later in this section.

Previously assembled customization tables may be converted to the on-line format simply by entering the JAUX Transaction (explained later in this section, under *THE AUXILIARY FUNCTIONS TRANSACTION*), which if no dynamic options are present, will display a screen asking if you wish to migrate the old JUGLTBL macro program to the newer dynamic version. If you answer YES, the table will be migrated and the Auxiliary Functions Transaction menu will be displayed. Subsequent invocations of this transaction will display the menu without asking to migrate. Once migration has been performed, you can remove the PPT table entry for JUGLTBL. If you choose not to migrate the table, the program will return to CICS.

CICS-JUGGLER will operate without converting the old JUGLTBL macro to the newer dynamic version; however, we strongly recommend conversion. The dynamic process is much more friendly, and nearly all changes made to the dynamic table will take effect immediately without the need to interrupt any operators that may currently be operating under CICS-JUGGLER. In addition, conversion from much older releases may not always be supported.

STEPS OF CONVERTING THE JUGLTBL MACRO

If migrating the JUGLTBL macro to the dynamic version is desired at this time, follow these steps:

1) Ensure that CICS-JUGGLER is **not** initialized. This may be verified by using CEMT to see if program JUGGLER is resident. If it is resident, you must issue the JUGL,BACKOUT command.
2) NEWCOPY all CICS-JUGGLER programs.
3) Ensure that the JUGLTBL program does have a PPT entry and it is enabled.
4) Ensure that no "O" (the character, not zero) prefix records exist in the JUGLFIL. This may be verified with the command **CECI READ DATASET(JUGLFIL) RIDFLD(O) GENERIC**. There must **not** be any option records, as this is one check that is performed by CICS-JUGGLER in the migration process. The JUGLFIL is distributed without any option records.
5) ENTER the **JAUX** transaction code. At this time, a screen should display asking if you wish to migrate the JUGLTBL macro.
6) If you selected the migrate option, the records will be converted and written to the JUGLFIL.

If the migration was successful, you may remove the JUGLTBL program from the PPT. If for some reason the migration was not successful, or if after following the above instructions, you were not prompted for migration, contact Unicom Systems technical support as explained in the appendix.

JUGLTBL MACRO STATEMENT FORMAT

Following are all the possible operands of the JUGLTBL macro, along with an explanation of each. This table must be coded if any of the three operands are required for your environment, otherwise it is not necessary. As explained above, nearly all options of the JUGLTBL macro are now controlled through on-line customization, so the remaining customization options are explained in *ON-LINE CUSTOMIZATION*, later in this section.

Underlined values listed are the default values if the associated operand is omitted. Values separated with a vertical bar (|) indicate mutually exclusive values that may be coded.

```
JUGLTBL       OPTFILE=(JUGLFIL | symbol),                              X
               PSGNXIT=[NO | symbol],
```

OPTFILE This is used to specify the name of the file to be used by CICS-JUGGLER for saving customization options and other internal control records. The default is "JUGLFIL". If you wish to use a different file name, you must code the name of the file that was created during the installation process.

PSGNXIT This is used to specify the name of a user exit program that will get control during a JUGL,ON or JUGL,INIT command when CICS-JUGGLER is ready to generate the Pseudo Terminal IDs. This exit program can generate a unique terminal ID for each virtual terminal using any method desired. If this feature is desired, code the name of your program, else code "NO" or omit.

For more information, see WRITING A USER EXIT FOR CICS-JUGGLER in section 14 - USER EXITS AND PROGRAM INTERFACES.

INSTALLATION OF THE JUGLTBL MACRO

The following JCL examples can be used to assemble and link-edit the JUGLTBL macro table for CICS-JUGGLER.

DOS Assembly and Link-Edit

```
// JOB JUGLTBL ASSEMBLY
// LIBDEF CL,TO=CICSCL                                                [Note 3]
// OPTION CATAL
  PHASE  JUGLTBL,*
// EXEC ASSEMBLY
                  JUGLTBL     OPTFILE=xxxxxxxx,                              X
                        PSGNXIT=xxxxxxxx
                  END
/*
// EXEC LNKEDT
/&
```

MVS Assembly and Link-Edit

```
//JUGLTBL      JOB     1.'ACCOUNT-DATA',MSGCLASS=X
//ASM          EXEC    PGM=IEV90,PARM='DECK'
//SYSLIB       DD      DSN=CICS.MACLIB,DISP=SHR                        [Note 1]
//             DD      DSN=SYS1.MACLIB,DISP=SHR                        [Note 2]
//SYSPRINT     DD      SYSOUT=*
//SYSLIN       DD      DUMMY
//SYSUT1       DD      UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSPUNCH     DD      UNIT=SYSDA,DISP=(,PASS),
//                     SPACE=(CYL,(1,1)),
//                     DSN=&&TEMP1,
//                     DCB=BLKSIZE=3200
//SYSIN        DD      *
```

```
                JUGLTBL ATCHXIT=xxxxxxxx,                                        X
                           OPTFILE=xxxxxxxx,                                     X
                           PSGNXIT=xxxxxxxx
                END
/*
//LINK          EXEC    PGM=IEWL,PARM=(LIST,XREF),
//                      COND=(4,LT)
//SYSLIB        DD      DUMMY
//SYSPRINT      DD      SYSOUT=*
//SYSLIN        DD      DSN=&&TEMP1,DISP=(OLD,DELETE)
//SYSLMOD       DD      DSN=CICS.LOADLIB(JUGLTBL),                        [Note 3]
//                      DISP=SHR
//SYSUT1        DD      UNIT=SYSDA,SPACE=(CYL,(1,1))
```

[Notes]:

1. Macro library for CICS (where JUGLTBL macro was loaded)
2. MVS System macro library
3. Core-Image / Load library for CICS applications

ON-LINE CUSTOMIZATION

There are fifteen categories of customization options that can be performed on-line These are:

1) Define the options of the "User Option" table. Most of the global options that were previously supplied by the JUGLTBL macro are now defined here.

2) Define a File table. This table is used when multiple CICS-JUGGLER VSAM files are to be used in one CICS system. This might be desirable when one or more different transaction codes (other that JUGL) are to be used, so that all users of a particular transaction code will use the same VSAM file, while other users will access a different file.

3) Customize "User Profiles". This table is used to pre-define some or all of the start-up options for a given terminal, or for all terminals. This enables operators to have pre-defined CICS-JUGGLER options that differ from or augment the Global User Option settings.

4) Define an "Auto-Init" table. This table is used by CICS-JUGGLER to associate an operator with a User Profile when a JUGL,ON command has been issued by that operator. In addition, this table can also be used to assign preselected pseudo terminal IDs to that operator.

5) Define an "Application Startup" table. This table provides a means of starting a user transaction automatically in each virtual terminal. You may define the same or different transactions in each session, and you may define data to be passed to the application program to vary the operation in each session.

6) Define a "Terminal Exclusion" table. This table is used to define one or more terminal IDs or operators that are to be excluded from using CICS-JUGGLER or included as the only terminals that can use CICS-JUGGLER.

7) Define a "Transaction Exclusion" table. This table is used to define one or more transaction codes that are to be excluded from recognizing the hot keys or included as the only transactions that will recognize the hot keys.

8) Define a "Sign-off Transaction" table. This table is used to define one or more transaction codes that designate a sign-off transaction, if you are using a transaction codes other than CSSF or CESF.

9) Define a "Sign-on Transaction" table. This table is used to define one or more transaction codes that designate a sign-on transaction, if you are using transaction codes other than CSSN OR CESN.

10) Define a "Single Occurring Transaction" table. This table is used to define one or more transaction codes that are "single-occurring", meaning it can only be active in one virtual terminal per physical terminal at a time.

11) Define a "Single Occurring Program" table. This table is used to define one or more program names which are "single-occurring".

12) Define a "Stopped Transactions" table. This table is used to define one or more transaction codes that will not be allowed to execute if CICS-JUGGLER is active on the terminal.

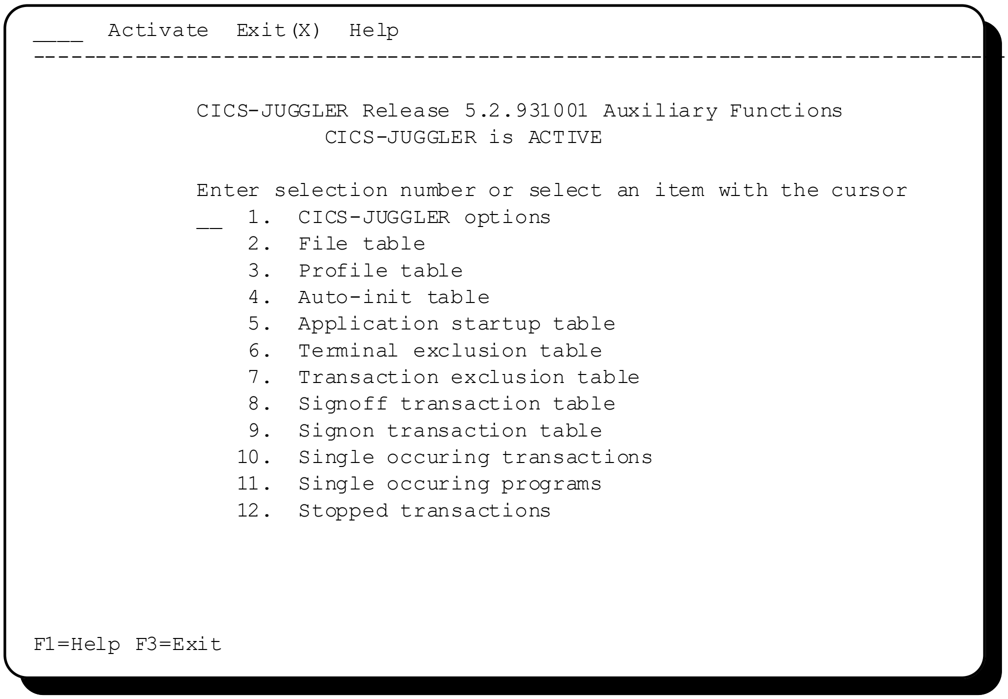
The relationships of the various on-line customization tables are as follows:

1) The User Options table specifies the global defaults.
2) The User Profile overrides or augments the User Option table for one or more operators. It may also be used to link the 'owner(s)' of the profile to a Start Transaction table.
4) The Auto-Init table links operators and terminals to their respective User Profiles.
5) The Sign-on and Sign-off tables are used to define sign-on and sign-off transactions to CICS-JUGGLER, which may use the information when working with certain customization options that deal with signing-on and/or signing-off.
6) The remaining tables have specific functions that do not directly hinge on other tables.

The dynamic customization options are accessed through the use of the Auxiliary Functions transaction - **JAUX**, which is explained in more detail in the following discussion.

THE AUXILIARY FUNCTIONS TRANSACTION

The Auxiliary Functions transaction is used for specifying nearly all the customization options for CICS-JUGGLER. To invoke the Auxiliary Functions transaction, key the transaction code JAUX from a clear screen. Upon pressing ENTER the Auxiliary Functions menu will appear:



```
____  Activate  Exit(X)  Help
-----------------------------------------------------------------------------

             CICS-JUGGLER Release 5.2.931001 Auxiliary Functions
                       CICS-JUGGLER is ACTIVE

             Enter selection number or select an item with the cursor
             __  1.  CICS-JUGGLER options
                 2.  File table
                 3.  Profile table
                 4.  Auto-init table
                 5.  Application startup table
                 6.  Terminal exclusion table
                 7.  Transaction exclusion table
                 8.  Signoff transaction table
                 9.  Signon transaction table
                10.  Single occuring transactions
                11.  Single occuring programs
                12.  Stopped transactions




F1=Help F3=Exit
```

[Note]: If the dynamic user options table record is not found in the JUGLFIL and a JUGLTBL program is found in this CICS, a different screen will display asking if you wish to migrate the old JUGLTBL macro to the newer dynamic customization version. If you choose to migrate, the JUGLTBL macro will be converted and then the above screen will display.

ENTERING COMMANDS AT THE AUXILIARY FUNCTIONS MENU

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

ENTER View and optionally modify the table that was specified in the entry field, or the table to which the cursor was pointing when ENTER was pressed

PF1 HELP Display a Help window for information pertaining to the field that the cursor is pointing when PF1 is pressed.

PF3 EXIT Exit the Auxiliary Functions transaction.

ACTION BAR FUNCTIONS AT THE AUXILIARY FUNCTIONS MENU

The following actions are available from the action bar:

ACTIVATE **Start CICS-JUGGLER** This has the same effect as issuing a JUGL,START command. This is used to cancel a previous STOP command and allows operators to perform a JUGL,ON command.

Stop CICS-JUGGLER This has the same effect as issuing a JUGL,STOP command. This will prevent operators from completing a JUGL,ON command until a START command is issued

Backout CICS-JUGGLER This has the same effect as issuing a JUGL,BACKOUT command. This is used for completely deactivating JUGGLER, usually to allow the application of a PTF or to allow JUGGLER to be newcopied.

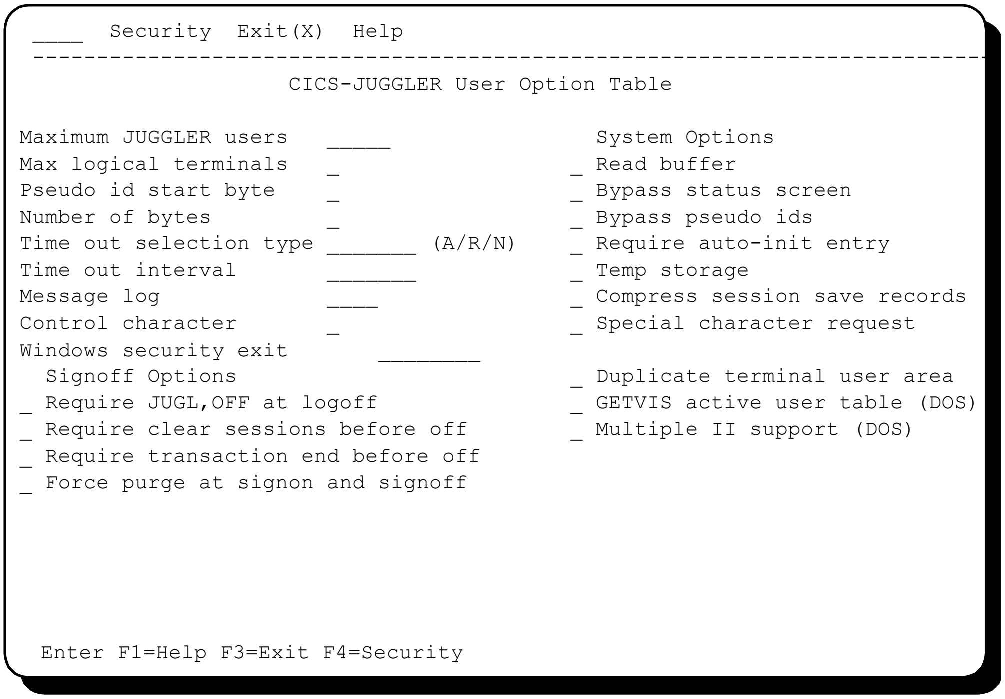
For more information on START, STOP and BACKOUT, please refer to section 13 - *SPECIAL-PURPOSE COMMANDS.*

EXIT **End Transaction** This has the same effect as PF3, which simply exits to CICS.

THE USER OPTIONS TABLE

The User Options table is used for setting most of the Global User Options available for CICS-JUGGLER.

To display the User Options table, from the Auxiliary Functions menu key a "1" and press ENTER. This screen will appear generally as follows:



```
 ____  Security  Exit(X)  Help
 ---------------------------------------------------------------------------
                     CICS-JUGGLER User Option Table

Maximum JUGGLER users   _____                  System Options
Max logical terminals   _                    _ Read buffer
Pseudo id start byte    _                    _ Bypass status screen
Number of bytes         _                    _ Bypass pseudo ids
Time out selection type _______ (A/R/N)      _ Require auto-init entry
Time out interval       _______              _ Temp storage
Message log             ____                 _ Compress session save records
Control character       _                    _ Special character request
Windows security exit       ________
  Signoff Options                            _ Duplicate terminal user area
_ Require JUGL,OFF at logoff                 _ GETVIS active user table (DOS)
_ Require clear sessions before off          _ Multiple II support (DOS)
_ Require transaction end before off
_ Force purge at signon and signoff




  Enter F1=Help F3=Exit F4=Security
```

You can change any value where the cursor will stop when the TAB key is pressed (except the PF key line). The following discussion describes each field.

FIELDS OF THE ON-LINE USER OPTION TABLE

BYPASS PSEUDO IDS

This field is used to specify whether pseudo terminal IDs are to be generated. It applies to any terminal using CICS-JUGGLER, which does not have pseudo terminal IDs defined in an Auto-Init table entry, either specifically or generically. It also applies if CICS-JUGGLER is activated with the JUGL,INIT command instead of the JUGL,ON command.Values are:

ON No additional terminal IDs are needed. CICS-JUGGLER will not allow operators to enter pseudo terminal IDs upon start-up.

OFF CICS-JUGGLER will generate pseudo terminal IDs as specified in the PSEUDO ID START, NUMBER OF BYTES and SPECIAL CHARACTER REQUEST fields or when ready to generate pseudo IDs.

BYPASS STATUS SCREEN

This option is used to specify whether the User Configuration screen, which is normally produced when a JUGL,ON, JUGL,INIT or JUGL,OFF command is issued, is to be bypassed when using the JUGL,ON or JUGL,OFF commands.

One purpose of bypassing the User Configuration Display would be when activating and terminating CICS-JUGGLER under program control (see *INITIALIZING CICS-JUGGLER FROM A USER PROGRAM)*. It may be that the user program produces a display after linking to the JUGGLER program. Without this option, the User Configuration Display would appear on the terminal as an intermittent flash prior to the user program display, which is usually not desired.

This option will not affect whether the User Configuration screen is displayed in response to the JUGL, JUGL,INIT or JUGL,INQ commands.

This is an ON/OFF type option:

ON The User Configuration screen is to be bypassed.

OFF The User Configuration screen is not to be bypassed.

COMPRESS SESSION SAVE RECORDS

This is used to specify whether the record held in either Temporary Storage or MAIN storage, for each terminal session, will be compressed by removing all repetitive characters. This can result in significant savings of file and memory space since these records contain an image of the screen for each session.

ON Compress the session save records.

OFF Do not perform any compression algorithms on the session save records.

CONTROL CHARACTER

This is used to specify a character to be used with control character commands such as Cut and Paste.

DUPLICATE TERMINAL USER AREA

This is used to specify whether the Terminal User Area will be common to all sessions of a terminal using CICS-JUGGLER.

It is often the case that an application system will store information in the Terminal User Area (user security information, etc.) which must be the same in all sessions. Without this option, the user area is initially set for all sessions to whatever it contains at JUGL,ON time, then maintained from session to session. If an application in one session changes the value, the original value is kept in all other sessions.

With this option, the Terminal User Area is not restored when toggling into a session. Thus, whatever value was placed in it by the previous application is carried into the next session. The only way to clear or reset the Terminal User Area is to JUGL,OFF and back on. This is an ON/OFF type option:

ON CICS-JUGGLER will maintain a common Terminal User Area in all sessions of a terminal.

OFF CICS-JUGGLER will not make any attempt to maintain a common Terminal User Area.

FORCE PURGE AT SIGNON AND SIGNOFF

This is used to specify whether CICS-JUGGLER is automatically terminated with a PURGE command before a new sign-on is accepted and before a sign-off is performed. This is an ON/OFF type option:

ON CICS-JUGGLER will be automatically terminated with a PURGE command before a new sign-on is accepted, and before a sign-off is allowed to complete. This means that any conversational transactions that are active on the terminal are abnormally ended with an AKCS abend.

OFF CICS-JUGGLER will not be automatically purged before a new sign-on is accepted or sign-off is performed.

GETVIS ACTIVE USER TABLE (DOS)

This option applies to VSE only. The Active User table is a dynamic table with an entry for each physical terminal for which CICS-JUGGLER is active. It requires approximately 256 bytes of storage per entry. This option specifies whether the table entries are to be contained in the GETVIS area or not. This is an ON/OFF type option:

ON Each entry will reside in the GETVIS area CICS partition. If there is insufficient GETVIS storage, all entries beyond that point will use DSA storage.

OFF All entries will reside in DSA storage.

MAX LOGICAL TERMINALS

This is used to establish the maximum number of virtual terminals that can be specified on any one physical terminal. This value will override a larger specification on the User Profile Table. Values are:

number A number from 2 to 9, inclusive.

MAXIMUM JUGGLER USERS

This is used to establish the maximum number of terminals in this CICS that can use CICS-JUGGLER at any one time. Values for this field are:

number A number from 1 to 34463, inclusive.

MESSAGE LOG

This is used to designate a 4-character Transient Data queue ID where you wish to route an activity log of messages from CICS-JUGGLER. Coding a destination ID establishes that you want a message to be sent to this destination every time a JUGL command is issued by a terminal operator.

If a destination ID is specified, a record will be written to the destination queue each time one of the following commands is issued from any terminal:

BACKOUT, DEBUG, INIT, MAIN, OFF, ON, PURGE, START, STOP, TEMP

The output message will take the following form:

JUGL1419. CICS-JUGGLER xxxxxx, TERMINAL yyyy, USER zzz, TIME hh:mm:ss

In the message, xxxxxx is the command (ON, OFF, PURGE, etc.), yyyy is the terminal ID where the command was given, zzz is the Operator User ID at that terminal and hh:mm:ss is the current time of day.

The Transient Data queue name may be any valid destination ID in the Destination Control Table (DCT). This includes the normal CICS statistics and terminal message queues, CSMT and CSTL.

[Note]: If the specified destination ID is invalid, that is, it is not defined in the DCT, CICS-JUGGLER does not issue any warning message, the MESSAGE LOG specification is simply ignored. Values are:

symbol This is the 4 digit Transient Data queue ID where you wish to route activity log messages from CICS-JUGGLER.

NO No activity log will be produced.

MODIFY TEMP STORAGE KEYS

This is used to specify whether Temporary Storage keys containing terminal IDs are to be modified by the session number prior to reading or writing a Temporary Storage record. This exit is provided to ensure that Temporary Storage records created and retrieved by application programs are not duplicated when the same transaction is run in multiple virtual terminals on one physical terminal. Is pseudo terminal IDs are in use, this option should not be used. This is an ON/OFF type option:

ON Modify the key with the session number before reading or writing to the Temporary Storage record. By selecting this option, it will be possible in most cases to eliminate the use of pseudo terminal IDs and use the same terminal ID in all sessions.

OFF Do not modify the Temporary Storage key. If you will be using pseudo terminal IDs all of the time, the Temporary Storage Modification exit is unnecessary and therefore should not be selected. This will eliminate the overhead of an unnecessary global exit.

For more information on Temporary Storage keys, please refer to the discussion on *PSEUDO TERMINAL ID CONSIDERATIONS* in the section entitled *SPECIAL CONSIDERATIONS.*

MULTIPLE II SUPPORT (DOS)

This option applies to VSE only. This parameter sets the global support level for the multiple interactive interface feature. This is an ON/OFF type option:

ON If selected, the specification of the VSE. Int. Interface of the User Profile table will be honored.

OFF If not selected, there can be only one session per physical terminal containing the interactive interface.

NUMBER OF BYTES

This specifies the number of bytes of the pseudo terminal ID to be duplicated from the real terminal ID when CICS-JUGGLER is generating pseudo IDs. values are:

number A number from 1 to 3, inclusive.

[Note]: The sum of PSEUDO ID START and NUMBER OF BYTES can not exceed 4.

PSEUDO ID START BYTE

This specifies the starting position in each pseudo terminal ID to be duplicated from the real terminal ID when CICS-JUGGLER is generating pseudo IDs. Values are:

number A number from 1 to 4, inclusive.

[Note]: The sum of PSEUDO ID START and NUMBER OF BYTES can not exceed 4.

READ BUFFER

This is used to specify whether CICS-JUGGLER performs a terminal Read Buffer command when you press the toggle key or the window key to enter window mode. Certain non-IBM terminals will sometimes not support the Read Buffer command, causing ATNI abends when the toggle key is pressed. This is the case if you are using VTAM PASSTHRU for your terminal. This is an ON/OFF type option:

ON CICS-JUGGLER will issue a Read Buffer command.

OFF CICS-JUGGLER will not issue a Read Buffer command. You need to operate with READ BUFFER=NO if any of the following conditions is true:

1). You want to use the VIEW function of the SYS display and always see the latest output screen regardless of when the operator last toggled.

2). You wish to improve the response time for remote terminals when a toggle occurs.

[Note]: This option is used to globally enable or disable READ BUFFER. If selected, READ BUFFER can be overridden in the User Profile. If not selected here, the READ BUFFER option in the User Profile is ignored (READ BUFFER will not occur).

REQUIRE AUTO-INIT ENTRY

This is used to specify whether you wish to exclude any operators that are not specifically identified in the Auto-Init table from performing a JUGL,ON. This is an ON/OFF type option:

ON Only those operators specified in the Auto-Init table will be allowed to perform a JUGL,ON command.

OFF All operators may perform a JUGL,ON.

REQUIRE CLEAR SESSIONS BEFORE OFF

This is used to specify whether users will need to end all tasks and clear all sessions before they are allowed to terminate CICS-JUGGLER. This is an ON/OFF type option:

ON Users will be required to end all tasks and clear all sessions, except the session from which the OFF command is entered

OFF Users will not be required to end all tasks and clear all sessions in order to terminate CICS-JUGGLER.

REQUIRE TRANSACTION END BEFORE OFF

This is used to specify whether operators will be required to end all transactions in each session before performing a JUGL,OFF command. This is an ON/OFF type option:

ON The operator must end all transactions, but is not neccessary to clear the screens, in order to execute a JUGL,OFF command.

OFF The operator can issue a JUGL,OFF command to terminate CICS-JUGGLER while interactive tasks are in progress.

[Note]: Regardless of the coding of this option, conversational tasks must be terminated in order to perform a JUGL,OFF.

REQUIRE JUGL,OFF AT LOGOF

This is used to specify whether the operator will be required to deactivate CICS-JUGGLER on their terminal before they are allowed to logoff from CICS, using the logoff transactions coded in the "Sign-on/Sign-off Transaction" table. If FORCE PURGE AT SIGNON/SIGNOFF is selected, this option is ignored. This is an ON/OFF type option:

ON JUGGLER must be terminated to perform logoff.

OFF Upon logoff, the user will be prompted with an option to terminate JUGGLER.

SECURED COMMANDS

This portion of the User Option Table is accessed by pressing PF4 from the User Option Table.

These fields are used to specify whether the associated JUGL command is to be a secured JUGL function. The commands that may be secured are:

BACKOUT, DEBUG, INIT, MAIN, PURGE, START, STOP, SYS, TEMP

If any of the above commands are selected, you must define an additional transaction code in the Program Control Table for CICS, as follows:

```
DFHPCT  TYPE=ENTRY,TRANSID=JUSC,PROGRAM=JUGGLER,          X
        TWASIZE=0,TRANSEC=xx
```

Use a different security key for the JUSC transaction than for the JUGL transaction, one that only systems or other MIS personnel are authorized to use.

Now, if any of the listed commands are selected and issued with the JUGL transaction code, the following message will display:

JUGL1417. USER IS NOT AUTHORIZED FOR THIS COMMAND

The command will only be accepted if issued with the JUSC transaction code. (i.e. JUSC,STOP) These fields are ON/OFF type options:

ON The associated command is to be secured.

OFF The associated command is <u>not</u> to be secured.

SPECIAL CHARACTER REQUEST

This is used to specify whether special characters may be used while generating pseudo terminal IDs. For more information, please refer to the PSEUDO IDs operand of the Auto-Init table, later in this section. This is an ON/OFF type option:

ON Use special characters when generating pseudo terminal IDs.

OFF Do <u>not</u> use special characters when generating pseudo terminal IDs.

TEMP STORAGE

This is used to specify whether Temporary Storage is to be used instead of MAIN storage above the 16M line for saving terminal data. This is an ON/OFF type option:

ON Use Temporary Storage instead of MAIN storage.

OFF Do <u>not</u> use Temporary Storage. Instead, use MAIN storage above the 16M line.

TIME OUT INTERVAL

This specifies the time interval for the conversational transaction time out. This field is only valid if TIME OUT SELECTION is <u>not</u> coded "ALL". Entries are:

number A 1 to 6 digit number in 'hhmmss' format. That is, the first 2 digits specify from 00 to 24 hours, the second two digits specify from 00 to 60 minutes, and the last 2 digits specify from 00 to 60 seconds. If less then 6 digits is coded, the number is filled with high-order zeros.

YES This specifies that the interval will be the interval that is specified in the PCT. Coding YES is only valid if TIME OUT SELECTION TYPE=RTIMOUT.

When a timeout purge of a transaction occurs, the transaction is abnormally terminated with an AKCS abend code and the message

JUGL1515. TASK xxxx PURGED BY JUGL TIMEOUT

replaces the saved screen display in that session.

TIME OUT SELECTION TYPE

This parameter provides a means of automatically terminating all or certain conversational transactions when they have been inactive for a specified time. This parameter provides an alternative to the

RTIMOUT operand of the Program Control Table, which will not work on a transaction that has been toggled out of. Values are:

ALL After toggling out of any conversational transaction, if the operator does not toggle back into that application for the specified interval, the transaction will be purged.

RTIMOUT After toggling out of any conversational transaction which is specified in the PCT with RTIMOUT, if the operator does not toggle back into that application for the specified interval, the transaction will be purged.

NO No automatic purge of conversational tasks will take place.

[Note]: To use the TIMEOUT feature, the following entry must be present in the Program Control Table:

```
DFHPCT  TYPE=ENTRY,TRANSID=JTMO,PROGRAM=JUGGLER
```

ENTERING COMMANDS AT THE USER OPTIONS TABLE

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

ENTER Apply any changes that you have made to the table.

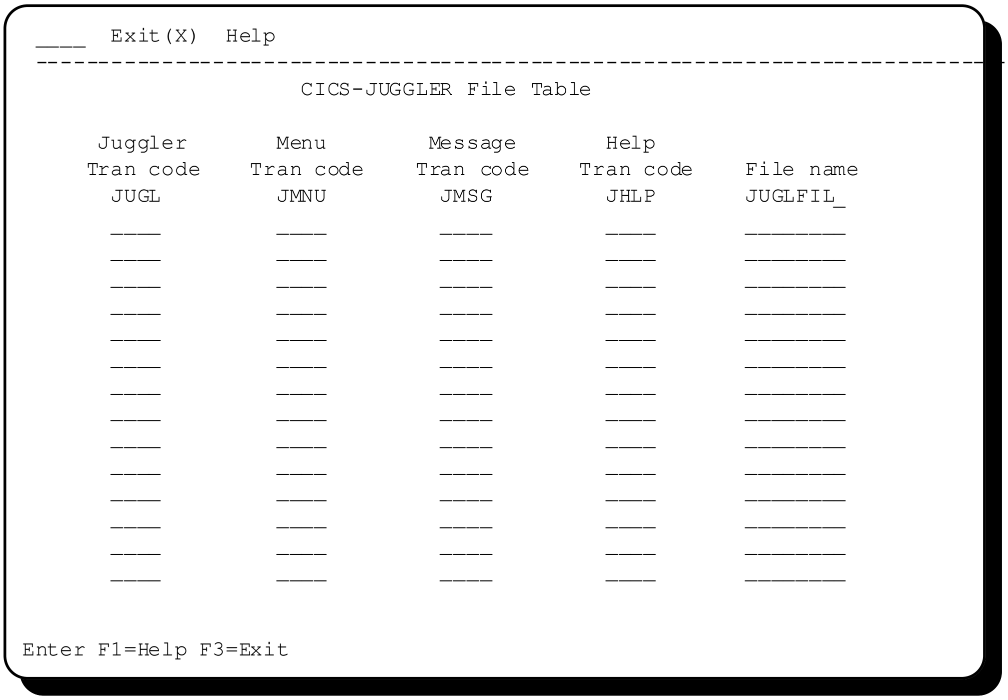
PF1 HELP Display a Help window for information pertaining to the field to which the cursor pointed when the PF key was pressed.

PF3 EXIT Return to the Auxiliary Functions menu.

PF4 SECURITY Display the Secured Commands entries.

THE FILE TABLE

The File table is used when multiple VSAM files are to be used within one CICS system. This might be desirable when one or more different transaction codes are to be used. This statement provides a means of associating a file name with a transaction code, so that all users of a particular transaction code will use the same VSAM file, while other users will access a different file. This statement is not necessary if 'JUGLFIL' is the only VSAM file in use, even though different transaction codes may be defined. To display the File table, from the Auxiliary Functions menu key a "2" and press ENTER. This screen will appear as follows:



```
____  Exit(X)  Help
-----------------------------------------------------------------------
                    CICS-JUGGLER File Table

     Juggler       Menu        Message       Help
    Tran code    Tran code    Tran code    Tran code     File name
      JUGL         JMNU         JMSG         JHLP        JUGLFIL_
      ____         ____         ____         ____        ________
      ____         ____         ____         ____        ________
      ____         ____         ____         ____        ________
      ____         ____         ____         ____        ________
      ____         ____         ____         ____        ________
      ____         ____         ____         ____        ________
      ____         ____         ____         ____        ________
      ____         ____         ____         ____        ________
      ____         ____         ____         ____        ________
      ____         ____         ____         ____        ________
      ____         ____         ____         ____        ________
      ____         ____         ____         ____        ________
      ____         ____         ____         ____        ________
      ____         ____         ____         ____        ________


Enter F1=Help F3=Exit
```

You can change any value where the cursor will stop when the TAB key is pressed (except the PF key line). The following discussion describes each field.

FIELDS OF THE FILE TABLE

FILE NAME This field defines the file name to be associated with the corresponding transaction code specified in the one of the TRAN CODE fields. Values are:

symbol Any valid VSAM file name. This should be the VSAM file that was created during the installation process.

[Note]: Multiple TRAN CODE fields on any one line of the table may be coded. For instance, if you wish to have both menu and message definitions in the same file, you may code the menu transaction ID, the message transaction ID and the file name on the same line.

[Note]: As described in the *INSTALLATION* section, using alternate transaction codes for CICS-JUGGLER does not require the definition of those transactions anywhere other than in the CICS PCT. The File table is only required if you want to use alternate VSAM file name(s).

JUGGLER TRAN CODE
This field is used to define the transaction code that is to be associated with the file name specified in the corresponding FILE NAME field. Values are:

symbol A valid transaction code that is used to invoke the sessions-management facilities of CICS-JUGGLER.

[Note]: The transaction code 'JUGL' does not need to be defined in a TRAN operand unless you intend to associate a file name other than JUGLFIL with it. In fact, by using alternate transaction codes, you do not need to define the transaction code 'JUGL' to CICS at all, if preferred.

MENU TRAN CODE
This field is used to define the Menu Definition transaction code that is to be associated with the file name specified in the corresponding FILE NAME field. Values are:

symbol A valid transaction code that is used to invoke the Menu Definition facility of CICS-JUGGLER.

MESSAGE TRAN CODE
This field is used to define the Message Broadcasting transaction code that is to be associated with the file name specified in the corresponding FILE NAME field. Values are:

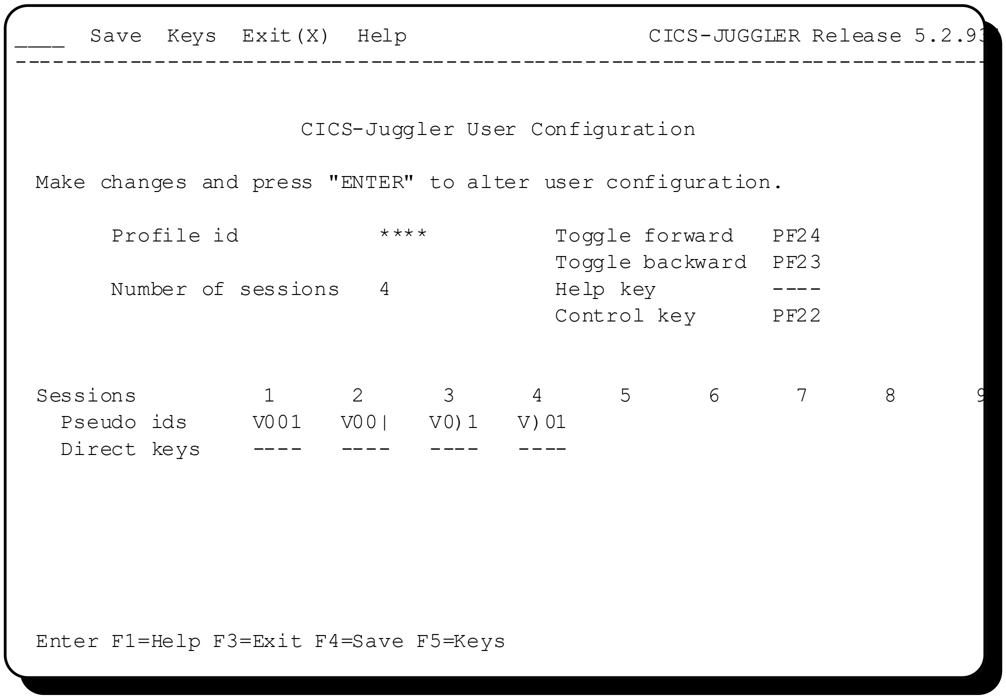
symbol A valid transaction code that is used to invoke the Message Broadcast facility of CICS-JUGGLER.

HELP TRAN CODE
This field is used to define the transaction code that is to be associated with the file name specified in the corresponding FILE NAME field. Values are:

symbol A valid transaction code that is used to invoke the help maintenance facilities of the HELP-WINDOWS feature.

THE USER PROFILE TABLE

The User Profile table is used to pre-define some or all of the start-up options for a given terminal, or for all terminals. This enables operators to have pre-defined CICS-JUGGLER options that differ from or augment the Global User Option settings. To display the User Profile table, from the Auxiliary Functions menu key a "3" and press ENTER. This screen will appear generally as follows:



```
____  Save  Keys  Exit(X)  Help                   CICS-JUGGLER Release 5.2.9
-------------------------------------------------------------------------------


                      CICS-Juggler User Configuration

 Make changes and press "ENTER" to alter user configuration.

       Profile id           ****          Toggle forward   PF24
                                          Toggle backward  PF23
       Number of sessions   4             Help key         ----
                                          Control key      PF22


 Sessions          1      2      3      4      5      6      7      8      9
   Pseudo ids     V001   V00|   V0)1   V)01
   Direct keys    ----   ----   ----   ----






 Enter F1=Help F3=Exit F4=Save F5=Keys
```

You can change any value where the cursor will stop when the TAB key is pressed (except the PF key line). The following discussion describes each field.

FIELDS OF THE USER PROFILE TABLE

AUTOSTART This specifies whether this profile will automatically activate CICS-JUGGLER for an operator when they sign on to CICS. Valid entries are:

YES Immediately after a CICS sign on is performed, CICS-JUGGLER will automatically perform a JUGL,ON command for operators with this profile.

NO CICS-JUGGLER will not be automatically started. The operator must perform a JUGL,ON command.

CONTROL KEY

This is used to specify a PF or PA key to be pressed that will invoke the CICS-JUGGLER Control Window. For more information on the Control Window, see section 02 - *OPERATION* under *THE CICS-JUGGLER CONTROL WINDOW.*

HELP KEY This is used to specify a PF or PA key to be used as the CICS-JUGGLER Help Access key. For more information on the Help Access key, see section 07 - *THE HELP-WINDOWS FEATURE*.

Valid entries are any PF or PA key or CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 11 - *SPECIAL CONSIDERATIONS*).

NUMBER OF SESSIONS

This is the number of sessions (virtual terminals) that an operator with this profile will have. Note that if this value exceeds the value coded for MAX LOGICAL TERMINALS on the User Options table, the maximum value will be used.

If this operand is omitted, the terminal operator will be prompted for the number of virtual terminals at initialization. Valid entries are:

NUMBER A number from 2 to 9, inclusive.

PROFILE ID This is the ID of which this profile will be referenced on the Auto-Init table. Note that the Profile ID can not be changed once the profile is created. To accomplish this you must first copy this profile to the desired Profile ID, then delete the old Profile. Valid entries are:

SYMBOLAny 1 to 4 character name.

PROTECT PROFILE

The User Configuration Display allows operators to change their configuration, and to save those changes in their profile. However if multiple operators are sharing the same profile, saving these changes would affect all operators associated with the profile. The Protect Profile option disables this feature so that the profile cannot be altered from the User Configuration Display. Values are:

YES Do not allow operators to alter their profile(s) from the User Configuration Display.

NO Allow operators to alter their profile(s) from the User Configuration Display.

READ BUFFER SUPPORT

This is used to specify whether CICS-JUGGLER issues a terminal Read Buffer command when the toggle key is pressed or the window key to enter window mode. This applies only to operators with this profile, and is only valid if READ BUFFER is coded "YES" in the User Option Table. Certain non-IBM terminals will sometimes not support the Read Buffer command, causing ATNI abends when the toggle key is pressed. This is the case if you are using VTAM PASSTHRU for your terminal. Valid entries are:

YES CICS-JUGGLER will issue a Read Buffer command for operators of this profile if READ BUFFER is coded "YES" in the User Option Table.

NO CICS-JUGGLER will not issue a Read Buffer command. You need to operate with READ BUFFER=NO if any of the following conditions is true:

1). You want to use the VIEW function of the SYS display and always see the latest output screen regardless of when the operator last toggled.

2). You wish to improve the response time for remote terminals when a toggle occurs.

SESSIONS DIRECT KEYS

Direct session toggle keys. If this feature is desired, code a PF or PA key for each virtual terminal present on this physical terminal. The keys correspond one for one with each virtual terminal. That is, the first key coded will transfer control directly to virtual terminal 1, the second to virtual terminal 2, etc.

You may omit one or more keys in the list by skipping that position. If this is done, the virtual terminal corresponding to that key position will not have a direct key assigned to it.

Valid keys are any PF or PA key and CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 11 - *SPECIAL CONSIDERATIONS)*. The key selected must not be the same as any other designated key.

See the section entitled *DIRECT SESSION KEY OPERATION* for an explanation of the use of Direct Session keys.

START TRANSACTION

This provides a means of starting a user transaction automatically in each virtual terminal by connecting to an entry in the Application Startup table. You may define the same or different transactions in each session, and you may define data to be passed to the application program to vary the operation in each session.

When used, the transaction specified for virtual terminal number 1 will automatically start when the JUGL,ON command is completed (the User Configuration screen normally produced by JUGL,ON is suppressed). The transactions corresponding to the remaining sessions will be automatically started the first time the operator toggles into that session. For more information, see *THE APPLICATION STARTUP TABLE*, later in this section.

To use automatic transaction starting, code the one to eight character APPL ID (as coded on the appropriate table entry on the Application Startup table) in this field.

If automatic transaction starting is not desired, omit this keyword.

TOGGLE BACKWARD KEY

Code the PF or PA key to be used to move "backward" from one virtual terminal to the next lower terminal number. Each time the Toggle-Backward key is pressed, control moves from the current virtual terminal to the previous one in sequence until terminal number 1 is reached, at which time control moves to virtual terminal number (NUMBER OF SESSIONS). Valid entries are:

Any PF or PA key or CSEL (see USING THE CURSOR-SELECT KEY AS A HOT KEY).

TOGGLE FORWARD KEY

This is the PF or PA key to be used to move "forward" from one virtual terminal to the next higher terminal number. Each time the Toggle-Forward key is pressed, control moves from the current virtual terminal to the next one in sequence until terminal number (NUMBER OF SESSIONS) is reached, at which time control moves to virtual terminal number one.

If the Toggle-Forward key is not desired, omit this field, in which case, the operator will be prompted for the Toggle-Forward key at JUGL,ON time. Valid entries are:

Any PF or PA key or CSEL (see USING THE CURSOR-SELECT KEY AS A HOT KEY in section 11 - SPECIAL CONSIDERATIONS).

VSE INT. INTERFACE
This operand establishes the method of operation for using multiple Interactive Interface selection panels with CICS-JUGGLER for DOS VSE users only. Valid entries are:

YES CICS-JUGGLER will duplicate the selection panel from the first session into all sessions and allow operation of the selection panel in all sessions with only one operator sign-on.

NO CICS-JUGGLER will allow the selection panel to be active in session 1 only. Transactions must be started using transaction codes in the other sessions.

SIGNON CICS-JUGGLER will require a different operator sign-on in each session and will support different selection panels in each session.

[Note]: If "YES" or "SIGNON" is coded for any profile, the JUGLVSP program must be installed.

For more information, see RUNNING CICS-JUGGLER WITH THE DOS VSE INTERACTIVE INTERFACE in the section entitled UNIQUE ENVIRONMENTS AND SPECIAL SITUATIONS.

ENTERING COMMANDS AT THE USER PROFILE TABLE

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

ENTER Apply any changes that you have made to the table.

PF1 HELP Display a Help window. If the cursor is positioned to an unprotected field, the help will pertain to that field, otherwise the help will pertain to the User Profile display.

PF2 FIND Initiate the FIND function, to display a different profile.

PF3 EXIT Return to the Auxiliary Functions menu.

PF4 ADD Add a new profile entry to the table.

PF5 COPY Copy this profile. This may be used when creating a new profile if it is advantageous to copy an existing profile, then later modify any differing fields as desired.

PF6 DELETE Delete this profile entry from the table.

PF7 BWD Browse backward to the previous table entry.

PF8 FWD Browse forward to the next table entry.

PF9 AUTOTBL Display the Auto-Init table. (Same as selection 4 from the Auxiliary Functions menu.)

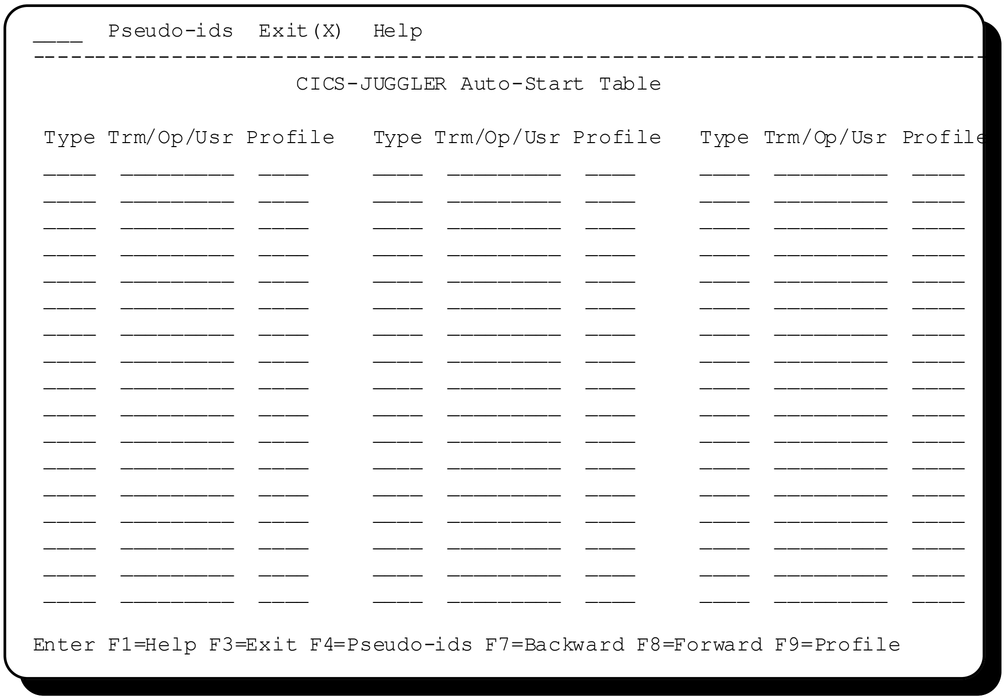
PF11 KEYS Display Function Key assignements. You may assign to open keys any function (e.g., Copy, Paste, Define) not previously defined in the profile.

THE AUTO-INIT TABLE

The Auto-Init table is used by CICS-JUGGLER to associate an operator with a User Profile when a JUGL,ON command has been issued by that operator. In addition, this table can also be used to assign preselected pseudo terminal IDs to that operator.

When performing a JUGL,ON, CICS-JUGGLER will use the profile of the first matching auto-init table entry; thus the relative position of auto-init table entries is crucial in determining the profile to be used. For instance, if a completely generic auto-init entry (solo asterisk) is the first entry in the table, that profile will be used for **all** operators that issue a JUGL,ON command. Usually, any generic or wildcard entries should appear at the bottom of the list.

To display the Auto-Init table, from the Auxiliary Functions menu key a "4" and press ENTER. This screen will appear generally as follows:



```
____  Pseudo-ids  Exit(X)  Help
------------------------------------------------------------------------------
                     CICS-JUGGLER Auto-Start Table

 Type Trm/Op/Usr Profile    Type Trm/Op/Usr Profile    Type Trm/Op/Usr Profile
 ____  _________  ____      ____  _________  ____      ____  _________  ____
 ____  _________  ____      ____  _________  ____      ____  _________  ____
 ____  _________  ____      ____  _________  ____      ____  _________  ____
 ____  _________  ____      ____  _________  ____      ____  _________  ____
 ____  _________  ____      ____  _________  ____      ____  _________  ____
 ____  _________  ____      ____  _________  ____      ____  _________  ____
 ____  _________  ____      ____  _________  ____      ____  _________  ____
 ____  _________  ____      ____  _________  ____      ____  _________  ____
 ____  _________  ____      ____  _________  ____      ____  _________  ____
 ____  _________  ____      ____  _________  ____      ____  _________  ____
 ____  _________  ____      ____  _________  ____      ____  _________  ____
 ____  _________  ____      ____  _________  ____      ____  _________  ____
 ____  _________  ____      ____  _________  ____      ____  _________  ____
 ____  _________  ____      ____  _________  ____      ____  _________  ____
 ____  _________  ____      ____  _________  ____      ____  _________  ____
 ____  _________  ____      ____  _________  ____      ____  _________  ____
 ____  _________  ____      ____  _________  ____      ____  _________  ____

Enter F1=Help F3=Exit F4=Pseudo-ids F7=Backward F8=Forward F9=Profile
```

You can change any value where the cursor will stop when the TAB key is pressed. The following discussion describes each field.

FIELDS OF THE AUTO-INIT TABLE

PROFILE This is the ID of the User Profile that is to be used for this table entry. Valid entries are any previously defined User Profile.

PSEUDO IDS Upon pressing PF4, the screen format will change to show the pseudo terminal IDs that are connected to each table entry. You may add new IDs or modify existing IDs in these fields. These fields may be coded in the following ways:

1). Code the explicit unique terminal IDs to be assigned to each of the virtual terminals. Code one 4-character ID per virtual terminal. Each ID must be unique, that is, it must not be present either in the Terminal Control Table (TCT) or specified as a Pseudo ID in any other entry.

    The first terminal uses the real terminal ID and is not present in the list. Thus, if NUMBER OF SESSIONS is specified as '4', the number of IDs to be coded is three. If NUMBER OF SESSIONS is omitted, the number of Pseudo IDs to be coded must equal the value coded for MAXIMUM VIRTUAL TERMINALS minus 1.

[Note]: You should not code specific terminal IDs unless the TYPE field specifies TERM and the Terminal ID is fully defined (non-generic).

2). If Pseudo IDs are to be automatically generated by CICS-JUGGLER, code '*GEN' in the first pseudo ID field. This will cause JUGGLER to generate unique pseudo IDs and bypass the pseudo ID prompt at initialization for the terminal.

When Pseudo IDs are generated by CICS-JUGGLER, the real terminal ID is used as the basis for constructing a unique ID for each virtual terminal. Each position of the real ID is copied to the corresponding position in the pseudo ID, then one character of the ID is set to uppercase or lowercase, or to a special character (see following illustration). The result is a unique ID for each virtual terminal that still resembles the real terminal ID.

The following illustration shows the position of the lowercase characters for each of the eight possible virtual terminals. (U=uppercase, L=lowercase).

Real terminal ID	UUUU Example:	TRMX
Virtual terminal 1	UUUL	TRMx
Virtual terminal 2	UULU	TRmX
Virtual terminal 3	ULUU	TrMX
Virtual terminal 4	LUUU	tRMX
Virtual terminal 5	LUUL	tRMx
Virtual terminal 6	LULL	tRmx
Virtual terminal 7	LLLL	trmx
Virtual terminal 8	ULLL	Trmx

When numeric characters are present in the real ID, some models of 3270 terminals will not properly display the lowercase numeric characters (3179, 3180 terminals). This does not cause any significant problems, but if you prefer not to use lowercase numerics, select SPECIAL CHARACTER REQUEST in the User Options table. This specifies the use of special characters in place of lowercase numerics in the terminal ID.

The special characters used are the characters corresponding to the shift position of most 3270 keyboards. They are:

'1'	=	'|'	(vertical bar)
'2'	=	'@'	(at sign)
'3'	=	'#'	(pound sign)
'4'	=	'$'	(dollar sign)
'5'	=	'%'	(percent sign)
'6'	=	'¢'	(cent sign)
'7'	=	'&'	(ampersand)
'8'	=	'*'	(asterisk)
'9'	=	'('	(left parenthesis)
'0'	=	')'	(right parenthesis)

You can also write your own user exit program to generate the pseudo IDs using any method that you prefer. See the discussion of *PSEUDO TERMINAL ID GENERATION EXIT* in section 14 - *USER EXITS AND PROGRAM INTERFACES*.

3). If Pseudo Terminal IDs are not to be assigned for this terminal code "NO" in the first field. This will cause each virtual terminal to use the real terminal ID.

4). If these fields are left blank, the operator will be prompted for the Pseudo IDs unless BYPASS PSEUDO IDS is selected in the User Option Table.

[Note]: If a pseudo ID generation exit is to be used (PSGNXIT in JUGLTBL), you must code "*GEN" in the first field.

[Note]: Pressing PF4 again will change the screen format back to the previous format (before PF4 was initially pressed).

TRM/OP/USR This field is used to supply the actual data that CICS-JUGGLER will use to identify an operator.

For TYPE=TERM
Code the terminal ID, as it is known to CICS. This keyword may be coded in any of the following four ways:

1) Code an individual statement for each terminal to be defined, coding the TERM keyword as the four-character terminal ID of the terminal to be configured.

2) Define a range of terminals to be configured using the parameters specified on this statement. Code the TERM operand as two terminal IDs separated by a hyphen. With this method, when a JUGL,ON command is performed at any terminal which has a terminal ID equal-to or greater than the first ID and equal-to or less than the second ID, that terminal will use the profile of this statement.

3) Define a generic definition using "wild-card" characters. Code the TERM operand with question marks (?) in one or more positions of the terminal ID. With this method, when a JUGL,ON command is performed at any terminal, if all positions in the terminal ID match the corresponding positions in the TERM operand where question marks are <u>not</u> coded, that terminal will use the profile of this statement.

4) Define a single statement that applies to all terminals. Code the TERM operand as "*". With this method, this statement applies to any terminal using **CICS-JUGGLER** which is not explicitly identified by another statement.

   If a solo asterisk is coded and other statements are present, this entry should be the last statement. This allows some terminals to be defined explicitly, using different start-up options than the generic entry.

For TYPE=OP
Code the operator name, as it appears in the TCT. This is the three-character CICS operator ID of the operator for which the configuration described by the corresponding profile applies. The operator ID keyword may be coded in any of the following four ways:

1) Code an individual statement for each operator to be defined, coding the three-character operator ID of the user to be configured.

2) Define a range of operators to be configured using the parameters specified on this statement. Code the two operator IDs separated by a hyphen. With this method, when a JUGL,ON command is performed at any terminal which has an operator ID equal-to or greater than the first ID and equal-to or less than the second ID, that terminal will use the profile on this statement.

3) Define a generic definition using "wild-card" characters. Code question marks (?) in one or more positions of the operator ID. With this method, when a JUGL,ON command is done at

any terminal, if all positions in the operator ID match the corresponding positions where question marks are not coded, that terminal will use the profile on this statement.

4) Define a single statement which applies to all users. Code the field as "*". With this method, this statement applies to any terminal using **CICS-JUGGLER** which is not explicitly identified by another statement.

   If a solo asterisk is coded and other statements are present, this entry should be the last statement. This allows some users to be defined explicitly, using different start-up options than the generic entry.

For TYPE=USER

Code the user ID, as it appears in the SNT. This is the eight-character CICS user ID of the operator for which the configuration described by the corresponding profile applies. The user ID keyword may be coded in any of the following ways:

1) Code an individual statement for each user to be defined, coding the eight-character user ID of the user to be configured.

2) Define a generic definition using "wild-card" characters. Code question marks (?) in one or more positions of the user ID. With this method, when a JUGL,ON command is done at any terminal, if all positions in the user ID match the corresponding positions where question marks are not coded, that terminal will use the profile on this statement.

3) Define a single statement which applies to all users. Code the field as "*". With this method, this statement applies to any user using **CICS-JUGGLER** which is not explicitly identified by another statement.

   If a solo asterisk is coded and other statements are present, this entry should be the last statement. This allows some users to be defined explicitly, using different start-up options than the generic entry.

TYPE This field is used to specify the type of data CICS-JUGGLER is to use to determine what profile to assign to an operator when a JUGL,ON command is issued. Valid entries are:

TERM This table entry is to associate the 4-character terminal ID with a User Profile.

OP This table entry is to associate the 3-character operator identifier with a User Profile.

USER This table entry is to associate the 8-character user identifier with a User Profile. This option is only valid for CICS releases 1.7 and later.

In addition to the entries that may be coded in this field, there are two commands that may be keyed in this field:

- The (**D**)elete command will delete an entry and shift all lower entries up into the vacated slot.
- The (**I**)nsert command will shift all lower entries down (including the entry in which the command was issued), which will create a vacant slot for inserting a new auto-init entry.

ENTERING COMMANDS AT THE AUTO-INIT TABLE

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

ENTER Apply any changes that you have made to the table.

PF1 HELP Display a Help screen for information on the field that the cursor was in when the PF key was pressed.

PF3 EXIT Return to the Auxiliary Functions menu.

PF4 PSEUDO IDS
View or modify the pseudo IDs that are coded for each Auto-Init table entry.

PF7 BACKWARD Browse backward to the previous page of table entries.

PF8 FORWARD Browse forward to the next page of table entries.

PF9 PROFILE Display the Profile for the entry that the cursor is pointing when PF9 is pressed. (Same as selection 2 from the Auxiliary Functions menu.)

In addition to the PF key functions, there are two commands that may be keyed in the TYPE field:

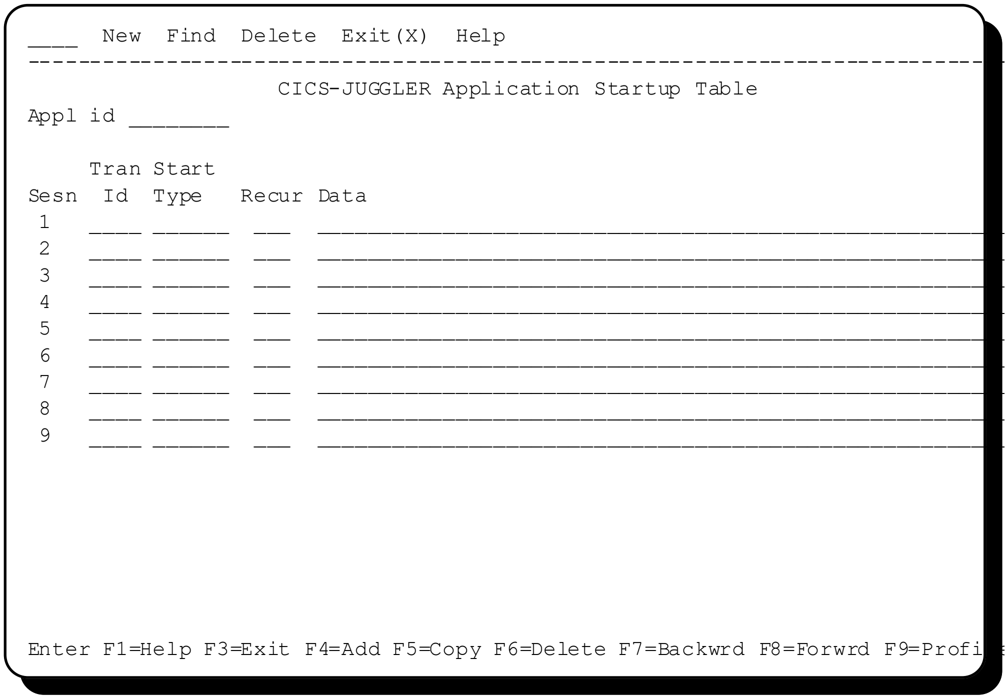
- The (**D**)elete command will delete an entry and shift all lower entries up into the vacated slot.
- The (**I**)nsert command will shift all lower entries down (including the entry in which the command was issued), which will create a vacant slot for inserting a new auto-init entry.

THE APPLICATION STARTUP TABLE

This table provides a means of starting a user transaction automatically in each virtual terminal. You may define the same or different transactions in each session, and you may define data to be passed to the application program to vary the operation in each session.

When used, the transaction specified for virtual terminal number 1 will automatically start when the JUGL,ON command is completed. The transactions corresponding to the remaining sessions will be automatically started the first time the operator toggles into that session.

To display the Application Startup table, from the Auxiliary Functions menu key a "5" and press ENTER. This screen will appear generally as follows:



```
____  New  Find  Delete  Exit(X)  Help
-----------------------------------------------------------------------------
                    CICS-JUGGLER Application Startup Table
Appl id ________

     Tran Start
Sesn  Id  Type   Recur Data
 1   ____ ______  ___  ______________________________________________________
 2   ____ ______  ___  ______________________________________________________
 3   ____ ______  ___  ______________________________________________________
 4   ____ ______  ___  ______________________________________________________
 5   ____ ______  ___  ______________________________________________________
 6   ____ ______  ___  ______________________________________________________
 7   ____ ______  ___  ______________________________________________________
 8   ____ ______  ___  ______________________________________________________
 9   ____ ______  ___  ______________________________________________________








Enter F1=Help F3=Exit F4=Add F5=Copy F6=Delete F7=Backwrd F8=Forwrd F9=Profi
```

You can change any value where the cursor will stop when the TAB key is pressed. The following discussion describes each field.

FIELDS OF THE APPLICATION STARTUP TABLE

APPL ID This is the name of this application startup table. This is the name that is coded in the START TRANSACTION field on a User Profile. In other words, this is the link between a profile and this Start Transaction definition. Values are:

Any valid 1 to 8 character ID.

DATA If START TYPE=ATTACH, the DATA keyword defines the data to be passed to the application program in the terminal I/O area. The data defined is the data which would be entered immediately following the transaction code if this transaction were entered on a terminal screen. Thus, if only the trancode is required as input, the DATA keyword should be omitted.

Example: To automatically start a transaction which would normally be entered on the screen in the form:

FSRV I ACCTFIL A*

the following coding would be required in the statement:

TRANSACTION ID	FSRV
START TYPE	ATTACH
RECUR	YES or NO
DATA	_I_ACCTFIL_A*

[Note]: Automatic initiation of transactions occurs when a toggle is performed.

RECUR This specifies whether the transaction to be started in this session will be initiated <u>every</u> time the session is toggled into or only the first time.

This feature is designed as an aid in integrating session applications. By initiating a transaction every time a session is entered, you can cause things to happen automatically when the operator presses the toggle key.

For instance, the transaction in session number 1 could write a record to temporary storage saving the control information for a given file. A recurring transaction could be specified in session number 2 which, upon invocation, would retrieve the saved control information, fetch the data record and display some related application information.

A special interface to CICS-JUGGLER is provided which allows you to change the transaction to be initiated in the recurring session under program control. This interface is described in the section entitled *USER EXITS AND PROGRAM INTERFACES*.

YES The transaction will be initiated every time the user toggles into this session.

NO The transaction will be initiated only upon the <u>first</u> toggle into this session.

The default is "NO".

SESN This is the session, or virtual terminal number where this transaction is to be initiated.

START TYPE This is the technique for CICS-JUGGLER to use when starting this transaction. Valid entries are:

ATTACH This means that the transaction will be started in the same way that CICS starts a transaction when the transaction code is entered from the terminal. That is, an I/O area is passed to the application containing the transaction code in the first position and any associated data following. ATTACH must be used if data is to be passed to the application, or if the transaction being started runs in a remote region through MRO or ISC.

START START means that the transaction will be initiated using an EXEC CICS START command. When using START, no terminal I/O area is passed to the application program. You can <u>not</u> pass any data to the application program when using the START operand.

[Note]: There is no "preferred" method here. If a transaction does not need incoming data, either method will work. ATTACH will be slightly quicker since it avoids the CICS Automatic Transaction Initiation mechanism.

TRANSACTION ID

This is the transaction code to be automatically started in the corresponding session number.

Any valid 1 to 4 character transaction code that is defined to CICS in the PCT.

ENTERING COMMANDS AT THE APPLICATION STARTUP TABLE

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

ENTER Apply any changes that you have made to the table.

PF1 HELP Display a Help screen for information pertaining to the field to which the cursor pointed when the PF key was pressed.

PF2 FIND Invoke the FIND function to display another Application Startup entry.

PF3 EXIT Return to the Auxiliary Functions menu.

PF4 ADD Add a new entry to the table.

PF5 COPY Copy the displayed entry. This may be used when creating a new table entry if it is advantageous to copy an existing entry, then later modify any differing fields as desired.

PF6 DELETE Delete the displayed entry from the table.

PF7 BACKWARD Browse backward to the previous Application Startup table.

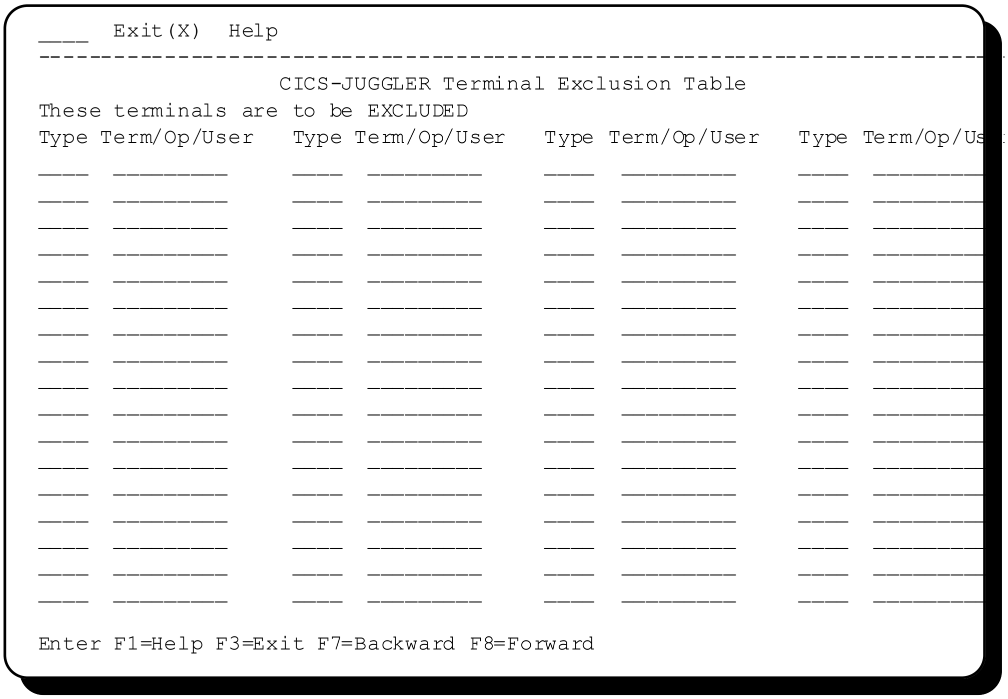
PF8 FORWARD Browse forward to the next Application Startup table.

PF9 PROFILE Display the Profile table. (Same as selection 3 from the Auxiliary Functions menu.)

THE TERMINAL EXCLUSION TABLE

The Terminal Exclusion table is used to define one or more terminal IDs or operators which are to be excluded from using CICS-JUGGLER or included as the only terminals which can use CICS-JUGGLER, according to the specification at the top of the screen.

To display the Terminal Exclusion table, from the Auxiliary Functions menu key a "6" and press ENTER. This screen will appear generally as follows:



```
____  Exit(X)  Help
---------------------------------------------------------------------------
                   CICS-JUGGLER Terminal Exclusion Table
These terminals are to be EXCLUDED
Type Term/Op/User   Type Term/Op/User   Type Term/Op/User   Type Term/Op/Us
____  _________     ____  _________     ____  _________     ____  ________
____  _________     ____  _________     ____  _________     ____  ________
____  _________     ____  _________     ____  _________     ____  ________
____  _________     ____  _________     ____  _________     ____  ________
____  _________     ____  _________     ____  _________     ____  ________
____  _________     ____  _________     ____  _________     ____  ________
____  _________     ____  _________     ____  _________     ____  ________
____  _________     ____  _________     ____  _________     ____  ________
____  _________     ____  _________     ____  _________     ____  ________
____  _________     ____  _________     ____  _________     ____  ________
____  _________     ____  _________     ____  _________     ____  ________
____  _________     ____  _________     ____  _________     ____  ________
____  _________     ____  _________     ____  _________     ____  ________
____  _________     ____  _________     ____  _________     ____  ________
____  _________     ____  _________     ____  _________     ____  ________
____  _________     ____  _________     ____  _________     ____  ________
____  _________     ____  _________     ____  _________     ____  ________

Enter F1=Help F3=Exit F7=Backward F8=Forward
```

You can change any value where the cursor will stop when the TAB key is pressed. The following discussion describes each field.

FIELDS OF THE TERMINAL EXCLUSION TABLE

These terminals are to be

This field determines if this table is to be an inclusion or an exclusion table. Valid entries are:

EXCLUDED
All entries on this table will not be allowed to use CICS-JUGGLER.

INCLUDED Only the entries appearing on this table will be allowed to use CICS-JUGGLER.

TERM/OP/USER
This field is used to supply the actual data that CICS-JUGGLER will use to match with the operator that is attempting to perform a JUGL,ON command.

For TYPE=TERM
Code the terminal ID, as it is known to CICS. This keyword may be coded in any of the following ways:

1) Code an individual statement for each terminal to be defined, coding the TERM keyword as the four-character terminal ID of the terminal to be excluded or included.

2) Define a range of terminals to be configured using the parameters specified on this statement. Code the TERM operand as two terminal IDs separated by a hyphen. With this method, when a JUGL,ON command is performed at any terminal that has a terminal ID equal-to or greater than the first ID and equal-to or less than the second ID, that terminal will be excluded or included.

3) Define a generic definition using "wild-card" characters. Code the TERM operand with question marks (?) in one or more positions of the terminal ID. With this method, when a

JUGL,ON command is issued at any terminal, if all positions in the terminal ID match the corresponding positions in the TERM operand where question marks are not coded, that terminal will be excluded or included.

For TYPE=OP
Code the operator name, as it appears in the TCT. This is the three-character CICS operator ID. The operator ID keyword may be coded in any of the following ways:

1) Code an individual statement for each operator to be defined, coding the three-character operator ID of the user to be excluded or included.

2) Define a range of operators to be configured using the parameters specified on this statement. Code the two operator IDs separated by a hyphen. With this method, when a JUGL,ON command is performed at any terminal that has an operator ID equal-to or greater than the first ID and equal-to or less than the second ID, that terminal will be excluded or included.

3) Define a generic definition using "wild-card" characters. Code question marks (?) in one or more positions of the operator ID. With this method, when a JUGL,ON command is performed at any terminal, if all positions in the operator ID match the corresponding positions where question marks are not coded, that terminal will be excluded or included.

For TYPE=USER
Code the user ID, as it appears in the SNT or RACF profile. This is the eight-character CICS user ID. The user ID keyword may be coded in any of the following ways:

1) Code an individual statement for each user to be defined, coding the eight-character user ID of the user to be excluded or included.

2) Define a generic definition using "wild-card" characters. Code question marks (?) in one or more positions of the user ID. With this method, when a JUGL,ON command is performed at any terminal, if all positions in the user ID match the corresponding positions where question marks are not coded, that terminal will be excluded or included.

TYPE This field is used to specify the type of data CICS-JUGGLER is to use to determine if it should allow an operator to perform a JUGL,ON command. Valid entries are:

TERM This is a terminal table entry.

OP This is an operator table entry.

USER This is a User ID table entry. This option is only valid for CICS releases 1.7 and later.

ENTERING COMMANDS AT THE TERMINAL EXCLUSION TABLE

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

ENTER Apply any changes that you have made to the table.

PF1 HELP Display a Help screen for information on the field that the cursor was in when the PF key was pressed.

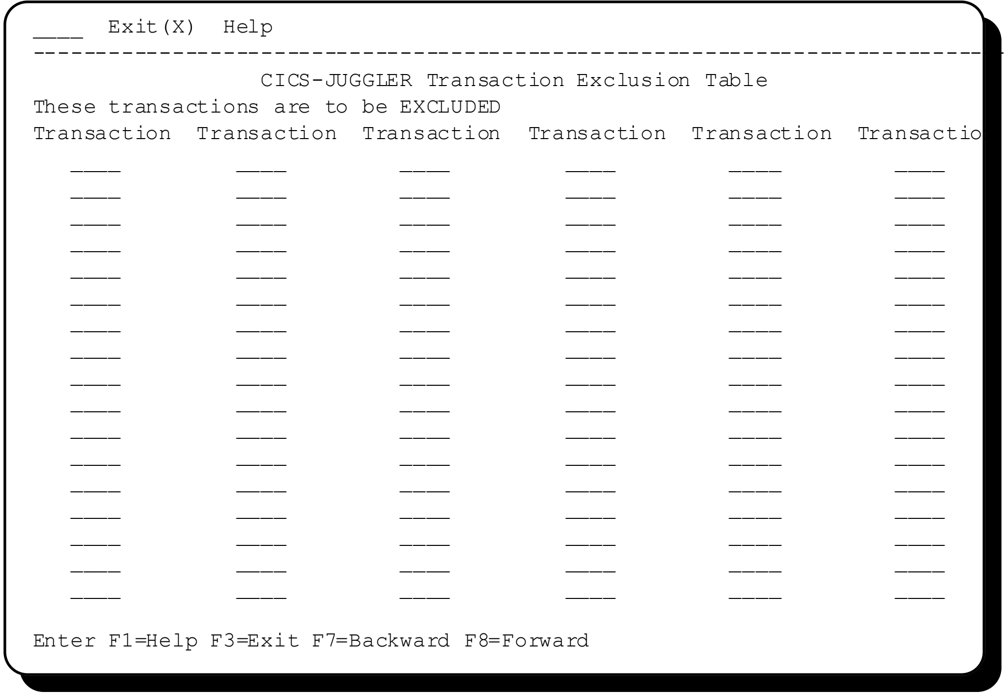
PF3 EXIT Return to the Auxiliary Functions menu.

PF7 BACKWARD Browse backward to the previous page of table entries.

PF8 FORWARD Browse forward to the next page of table entries.

THE TRANSACTION EXCLUSION TABLE

An excluded transaction is one which overrides the "hot" PF keys when active. That is, if a toggle key, window key, switch key, control key or any of the PF key functions that can be assigned is pressed while the transaction is active, that PF key is passed directly to the transaction without performing the assigned CICS-JUGGLER function. Note that transaction exclusion does not preclude the use of toggling through the use of "transparent" commands with the control character. This is intended as a method of disabling hot keys for transactions that use all available keys. This table is used to define one or more transaction codes which are to be excluded from recognizing the toggle keys or included as the only transactions that will recognize the toggle keys, according to the specification in the table. To display the ransaction Exclusion table, from the Auxiliary Functions menu key a "7" and press ENTER. This screen will appear generally as follows:



```
____  Exit(X)  Help
---------------------------------------------------------------------------
                  CICS-JUGGLER Transaction Exclusion Table
These transactions are to be EXCLUDED
Transaction  Transaction  Transaction  Transaction  Transaction  Transactio
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____

Enter F1=Help F3=Exit F7=Backward F8=Forward
```

You can change any value where the cursor will stop when the TAB key is pressed. The following discussion describes each field.

FIELDS OF THE TRANSACTION EXCLUSION TABLE

These TRANSACTIONS are to be

This field determines if this table is to be an inclusion or an exclusion table. Valid entries are:

EXCLUDED All transactions on this table will <u>not</u> be allowed to toggle while using CICS-JUGGLER.

INCLUDED <u>Only</u> the transactions appearing on this table will be allowed to toggle while using CICS-JUGGLER.

TRANSACTION

This is the transaction code as it is known to CICS. Each transaction code may be coded in one of three ways, any combination of which can be used together for different codes:

1). Code the full 4-character transaction code.

2). Code the first 1 to 3 characters followed by an asterisk (*). All transactions with the same 1-3 characters (up to the asterisk) will be matched.

3). Define a generic definition using "wild-card" characters. Code the TRAN operand with question marks (?) in one or more positions of the code. With this method, If all positions in the transaction code match the corresponding positions in the TRAN operand where question marks are <u>not</u> coded, that transaction will be considered a match.

Do not code TRAN=* or TRAN=???? (all wild-cards). This will specify all transactions.

ENTERING COMMANDS AT THE TRANSACTION EXCLUSION TABLE

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

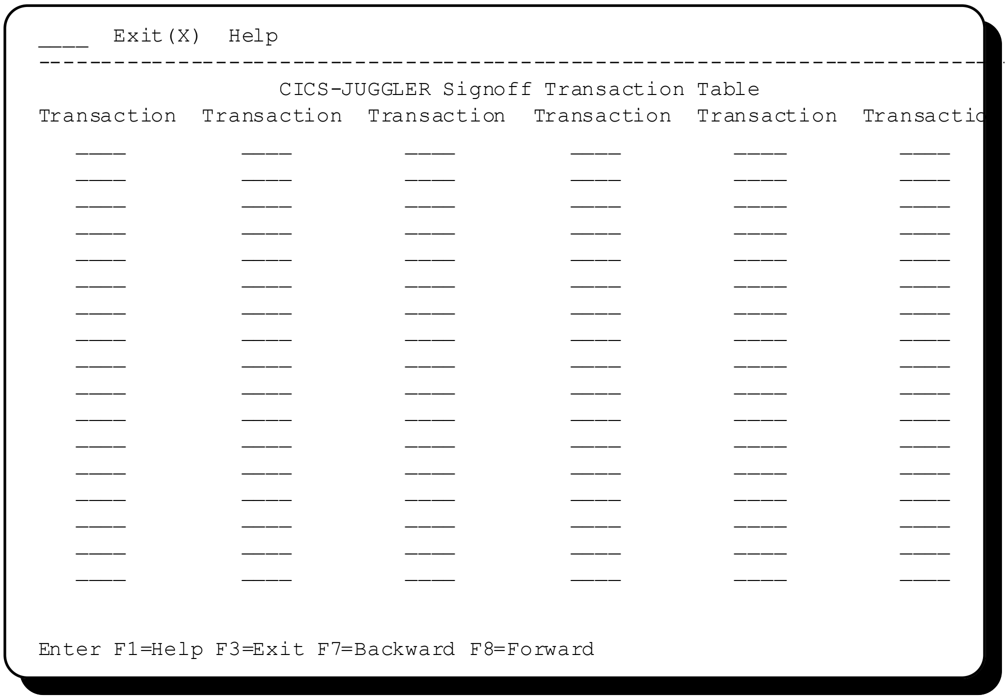
ENTER	Apply any changes that you have made to the table.
PF1 HELP	Display a Help screen for information on the field that the cursor was in when the PF key was pressed.
PF3 EXIT	Return to the Auxiliary Functions menu.
PF7 BACKWARD	Browse backward to the previous page of table entries.
PF8 FORWARD	Browse forward to the next page of table entries.

THE SIGNOFF TRANSACTION TABLE

The Sign-off Transaction table is used to define one or more transaction codes which designate a sign-off transaction.

If you are using something other than CSSF or CESF as a sign-off transaction code, or in DOS/VSE Interactive Interface if exiting the highest level menu is not the same as a sign-off, you will need to code all transaction codes as well.

To display the Sign-off Transaction table, from the Auxiliary Functions menu key a "8" and press ENTER. This screen will appear generally as follows:



```
____  Exit(X)  Help
---------------------------------------------------------------------------
                    CICS-JUGGLER Signoff Transaction Table
Transaction  Transaction  Transaction  Transaction  Transaction  Transactio
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____


Enter F1=Help F3=Exit F7=Backward F8=Forward
```

You can change any value where the cursor will stop when the TAB key is pressed. The following discussion describes each field.

FIELDS OF THE SIGNOFF TRANSACTION TABLE

TRANSACTION

The TRANSACTION operand is the only field of the Sign-off table. It is used to define one or more transaction codes which designate a sign-off transaction to CICS. If the only sign-off transaction code in use is CSSF or CESF or, when using the DOS/VSE Interactive Interface, if exiting the highest-level selection panel via PF3 (or other PF key) constitutes a sign-off, this statement need not be present. You need only specify transactions other than CSSF in this table. This statement is only necessary if you are using the REQUIRE JUGL,OFF AT LOGOFF and FORCE PURGE AT SIGNON AND SIGNOFF. See the discussion of *SESSION TERMINATION OPTIONS* in the section on *SPECIAL CONSIDERATIONS* for more details.

Each transaction code may be coded in one of three ways, any combination of which can be used together for different codes:

1). Code the full 4-character transaction code.

2). Code the first 1 to 3 characters followed by an asterisk (*). All transactions with the same 1-3 characters (up to the asterisk) will be matched.

3). Define a generic definition using "wild-card" characters. Code the operand with question marks (?) in one or more positions of the code. With this method, If all positions in the transaction code match the corresponding positions in the TRANSACTION operand where question marks are <u>not</u> coded, that transaction will be considered a match.

    Do not code "*" or "????" (all wild-cards). This will cause all transactions to be treated as sign-off transactions.

ENTERING COMMANDS AT THE SIGNOFF TRANSACTION TABLE

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

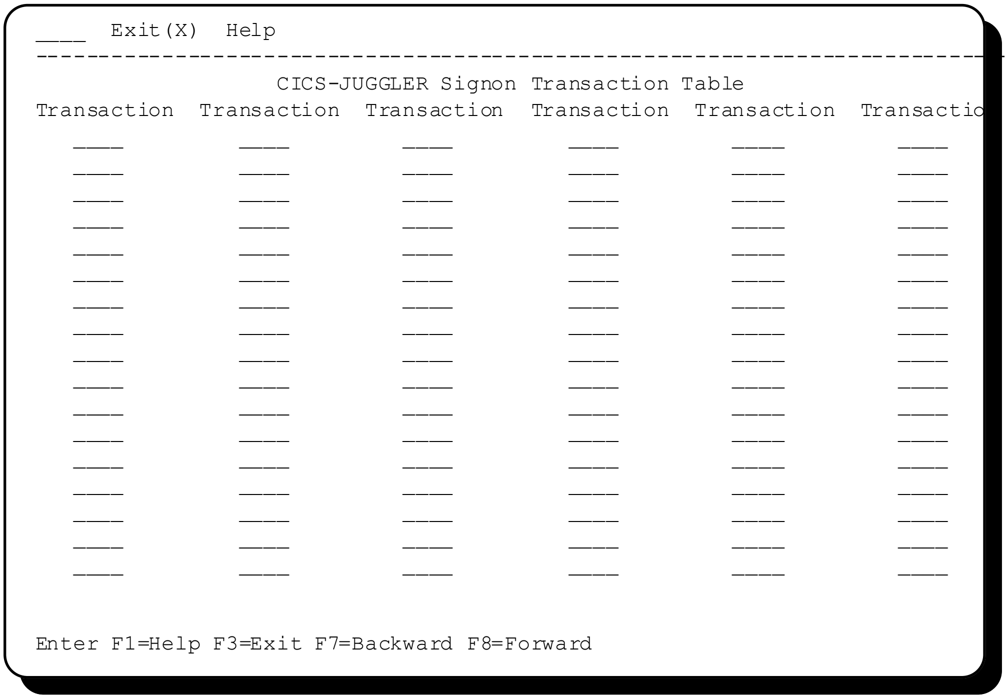
ENTER	Apply any changes that you have made to the table.
PF1 HELP	Display a Help screen for information on the field that the cursor was in when the PF key was pressed.
PF3 EXIT	Return to the Auxiliary Functions menu.
PF7 BACKWARD	Browse backward to the previous page of table entries.
PF8 FORWARD	Browse forward to the next page of table entries.

THE SIGNON TRANSACTION TABLE

The Sign-on Transaction table is used to define one or more transaction codes which designate a sign-on transaction.

If you are using something other than CSSN or CESN as a sign-on transaction code, you will need to code all transaction codes as well.

To display the Sign-on Transaction table, from the Auxiliary Functions menu key a "9" and press ENTER. This screen will appear generally as follows:



```
____  Exit(X)  Help
----------------------------------------------------------------------------
                   CICS-JUGGLER Signon Transaction Table
Transaction  Transaction  Transaction  Transaction  Transaction  Transactio
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____


Enter F1=Help F3=Exit F7=Backward F8=Forward
```

You can change any value where the cursor will stop when the TAB key is pressed. The following discussion describes each field.

FIELDS OF THE SIGNON TRANSACTION TABLE

TRANSACTION

The TRANSACTION operand is the only field of the Sign-on table. It is used to define one or more transaction codes which designate a sign-on transaction to CICS. If the only sign-on transaction code in use is CSSN or CESN this statement need not be present. This statement is only necessary if you are using the FORCE PURGE AT SIGNON AND SIGNOFF option. See the discussion of *SESSION TERMINATION OPTIONS* in section 11 - *SPECIAL CONSIDERATIONS* for more details.

Each transaction code may be coded in one of three ways, any combination of which can be used together for different codes:

1). Code the full 4-character transaction code.

2). Code the first 1 to 3 characters followed by an asterisk (*). All transactions with the same 1-3 characters (up to the asterisk) will be matched.

3). Define a generic definition using "wild-card" characters. Code the operand with question marks (?) in one or more positions of the transaction code. With this method, If all positions in the transaction code match the corresponding positions in the TRANSACTION operand where question marks are <u>not</u> coded, that transaction will be considered a match.

Do not code "*" or "????" (all wild-cards). This will cause all transactions to be treated as sign-off transactions.

ENTERING COMMANDS AT THE SIGNON TRANSACTION TABLE

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

ENTER Apply any changes that you have made to the table.

PF1 HELP Display a Help screen for information on the field that the cursor was in when the PF key was pressed.

PF3 EXIT Return to the Auxiliary Functions menu.

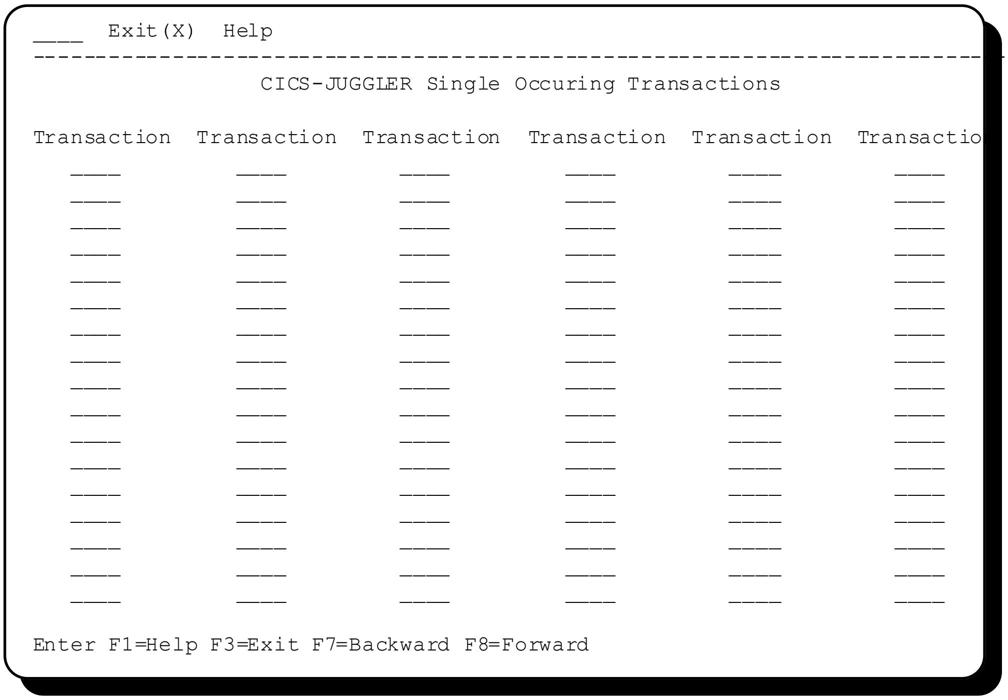
PF7 BACKWARD Browse backward to the previous page of table entries.

PF8 FORWARD Browse forward to the next page of table entries.

THE SINGLE OCCURRING TRANSACTIONS TABLE

This table is used to define one or more transaction codes which are "single-occurring", meaning it can only be active in one virtual terminal per physical terminal at a time. This option should be used if you have certain packages or application systems which can not run in two sessions at once on the same physical terminal. There are very few transactions where this is a problem, but ADR's VOLLIE system is one example of such a package. Running VOLLIE in multiple sessions on the same terminal can result in transaction ABENDS in the VOLLIE program. You should also consider using this feature if you are not using Pseudo terminal IDs to establish a unique terminal ID in each virtual terminal.

To display the Single Occurring Transactions table, from the Auxiliary Functions menu key a "12" and press ENTER. This screen will appear generally as follows:



```
____  Exit(X)  Help
----------------------------------------------------------------------------
                  CICS-JUGGLER Single Occuring Transactions

Transaction  Transaction  Transaction  Transaction  Transaction  Transactio
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____

Enter F1=Help F3=Exit F7=Backward F8=Forward
```

You can change any value where the cursor will stop when the TAB key is pressed. The following discussion describes each field.

FIELDS OF THE SINGLE OCCURRING TRANSACTIONS TABLE

TRANSACTION

This is the only operand of the Single Occurring Transactions table. It is used to define one or more transaction codes which are "single-occurring", meaning it can only be active in one virtual terminal per physical terminal at a time.

Each transaction code may be coded in one of three ways, any combination of which can be used together for different codes:

1). Code the full 4-character transaction code.

2). Code the first 1 to 3 characters followed by an asterisk (*). All transactions with the same 1-3 characters (up to the asterisk) will be matched.

3). Define a generic definition using "wild-card" characters. Code the TRAN operand with question marks (?) in one or more positions of the code. With this method, If all positions in the transaction code match the corresponding positions in the TRAN operand where question marks are <u>not</u> coded, that transaction will be considered a match.

Do not code "*" or "????" (all wild-cards). This will allow a single transaction to operate in only one session.

ENTERING COMMANDS AT THE SINGLE OCCURING TRANSACTION TABLE

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

ENTER Apply any changes that you have made to the table.

PF1 HELP Display a Help screen for information on the field that the cursor was in when the PF key was pressed.

PF3 EXIT Return to the Auxiliary Functions menu.

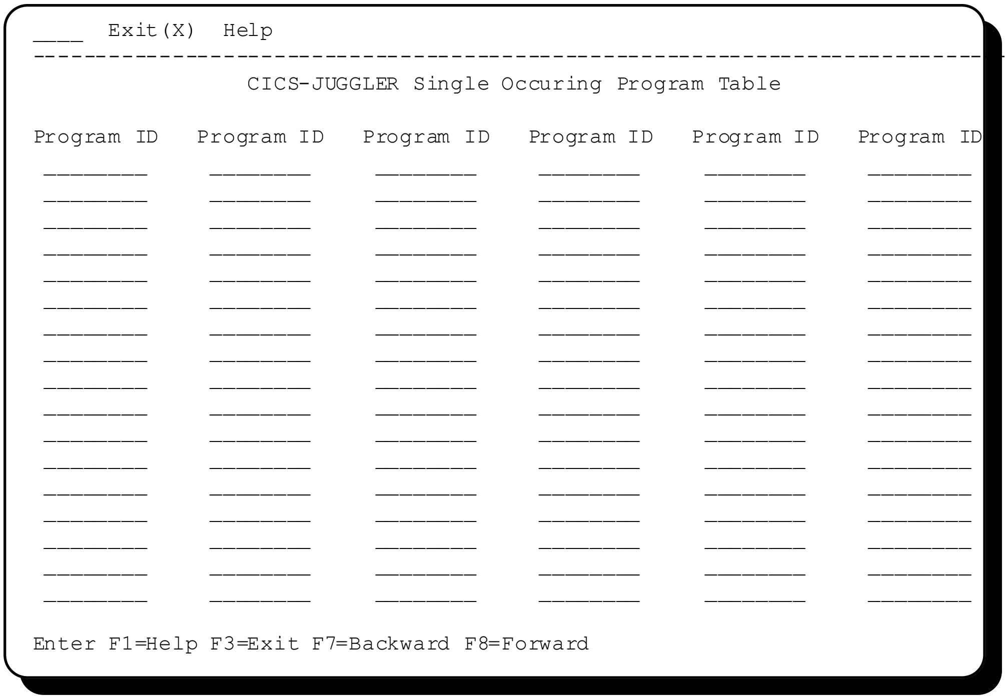
PF7 BACKWARD Browse backward to the previous page of table entries.

PF8 FORWARD Browse forward to the next page of table entries.

THE SINGLE OCCURRING PROGRAMS TABLE

This table is used to define one or more program names which are "single-occurring", meaning it can only be active in one virtual terminal per physical terminal at a time. This option should be used if you have certain packages or application systems which can not run in two sessions at once on the same physical terminal. You should also consider using this feature if you are not using Pseudo terminal IDs to establish a unique terminal ID in each virtual terminal.

To display the Single Occurring Programs table, from the Auxiliary Functions menu key a "13" and press ENTER. This screen will appear generally as follows:



```
____  Exit(X)  Help
------------------------------------------------------------------------------
                    CICS-JUGGLER Single Occuring Program Table

Program ID   Program ID   Program ID   Program ID   Program ID   Program ID
 ________     ________     ________     ________     ________     ________
 ________     ________     ________     ________     ________     ________
 ________     ________     ________     ________     ________     ________
 ________     ________     ________     ________     ________     ________
 ________     ________     ________     ________     ________     ________
 ________     ________     ________     ________     ________     ________
 ________     ________     ________     ________     ________     ________
 ________     ________     ________     ________     ________     ________
 ________     ________     ________     ________     ________     ________
 ________     ________     ________     ________     ________     ________
 ________     ________     ________     ________     ________     ________
 ________     ________     ________     ________     ________     ________
 ________     ________     ________     ________     ________     ________
 ________     ________     ________     ________     ________     ________
 ________     ________     ________     ________     ________     ________
 ________     ________     ________     ________     ________     ________
 ________     ________     ________     ________     ________     ________

Enter F1=Help F3=Exit F7=Backward F8=Forward
```

You can change any value where the cursor will stop when the TAB key is pressed. The following discussion describes each field.

FIELDS OF THE SINGLE OCCURRING PROGRAMS TABLE

PROGRAM ID This is the only field of the Single Occurring Programs table. It is used to define one or more program names which are "single-occurring", meaning it can only be active in one virtual terminal at a time.

Each program may be coded in one of the following ways, any combination combination of which can be used for different codes:

1). Code the full 8-character program name.

2). Code the first 1 to 7 characters followed by an asterisk (*). All programs with the same 1-7 characters (up to the asterisk), will be matched.

3). Define a generic definition using "wild-card" characters. Code the operand with question marks (?) in one or more positions of the name. With this method, If all positions in the program name match the corresponding positions in the PROGRAM ID operand where question marks are <u>not</u> coded, that program will be considered a match.

Do not code "*" or "????????" (all wild-cards). This will allow a single program to operate in only one session.

ENTERING COMMANDS AT THE SINGLE OCCURRING PROGRAMS TABLE

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

ENTER Apply any changes that you have made to the table.

PF1 HELP Display a Help screen for information on the field that the cursor was in when the PF key was pressed.

PF3 EXIT Return to the Auxiliary Functions menu.

PF7 BACKWARD Browse backward to the previous page of table entries.

PF8 FORWARD Browse forward to the next page of table entries.

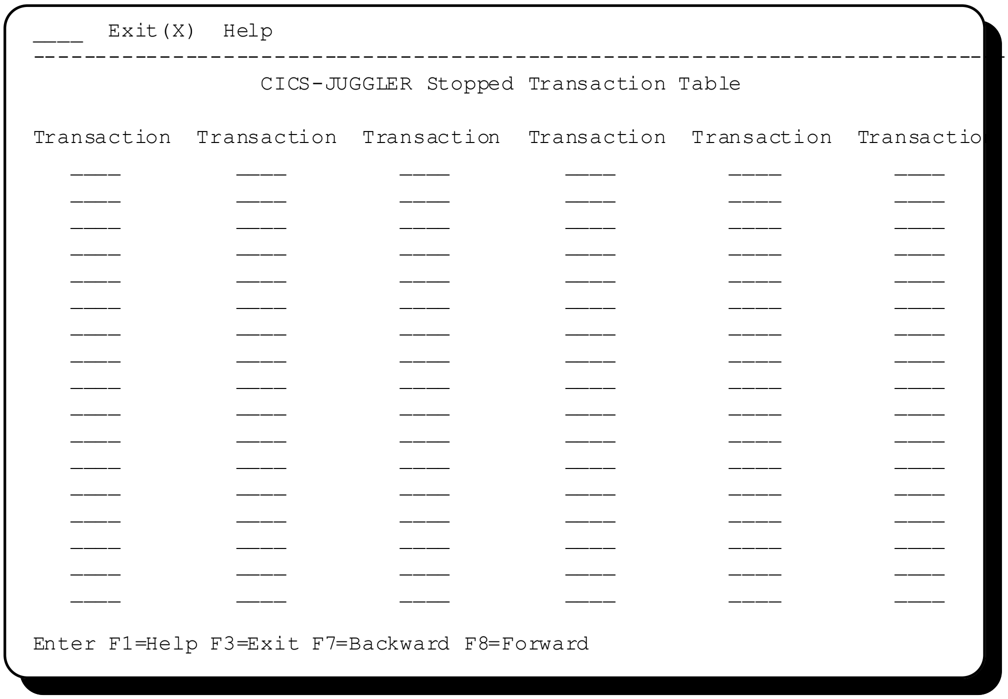
THE STOPPED TRANSACTIONS TABLE

This table is used to define one or more transaction codes which will not be allowed to execute if CICS-JUGGLER is active on the terminal. If one of the transaction codes in this table is entered the following message will display:

JUGL0022. xxxx CAN'T BE STARTED WITH CICS-JUGGLER

There are no known transactions that cannot run with CICS-JUGGLER. This table is provided as a "stop-gap" measure to prevent problems if one is encountered until Unicom technicians can resolve the conflict.

To display the Stopped Transactions table, from the Auxiliary Functions menu key a "14" and press ENTER. This screen will appear generally as follows:



```
____  Exit(X)  Help
--------------------------------------------------------------------------
                  CICS-JUGGLER Stopped Transaction Table

Transaction  Transaction  Transaction  Transaction  Transaction  Transactio
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____
   ____         ____         ____         ____         ____         ____

Enter F1=Help F3=Exit F7=Backward F8=Forward
```

You can change any value where the cursor will stop when the TAB key is pressed. The following discussion describes each field.

FIELDS OF THE STOPPED TRANSACTIONS TABLE

TRANSACTION

This is the only field of the Stopped Transactions table. It is used to define one or more transaction codes that will not be allowed to execute if CICS-JUGGLER is active on the terminal.

Each transaction code may be coded in the following ways, any combination of which can be used together for different codes:

1). Code the full 4-character transaction code.

2). Code the first 1 to 3 characters followed by an asterisk (*). All transactions with the same 1-3 characters (up to the asterisk) will be matched.

3). Define a generic definition using "wild-card" characters. Code the operand with question marks (?) in one or more positions of the code. With this method, If all positions in the transaction code match the corresponding positions in the TRANSACTION operand where question marks are <u>not</u> coded, that transaction will be considered a match.

Do not code "*" or "????" (all wild-cards). This will cause all transactions to be treated as non-executable transactions.

ENTERING COMMANDS AT THE STOPPED TRANSACTIONS TABLE

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

ENTER Apply any changes that you have made to the table.

PF1 HELP Display a Help screen for information on the field that the cursor was in when the PF key was pressed.

PF3 EXIT Return to the Auxiliary Functions menu.

PF7 BACKWARD Browse backward to the previous page of table entries.

PF8 FORWARD Browse forward to the next page of table entries.

SPECIAL CONSIDERATIONS

PERFORMANCE TIPS

Generally speaking, CICS-JUGGLER adds so little overhead to the CICS system that performance is not much of a consideration. However, in certain situations, CPU overhead and/or storage overhead can impact overall system performance. In these cases there are certain customization options that can be avoided in order to effect less overhead, and others that can be activated in order to better control the environment.

Some situations where performance needs to be considered are as follows:

1). **Conversational Tasks**. If there are many conversational transactions in use, or if there are a few conversational tasks that are very high in storage utilization, you may want to place certain limits on the use of these transactions with CICS-JUGGLER. When a conversational task is toggled out-of, the dynamic storage in use by that task cannot be released. Thus, adding multiple session capability means that each operator can start more of these tasks, all of which will tie up resources.

There are three ways to control this situation, any or all of which may be used:

a). Use the Single-Occurring Program and/or Single-Occurring Transaction tables to limit each physical terminal to only one occurrence at a time of the transactions which are conversational.

b). Use the Transaction Exclusion table to prevent an operator from toggling out of specified tasks. This will force the operator to terminate the task before any other session can be entered.

c). Use the TIME OUT SELECTION TYPE option to cause CICS-JUGGLER to automatically purge any conversational task if the session containing that task is not entered for a specified interval of time.

2). **Extremely High Usage.** If you plan to allow 1500 to 2000 or more terminal operators to activate CICS-JUGGLER, CPU overhead may become a consideration. CICS-JUGGLER keeps an in-memory table called the "active user table" where an entry is kept for each active terminal. This table must be scanned at various points during operation in order to recognize an active terminal and test for certain activity to be handled by the JUGGLER program.

You can reduce the amount of scan time and other overhead factors with the following options:

a). Use READ BUFFER=YES. The READ BUFFER=NO option causes CICS-JUGGLER to get involved with every terminal output that occurs on every terminal. A scan must be performed to determine if this is a CICS-JUGGLER terminal and, if so, the terminal output is saved in order to reconstruct the terminal buffer when the toggle key is pressed.

With READ BUFFER=YES, terminal output is not intercepted unless the terminal is in window mode and no datastream saving is performed. While it is true that a read-buffer command takes longer to complete on a remote terminal, the overall savings in system performance may be worth it.

b). Use MODIFY TEMP STORAGE KEYS=NO in the User Options table and specify Pseudo Terminal IDs. MODIFY TEMP STORAGE KEYS should never be selected if pseudo IDs are in use. It has no effect when pseudo IDs are in use, but it causes a global CICS exit to be invoked on all temporary storage activity, and a scan of the active user table in order to determine that the terminal is using pseudo IDs.

c). Use MAIN storage rather than temporary. No I/O is required at all. The MAIN option will store all terminal images above the 16M line.

3). **Avoid Use of the Multiple Interactive Interface Feature.** The multiple interactive interface feature (VSE INTERACTIVE INTERFACE SUPPORT=YES) in DOS/VSE systems adds significant storage overhead to the system. It should not be used unless it is limited to a very few operators.

INITIALIZING CICS-JUGGLER IN THE PLT

CICS-JUGGLER initializes during the first invocation of the JUGGLER phase. Usually this occurs the first time the **JUGL** transaction code is issued; however there are some benefits to initializing CICS-JUGGLER at PLT time:

- If AUTOSTART=YES is coded in any User Profile table, no terminals, operators or users can be auto-started until CICS-JUGGLER has been initialized.
- If your environment has multiple products that must be initialized in a certain order, you can ensure that the products are initialized correctly.

However you should read the following considerations before activating CICS-JUGGLER in the PLT:

- If you are installing CICS-JUGGLER for the first time or if this is a subsequent installation of a more current release level, you should not activate CICS-JUGGLER in the PLT until you are certain that the product has been installed correctly, as some error messages may not display on the system console.

- If this is a subsequent installation of a more current release level, you may encounter some difficulty migrating the customizational options of your previous version of CICS-JUGGLER to the newer version.

If after reading the above, you do wish to initialize CICS-JUGGLER in the PLT, you will need to assemble and link-edit a system initialization Program List Table (or add JUGGLER to an existing table) as follows:

```
DFHPLT  TYPE=INITIAL,SUFFIX=xx
DFHPLT  TYPE=ENTRY,PROGRAM=JUGGLER
DFHPLT  TYPE=FINAL
```

NEWCOPY COMMANDS FOR JUGGLER PROGRAMS

Once activated, do not attempt to do a CSMT or CEMT NEW COPY command for the JUGGLER or JUGLTBL programs. Both programs are permanently "in-use" and the NEW COPY function may disable the program. This will result in APCT ABENDs for all transactions on that terminal, until the the program is enabled.

In order to load a new version of either the JUGGLER or JUGLTBL modules you must use the JUGL,BACKOUT command, as described under *SPECIAL FUNCTION COMMANDS*, then perform the NEW COPY function.

PSEUDO TERMINAL ID CONSIDERATIONS

The use of Pseudo Terminal IDs is not required, but should be used in most CICS configurations. The Pseudo ID is used by CICS-JUGGLER to alter the terminal ID in the TCTTE (TCTTETI) each time the "toggle" key is pressed. The Pseudo ID will stay in the TCTTE until another virtual terminal is used or CICS-JUGGLER is ended on that terminal.

For terminal dependent applications (applications that base their logic on the terminal ID in any way) Pseudo IDs are necessary in order to run the same transaction in multiple virtual terminals on the same physical terminal.

The one exception to this is applications that use the terminal ID as part of a Temporary Storage key. This is a common practice and the use of pseudo IDs will resolve any conflicts. However, you can often operate without pseudo IDs by selecting the MODIFY TEMP STORAGE KEYS parameter of the User Options table, Temporary Storage keys containing terminal IDs are modified with the session number prior to writing or reading a Temporary Storage record. This technique prevents duplication of Temporary Storage keys, even though the terminal ID is the same in each session.

In general, you should use Pseudo IDs in any of the following situations:

- If applications are terminal dependent in some way other than by use of terminal ID with Temporary Storage.

- In an MRO/ISC environment where separation of terminal IDs in the remote region is required.

- Multiple Interactive Interface support for DOS/ VSE users.

- When time-initiated continual refresh transactions are in use.

If you have one of the situations listed above but do not want to use Pseudo IDs, you can avoid problems by defining a single-occurring program or single-occurring transaction table, as described in section 10 - *CUSTOMIZATION*. These tables prevent the simultaneous execution in multiple sessions of the same terminal of competing transactions.

When using Pseudo IDs, CICS terminal statistics are all accumulated for the real terminal ID only.

ACF2 users (prior to release 5.0) should not use Pseudo IDs. These versions of ACF2 require the operator to sign on in each virtual terminal if Pseudo IDs are in use. For release 5.0 and above, you can code the 'terminal identification' user exit in ACF2, using one of the interface routines for

CICS-JUGGLER which obtains the real terminal ID and returns it to ACF2. For more information, please refer to *OBTAINING THE REAL TERMINAL ID FROM A PSEUDO ID* in section 14 - *USER EXITS AND PROGRAM INTERFACES*.

You might want to suppress or deactivate Pseudo IDs for certain transactions, usually editors, in order to get two versions of the transaction going on one physical terminal. Pansophic's O-W-L is a good example of this. If Pseudo IDs are set, O-W-L will not let you sign on to more than one virtual terminal. However, without Pseudo IDs, you can have multiple sessions of O-W-L running at a time.

AUTO-INITIATED TRANSACTIONS

If auto-initiated transactions which do continual timed displays are in use with CICS-JUGGLER, Pseudo Terminal IDs must be used. An auto-initiated task will only display in the virtual terminal where it was started if Pseudo Terminals are in effect. Without using Pseudo Terminals, when an auto-initiated task is started in one virtual terminal it will over-write the display in the next virtual terminal after pressing the "toggle" key.

ICCF REQUIREMENTS AND PECULIARITIES

DOS/VSE users with IBM's ICCF editor must be aware of the following conditions when using ICCF with CICS-JUGGLER.

- If you press any Toggle key after doing a SUBMIT LIBC, or $DA (any time initiated transaction in ICCF) before the completion message is displayed, you must use Pseudo Terminal IDs or the completion message will over-write the display in the next session. This will also cause an ICCF error.

- You must use Pseudo Terminal IDs if you are using the Forced Log-Off option in ICCF.

- You may have more than one session with ICCF active. DOS VSE (non SP or ESA) users may have the same or different ICCF Log-Ons in each session. VSE/SP and VSE/ESA users must have a different ICCF Log-on in each session.

- When running multiple ICCF sessions, you can not edit the same member in each session. (ICCF will prevent this).

- When running multiple ICCF sessions, you can not copy or move text from one session to the other. Each session has its own unique Scratch-Pad area.

- Because of the peculiarities of ICCF auto-initiated transactions, it is recommended that when running multiple ICCF sessions, you separate each ICCF session with a non-ICCF session. Thus, you could have four virtual terminals, ICCF in 1 and 3 and something else in 2 and 4.

  The only time you will encounter a problem by not doing this is in the case mentioned above, where you toggle out of a SUBMIT (or other time initiated task) before the completion message is displayed. When you do this with two ICCF sessions next to each other, the completion message will sometimes come back in the wrong session.

  [Note]: The above-mentioned peculiarity does not apply to VSE/SP and VSE/ESA users.

- You must use Pseudo Terminal IDs to run multiple ICCF sessions.

USING THE CURSOR-SELECT KEY AS A HOT KEY

There is an additional key on most 3270 keyboards which can be used for any of the CICS-JUGGLER function keys (toggle, back-toggle, or direct-session). It is the "cursor-select" key, usually labelled "cur sel" or "curs slct". It is not present on the older 3277 terminals, but is present on the 3278 and most later models.

To specify this key in the customization entries of the User Option Table, code CSEL as the key name for TOGGLE FORWARD, TOGGLE BACKWARD, SESSIONS DIRECT keys. To specify this key in response to the initial prompts for the toggle key, simply press the cursor-select key, as you would any PF or PA key.

Use of the cursor-select key provides full functionality in most cases, but may take some special handling in some instances. The key is seldom used in application systems, which makes it a nice key to use as a "hot" key, since there is no conflict. However, you need to understand the hardware constraints of the key, and the special handling that CICS-JUGGLER does to allow its use in order to make a decision as to its usage in your environment. You may find that it is more trouble than it's worth.

The cursor-select key is provided as an alternative to the light-pen, for use in menu-selection type applications. IBM's intention was that you could program pen-detectable fields on the screen, position the cursor to the desired field and press the cursor-select key to make your selection. It simulates the light-pen in that it passes a special "aid" byte, along with the cursor position to the application program.

However, if the cursor is not positioned in a pen-detectable field where the first position of that field is either a blank (x'40') or a null (x'00'), the key will not cause an interrupt and will not function at all. You get a keyboard-inhibit indicator and must press RESET to free the keyboard and continue.

In order to allow the cursor-select key to be used, CICS-JUGGLER does the following, if the cursor-select key is used as any designated function key:

- Every displayable field of every output display on this terminal will have the "pen-detectable" attribute bit set on.

- If no fields are present in the output display (no attributes are found), a pen-detectable attribute is added to the end of the datastream.

- If the cursor is not explicitly positioned in the output display, the cursor will be positioned to the pen-detectable attribute at the end of the datastream.

- When the CLEAR key is pressed, a datastream consisting of one unprotected, pen-detectable field is written back to the screen.

[Note]: These modifications are done only for the terminal where the cursor-select key is designated as a CICS-JUGGLER function key and not for any terminal where it is not in use.

By performing these modifications to the output datastreams, you will find that in many cases, the cursor will be positioned to a blank, pen-detectable field on the screen. Pressing the cursor-select key at that point will cause an interrupt and the designated function will be performed.

If the cursor is not positioned to a blank field, if the first position of the field wherein it is positioned is blank, it will still work.

If the first position of the field where the cursor is positioned is non-blank, you can often move the cursor to a blank spot on the screen and press the cursor-select key and it will work. The same rules apply, that is, the first position of that field must be blank. It does not matter whether the field is protected or unprotected.

If none of the above seems to work, you can tab to an unprotected field on the screen and erase the first position of that field (or erase the whole field). You will now be able to use the cursor-select key. However, when you toggle back into this session, the data that you erased is still erased. CICS-JUGGLER has no way of restoring the field.

Also, you can always clear the screen, which causes the entire screen to be set to unprotected, pen-detectable, thereby allowing the cursor-select key to work.

As previously stated, you may find this more trouble than it is worth. However, depending on the application screens in your environment, you may find that it works quite well, thereby giving you the availability of a "hot" key which does not conflict with your user applications.

[Note]: You should not attempt to use the cursor-select key if you have applications that depend on pen-detectable fields, using either the light-pen or the cursor-select key for application logic.

SESSION TERMINATION OPTIONS

Depending on your security requirements, the choice of the session termination options in the User Option Table (options REQUIRE JUGL,OFF AT LOGOFF, REQUIRE CLEAR SESSIONS BEFORE OFF, REQUIRE TRANSACTION END BEFORE OFF, and FORCE PURGE AT SIGNON AND SIGNOFF should be carefully considered). Each option offers a slightly different variation to the action that CICS-JUGGLER performs at sign-off, sign-on or JUGGLER termination time.

You must consider four situations:

1). What happens when a JUGL,OFF command is issued?

2). What happens when a CICS sign-off transaction is issued?

3). What happens when another CICS sign-on transaction is issued after previously signing on and activating CICS-JUGGLER?

4). What happens, in VTAM systems, when, for any reason, the terminal is dropped from CICS and returns to VTAM?

Referring to these four questions, the following discussion describes the action taken by CICS-JUGGLER for each option, as well as the absence of all four options:

[Note]: There is another alternative to handling session termination which involves using one of the program interface routines. This involves calling CICS-JUGGLER with a PURGE command when an operator signs off and also when an operator signs on. Two sample programs, JUGLCSSN and JUGLCSSF are present in the first file on the installation tape which do these functions. See *SAMPLE PROGRAMS ON THE INSTALLATION TAPE* in section 14 - *USER EXITS AND PROGRAM INTERFACES* for more information.

If no options are coded (default action).

Question 1. What happens at JUGL,OFF?

If there are no conversational tasks active in any session, CICS-JUGGLER is terminated. Otherwise, the following message will display:

JUGL1409. ACTIVE TASK xxxx ON LOGICAL TERMINAL n MUST BE ENDED.

The conversational transaction must be ended, then the JUGL,OFF command must be re-issued.

Question 2. What happens at CICS sign-off?

The sign-off is allowed to complete, then CICS-JUGGLER starts a task at the terminal to issue the message:

JUGL0004. DO YOU WANT TO TERMINATE CICS-JUGGLER ?

The operator replies with 'Y' or 'N'. Replying 'Y' causes normal JUGL,OFF logic to be taken as described in question 1, above. Replying 'N' allows the operator to leave this CICS system and go to another system, then return to this CICS, press the toggle key and have the session restored.

[Note]: Since this message appears after sign-off is complete, you must not code security on the JUGL transaction for this to work correctly.

Question 3. What happens with a subsequent CICS sign-on?

No action is taken, the operator can sign-on again without interruption.

Question 4. What happens if the terminal is dropped?

No action is taken. The operator may sign on again or simply re-enter CICS after re-acquiring the terminal. Note, however, that this can sometimes be a dangerous situation, depending on the security system in use. If CICS-JUGGLER is not terminated, previously saved addresses that are no longer in use can be restored, resulting in storage violations.

If the Require Transaction End Before Off option is coded.

Question 1. What happens at JUGL,OFF?

If there are no conversational or Pseudo-conversational tasks active in any session, CICS-JUGGLER is terminated. Otherwise, the message

JUGL1409. ACTIVE TASK xxxx ON LOGICAL TERMINAL n MUST BE ENDED.

will display. The transaction must be ended, then the JUGL,OFF command re-issued.

CICS-JUGGLER recognizes a pseudo-conversational task by the presence of a return transaction code (next-tran) in the session.

Question 2. What happens at CICS sign-off?

The sign-off is allowed to complete, then CICS-JUGGLER starts a task at the terminal to issue the message:

JUGL0004. DO YOU WANT TO TERMINATE CICS-JUGGLER ?

The operator replies with 'Y' or 'N'. Replying 'Y' causes normal JUGL,OFF logic to be taken as described in question 1, above. Replying 'N' allows the operator to leave this CICS system and go to another system, then return to this CICS, press the toggle key and have the session restored.

[Note]: Since message 0004 appears after sign-off is complete, you must not code security on the JUGL transaction for this to work correctly.

Question 3. What happens with a subsequent CICS sign-on?

No action is taken, the operator can sign-on differently or the same in each session.

Question 4. What happens if the terminal is dropped?

No action is taken. The operator may sign on again or simply re-enter CICS after re-acquiring the terminal. Note, however, that this can sometimes be a dangerous situation, depending on the security system in use. If CICS-JUGGLER is not terminated, previously saved addresses that are no longer in use can be restored, resulting in storage violations.

If the Require Clear Sessions Before Off option is coded.

This option works exactly like REQUIRE TRANSACTION END BEFORE OFF, in that it prevents CICS-JUGGLER termination if there are any active sessions. However, it is more stringent that REQUIRE TRANSACTION END BEFORE OFF in that it requires that every session screen except the one where the JUGL,OFF command is being issued be absolutely clear.

Usually this is unnecessary, and makes it quite difficult to terminate CICS-JUGGLER. Therefore it is not recommended. It is provided for those users who have non-conversational tasks, not using RETURN TRANSID, that must be ended before terminating JUGGLER.

If the Require JUGL,OFF At Logoff option is coded.

Question 1. What happens at JUGL,OFF?

Normal JUGL,OFF logic is used, as described above. If either REQUIRE TRANSACTION END BEFORE OFF or REQUIRE CLEAR SESSIONS BEFORE OFF are coded, their logic will be invoked.

Question 2. What happens at CICS sign-off?

If CICS-JUGGLER is active on the terminal, the sign-off is not allowed to complete. The following message is displayed:

JUGL0009. CICS-JUGGLER MUST BE TERMINATED TO LOG-OFF

The operator must issue a JUGL,OFF command to terminate CICS-JUGGLER, which will take normal logic as described above, then re-issue the sign-off transaction.

[Note]: Since message 0009 appears before sign-off is complete, you can code security on the JUGL transaction with this option.

Question 3. What happens with a subsequent CICS sign-on?

When CSSN, or your sign-on transaction is entered and CICS-JUGGLER is active on the terminal, it is assumed that it is a second sign-on (REQUIRE JUGL,OFF AT LOGOFF implies that you sign-on before performing a JUGL,ON command). With REQUIRE JUGL,OFF AT LOGOFF, the subsequent sign-on is treated like a sign-off and message 0009 is displayed as described above. Transactions other than CSSN which perform a sign-on should be defined in the Sign-on Transaction table.

Question 4. What happens if the terminal is dropped?

No action is taken. The operator may sign on again or simply re-enter CICS after re-acquiring the terminal. If the operator signs on, it will be treated as described in question 3, above. Note, however, that this can sometimes be a dangerous situation, depending on the security system in use. If CICS-JUGGLER is not terminated, previously saved addresses that are no longer in use can be restored, resulting in storage violations.

If the Force Purge at Signon and Signoff option is coded.

Question 1. What happens at JUGL,OFF?

Normal JUGL,OFF logic is used, as described above. If REQUIRE JUGL,OFF AT LOGOFF, REQUIRE TRANSACTION END BEFORE OFF or REQUIRE CLEAR SESSIONS BEFORE OFF are coded, their logic will be invoked.

Question 2. What happens at CICS sign-off?

If CICS-JUGGLER is active on the terminal, the terminal is purged from CICS-JUGGLER, then the sign-off is allowed to complete.

Question 3. What happens with a subsequent CICS sign-on?

CICS-JUGGLER is automatically terminated with a PURGE command before a new sign-on is accepted. This means that any conversational transactions that are active on the terminal are abnormally ended with an AKCS abend.

This option effectively prevents multiple sign-ons at a terminal, plus insuring that CICS-JUGGLER is off when a user tries to sign-on again.

[Note]: If your sign-on transaction is something other than 'CSSN' or 'CESN', it should be coded in the Sign-on/Sign-off Transaction table.

Question 4. What happens if the terminal is dropped?

No action is taken at the time the terminal is dropped, since CICS-JUGGLER usually can not recognize when this occurs. However, assuming that the operator must sign-on in order to re-acquire the terminal, it will be treated as described in question 3, above. This option effectively resolves the danger of storage violations described previously.

[Notes]:

1). If you are using a security package, you should always use FORCE PURGE AT SIGNON AND SIGNOFF. Most security packages release their control blocks at sign-off. Therefore leaving CICS-JUGGLER active after signing off can cause a bad control block address to be restored, causing a possible system abend or storage violations.

Security packages RACF, ACF2, TOP SECRET and SURVEILLANCE are known to require these options.

2). Even if you don't have a security package, omitting REQUIRE JUGL,OFF AT LOGOFF or PURGE AT SIGNON AND SIGNOFF involves a security risk, since it allows a new operator to toggle into a session transaction which they might not be authorized to use.

3). In general, REQUIRE TRANSACTION END BEFORE OFF and REQUIRE CLEAR SESSIONS BEFORE OFF need never be used. It causes no problems to terminate a session when a pseudo-conversational transaction is active in that session, since the transaction is at a logical sync point anyway. It's the same situation as if CICS came down while a pseudo-conversational transaction was awaiting input.

However, if database integrity could be disturbed by ending a pseudo-conversational transaction without going through normal completion, you should use one of these options.

UNIQUE ENVIRONMENTS AND SPECIAL SITUATIONS

The following discussion concerns the special considerations and facilities provided for handling different CICS environments and situations. In particular, the MRO/ISC environment and the VSE Interactive Interface are discussed.

RUNNING CICS-JUGGLER WITH MRO/ISC

If the Multiple Region Operation (MRO) or Inter-System Communication (ISC) feature of CICS is in use, the following considerations will apply:

- All CICS-JUGGLER programs must be loaded in the Terminal Owning Region, not in the remote regions. If there are multiple Terminal Owning Regions, they may be loaded in each one.
- If Pseudo Terminals ID's are not specified (all virtual terminals have the same terminal ID), CICS-JUGGLER will work and you can effectively toggle across regions through MRO, using either the CRTE transaction or remote transactions in the PCT. However, the following conditions can occur:

1). When a remote transaction is started from a JUGGLER virtual terminal that is conversational (issues a terminal read), the remote TCT entry is considered active by CICS until that transaction is ended.

If you then press the "toggle" key and move to another session, then start another transaction routed to the same remote region, you will receive CICS Abend ATZW, which says that an attempt was made to attach a task to a remote terminal that was already running a task. The Abend will not effect anything going on in the other sessions and if you toggle back to the conversational transaction and end it, you may then start another remote task to that system.

2). When a remote pseudo-conversational program is started which uses a COMAREA, then the operator toggles to another virtual terminal and starts another transaction in the same region which also uses a COMAREA, the second transaction will, on some occasions, obtain the COMAREA that was established by the first transaction. This can cause unpredictable results.

- When Pseudo Terminal IDs <u>are</u> in use, for CICS 1.7 and above, you should use the "shippable" TCT option of CICS. This option allows the remote CICS region to accept whatever Terminal ID is sent from the Terminal Owning Region. For versions 3.0 and above of CICS, shippable TCT is required (JUGLMRO is no longer available for CICS versions 3.0 and above).

- For all CICS releases <u>prior</u> to 1.7 or if shippable TCTs are not in use, if Pseudo Terminal IDs <u>are</u> in use, something must be done in the remote regions in order to recognize the pseudo terminal ID and correlate it with the real sending terminal ID. There are two approaches to this problem:

1) Defining dummy remote TCT entries.
2) Using the JUGLMRO program.

DEFINING DUMMY REMOTE TCT ENTRIES

Generate dummy terminal entries in the remote CICS Terminal Control Table (TCT), with terminal ID's that match the Pseudo ID's in the sending region. In order to control the specification of Pseudo ID's, code the Auto-Start table, pre-defining all Pseudo ID's that will be in use in the sending region, thereby eliminating the possibility that a terminal operator will specify an ID (at JUGGLER initiation) that has not been defined in the remote region.

If this approach is used, you need to define the dummy terminal entries in the remote system's TCT only, not in the sending region (terminal-owning region).

[Note]: If PSEUDO IDS=YES is coded in the User Option Table, the generated Pseudo IDs contain lower-case characters (see the discussion of *GENERATED PSEUDO TERMINAL IDS* in section 10 - *CUSTOMIZATION)*. Unless you have an editor which will let you enter lower-case, it is difficult to code these values in the remote TCT. We suggest, therefore that you explicitly define all terminals in the remote regions using Auto-Init table statements.

This approach provides total flexibility for all CICS-JUGGLER functions. Each dummy remote terminal entry requires 20 bytes of storage in the remote TCT, so storage overhead should not be a major consideration.

However, if your installation contains a large number of terminals, and coding the Dummy TCT entries is a problem, you may want to use the second approach, below.

USING THE JUGLMRO PROGRAM

The installation tape includes the JUGLMRO program as one of the link-edited object modules. Note that JUGLMRO cannot be used in version 3 of CICS; you must use shippable TCTs.

The JUGLMRO program must be link-edited and defined in the remote regions, along with the Auto-Start table, as described below. This program provides a facility in the remote system CICS to recognize a Pseudo ID and use it in building the surrogate terminal entry, thereby eliminating the need for dummy TCT entries.

Use of the JUGLMRO program in the remote regions will provide full flexibility and support for all CICS-JUGGLER with no restrictions:

To install and activate the JUGLMRO program, perform the following steps.

a). The JUGLMRO program is link-edited along with the other programs during installation. If the remote regions use the same Core-Image or Load library as the terminal owning regions, nothing more need be done. If different libraries are in use for the remote regions, copy or move the JUGLMRO program to the desired library. This could be performed using LIBR in DOS/VSE or TSO in MVS.

b). Define the VSAM file, JUGLFIL, which contains all customization statements, as a remote file available to the AOR. JUGLMRO must have access to this file.

[Note]: You must define your Auto-Init table statements using either the TERM keyword or using USER=*. Individual or group Auto-Init table statements that define operators with the USER keyword may be used in the Terminal Owning Region, but will be ignored by the JUGLMRO program in the remote regions. For total flexibility, we recommend that you have a TERM=* or USER=* statement as the last (or only) Auto-Init table statement.

c). Define the following PPT entries in the remote region only:

```
DFHPPT  TYPE=ENTRY,PROGRAM=JUGLMRO,RES=YES

DFHPPT  TYPE=ENTRY,PROGRAM=JUGLTBL,RES=NO
                        (only needed if PSGNXIT is coded)

DFHPPT  TYPE=ENTRY,PROGRAM=JUGLAUXL,RES=NO

DFHPPT  TYPE=ENTRY,PROGRAM=DFHPLTxx
                        (xx=PLT SUFFIX, below)
```

d). Assemble and Link-Edit a system initialization Program List Table (or add JUGLMRO to an existing table) as follows:

```
DFHPLT  TYPE=INITIAL, SUFFIX=xx
DFHPLT  TYPE=ENTRY,PROGRAM=JUGLMRO
DFHPLT  TYPE=FINAL
```

e). Add the following System Initialization override parameter to your CICS startup JCL for the remote region(s):

```
PLTPI=xx
```

where xx is the DFHPLT suffix. After completion of these steps, you should receive the following message during the CICS startup in the remote region:

WMR0100. JUGLMRO SUCCESSFULLY ACTIVATED

You may now activate CICS-JUGGLER in the Terminal Owning Region, using Pseudo Terminal IDs. Full separation of terminal IDs is provided in the remote region and you can have any mixture of transaction types (conversational, pseudo-conversational and non-conversational) running on one terminal.

MULTIPLE TERMINAL OWNING REGIONS

If there is only one Terminal Owning Region and one or more application (remote) regions, the distribution of programs when using JUGLMRO is as follows:

- All CICS-JUGGLER programs must be installed in the Terminal Owning Region.
- JUGLMRO and JUGLAUXL must be installed in each remote region. If the PSGNXIT operand is coded, then JUGLTBL must be installed as well.

If there are multiple Terminal Owning Regions where a given region can be both Terminal Owning and remote, you must install all CICS-JUGGLER programs in all regions. The rules to follow are:

- If a region has real terminals defined in its TCT (Terminal Owning Region), that region must have the JUGGLER, JUGLMAIN, JUGLENAB, JUGLINIT and optionally JUGLTBL programs defined locally in its PPT and PCT.
- If a region has remote terminals defined in its TCT (either exclusively or in addition to real terminals), that region must have the JUGLMRO, JUGLAUXL, and optionally JUGLTBL programs defined in its PPT and PCT.
- The VSAM control file, JUGLFIL, must be available to all regions.

[Note]: None of the above considerations apply if shippable TCTs are in use. In that case, install all CICS-JUGGLER programs in every TOR. Either use separate VSAM files or define the same file as accessible from multiple TORs, and nothing more is required. Do **not** define JUGLMRO anywhere.

RUNNING JUGGLER WITH THE DOS VSE INTERACTIVE INTERFACE

For DOS VSE users where the Interactive Interface is in use, CICS-JUGGLER provides three methods of operation. The decision is whether to have an Interactive Interface selection panel (menu) in each virtual terminal or only one and how to control the operator sign-on procedure.

The three methods of operation are as follows:

1). Run with a selection panel in the first virtual terminal only, using only one operator sign-on.

With this method, the selection panel can be used to initiate transactions in the first session only. In all other sessions, transactions must be started using transaction code entry. When you are in the other sessions and try to initiate the Interactive Interface, you receive a message from CICS-JUGGLER to the effect that multiple interactive interface menus are not allowed.

[Note]: This method does not require that Pseudo Terminal IDs be in use. Each virtual terminal may have the same ID as the real terminal or it may have a unique ID.

2). Run with the same selection panel in all virtual terminals, using only one operator sign-on.

With this method, the selection panel that was activated when you first signed-on to CICS is automatically started in each virtual terminal the first time you toggle into that terminal. Each time you press PF3 to return to the selection panel, the selection panel is restored, no matter what virtual terminal you are in.

Full support for all options of the selection panel is provided with the following restriction:

If you are using the system supplied selection panel, you can not select an option which activates ICCF from more than one virtual terminal at a time. You can have multiple versions of ICCF active, however, by using PF6 to exit the selection panel and starting ICCF from a transaction code, as long as you use a different ICCF log-on.

If you exit a selection panel using PF3, which signs you off from CICS, you can not sign back on using the same CICS sign-on. You must either use a different sign-on, sign-off of all sessions, or do a JUGL,OFF.

[Note]: This method requires that Pseudo Terminal IDs be in use. Each virtual terminal must have a unique ID.

3). Run with the same or different selection panel in each virtual terminal using a different CICS sign-on in each session.

With this method, the first time that you toggle into an unused virtual terminal, you are presented with the CICS sign-on screen. You must sign-on with a unique operator name and password in each session. The selection panel associated with this sign-on will be started in this virtual terminal. This can be the same panel or a different panel from the one started when you first signed-on to CICS.

All options of every selection panel are supported in each virtual terminal, including multiple sessions of ICCF started from the system supplied selection panel.

If you exit a selection panel using PF3, which signs you off from CICS, you can sign back on using the same CICS sign-on that was previously in use in that virtual terminal (or with a sign-on that is unique from the remaining sessions).

If you do a JUGL,OFF command, you will automatically be signed-off from all virtual terminals except the one you are currently in.

[Note]: This method requires that Pseudo Terminal IDs be in use. Each virtual terminal must have a unique ID.

INSTALLING THE MULTIPLE INTERACTIVE INTERFACE FEATURE

You can choose one method of operation with the Interactive Interface that all terminals will use, or you can mix the three methods by terminal, if desired. The User Profile Table is used to specify the type of operation desired. In addition, if method 2 or method 3 is to be used, you must install the JUGLVSP program in the same CICS partition with the JUGGLER program. If you do nothing other than install the basic package, method 1 (selection panel in virtual terminal 1 only) is the default.

CHOOSING THE METHOD OF OPERATION

You must code the VSE INTERFACE SUPPORT operand (on the User Profile table) as YES, in order to enable the use of the Interactive Interface. Code one of the following operands on each User Profile table:

- VSE INTERACTIVE INTERFACE=NO This specifies that no multiple Interactive Interface sessions be allowed for this terminal. Method 1, described above, will be in force for all terminals using this profile.
- VSE INTERACTIVE INTERFACE=YES This specifies that Method 2, described above, will be in force for all terminals using this profile. Multiple selection panels are supported using only one operator sign-on.
- VSE INTERACTIVE INTERFACE=SIGNON This specifies that Method 3, described above, will be in force for all terminals using this profile. Multiple selection panels are supported, requiring a different operator sign-on in each session.

  [Note]: You must define Pseudo Terminal IDs in the Auto-Init table entries if the VSE INTERFACE SUPPORT value for that entry is YES or SIGNON.

ACTIVATING CICS-JUGGLER FROM A SELECTION PANEL

There are two ways to activate CICS-JUGGLER when using Interactive Interface selection panels:

1). Exit the selection panel using PF6 and start CICS-JUGGLER the normal way with a JUGL,ON command. After activating it, when you press PF3 you will return to the selection panel. When you press the Toggle key to move to the next virtual terminal, either a blank screen, the selection panel or the sign-on screen (depending on the VSE INTERFACE SUPPORT option chosen for this terminal) will be displayed.

2). Put CICS-JUGGLER as a selection on a selection panel. Using the PANEL TAILORING option of the system supplied selection panel, you can add CICS-JUGGLER as one of the selections to any selection panel.

To do this, first define an Application Profile which will start CICS-JUGGLER. You must use the ATTACH TRANSACTION WITH DATA option, the transaction code is "JUGL", and the data to be passed to the program is ',ON' (in uppercase). Then add a sequence number to a selection panel and provide the name of the application profile for that sequence number.

Upon completion you will be able to select CICS-JUGGLER from the selection panel and it will be activated. Likewise, you could have the JUGL,OFF command be a selection panel option, if desired.

CONSIDERATIONS FOR USING ICCF AND CICS-JUGGLER IN VSE

Many of the options of the system supplied selection panel for VSE invoke ICCF. As previously mentioned, you can not enter ICCF from two selection panels of the same physical terminal unless you are using two different operator sign-ons. This is because VSE/SP and VSE/ESA ICCF will not allow two sessions to be started using the same log-on, as is possible on non-SP systems.

You can, however, start one ICCF session from a selection panel, toggle, exit the selection panel with PF6 and start another ICCF session by entering the ICCF transaction code and logging on with a different Operator ID. When you do this, you must log off from ICCF and press PF3 in order to return to the selection panel.

When you terminate CICS-JUGGLER with a JUGL,OFF command, CICS-JUGGLER checks all virtual terminals to insure that ICCF is not active in any one of them, and will not allow you to JUGL,OFF if it is. If you could terminate CICS-JUGGLER with an active ICCF session, you would not be able to log back on to ICCF with that Operator ID until CICS is recycled.

A somewhat confusing situation can exist if the following sequence of events occurs:

1). ICCF is entered from a selection panel in any virtual terminal, let's say terminal number 2.

2). PF3 is then pressed (still in the same virtual terminal), causing a return to the selection panel. At this point, and automatic log-off of ICCF has <u>not</u> yet occurred, and will not occur until some non-ICCF function is invoked from the panel, such as exiting the panel with PF6.

3). The toggle key is pressed to move to another virtual terminal (terminal 3).

4). A JUGL,OFF command is issued in terminal 3.

5). Message JUGL0010 is displayed indicating that ICCF is still active in terminal 2. This does not appear to be the case, since the selection panel is displayed in terminal 2, but in fact it is true because the automatic log-off for ICCF has not occurred.

In order to complete the JUGL,OFF, you must return to virtual terminal 2, exit the panel using PF6 (or any non-ICCF selection). This will cause ICCF to be logged-off. You may then issue the JUGL,OFF command and it will complete.

OVERHEAD CONSIDERATIONS WITH MULTIPLE INTERACTIVE INTERFACE SESSIONS

We at Unicom Systems don't want to discourage you from using multiple Interactive Interface sessions with CICS-JUGGLER because we think it is a nice feature. However, you should know that there is a price to pay in system overhead.

The Interactive Interface (without CICS-JUGGLER involved) operates pseudo-conversationally by allocating a buffer of static storage from the Shared storage subpool to each terminal using a selection panel. This buffer varies in size depending on the number of selections on the panel, but the size of the buffer for the system supplied panel is just over 2K. This storage does not get released until the operator signs off. Nor does it get written to Temporary Storage or to any file. It stays in virtual memory.

When you use multiple selection panels with CICS-JUGGLER, a buffer of the same size is allocated from the Shared subpool for each virtual terminal. Thus, a physical terminal with four virtual terminals and a selection panel in each session could tie up as much as 8k of the Dynamic Storage Area of CICS.

The only way to free these areas is to sign off from CICS. Thus, it might be preferable to use the VSE INTERACTIVE INTERFACE=SIGNON option and assign multiple sign-ons to each operator, rather than use the VSE INTERACTIVE INTERFACE=YES option. With the VSE INTERACTIVE INTERFACE=YES option, the buffer from the first virtual terminal is copied into all virtual terminals at JUGL,ON time. With the VSE INTERACTIVE INTERFACE=SIGNON option, if the operator has not signed on in a virtual terminal, no storage will be allocated.

SPECIAL-PURPOSE COMMANDS

A series of special commands are available with CICS-JUGGLER which are useful in controlling operation, installing new versions, problem solving, etc. Among these are commands to:

- Back out the 'hooks' in CICS to allow a new copy of the programs to be loaded. (BACKOUT)

- Stop anyone from activating CICS-JUGGLER at a terminal. (STOP)

- Restart CICS-JUGGLER operation (allow activation at a terminal). (START)

- Switch from using Temporary Storage to using MAIN storage above the line for saving data. (MAIN)

- Switch from using MAIN to Temporary Storage for saving data. (TEMP)

- Force deactivation of CICS-JUGGLER at a terminal or all terminals. (PURGE)

- Activate CICS-JUGGLER at a terminal, bypassing the Auto-start table. (INIT)

- Produce a series of transaction dumps to aid Unicom Systems Technical Support in solving a problem. (DEBUG)

[Note]: All of the Special Purpose commands can be secured so that only authorized personnel can issue them. See the discussion on command security in section 1 - *CUSTOMIZATION*, under *THE USER OPTIONS TABLE.*

COMMAND SYNTAX AND USAGE

JUGL,BACKOUT

The BACKOUT command can be used when a new version of CICS-JUGGLER needs to be installed, a PTF applied or for any reason a new copy of either the JUGGLER, JUGLMAIN, or JUGLTBL programs need to be loaded into CICS.

[Note]: If both the JUGL,STOP command and the JUGL,BACKOUT command is issued, a new copy must be performed before CICS-JUGGLER can be reactivated.

The BACKOUT command performs the following functions when invoked:

1) All intercept points ('hooks') in CICS are removed and the normal CICS addresses are replaced.
2) The Terminal Control User Exits are disabled.
3) The 'in-use' condition for the two modules JUGGLER and JUGLTBL is set to zero,which allows a CEMT NEWCOPY command on these modules to be performed.

The end result of the BACKOUT command is to put CICS-JUGGLER back to the same status as if CICS had just been initialized and no one had issued any JUGL command.

The BACKOUT command can only be issued if all users have deactivated CICS-JUGGLER at their terminals by issuing a JUGL,OFF command. In order to prevent anyone from performing a JUGL,ON while BACKOUT and NEWCOPY are being performed, you may want to issue the JUGL,STOP command (described below) before issuing the BACKOUT.

JUGL,DEBUG,xxxx,ON|OFF

The DEBUG command is to be used on request by Unicom Systems Technical Support when some sort of technical problem has arisen with CICS-JUGGLER. The DEBUG command causes a series of transaction coredumps to be output at strategic points in CICS-JUGGLER operation. To use the DEBUG command, enter the following:

```
JUGL,DEBUG,xxxx,ON
```

where xxxx is the terminal ID of the terminal to be used in recreating the problem situation. After executing the transaction which causes the problem on terminal xxxx, enter the following:

```
JUGL,DEBUG,xxxx,OFF
```

This terminates DEBUG mode for terminal xxxx. The transaction dumps will be in the Dump Dataset. Run the Dump Utility and send the dumps to Unicom Systems Technical Support.

JUGL,**INIT** The INIT command can be used instead of a JUGL,ON command when it is desired to activate CICS-JUGGLER without using the values coded in the User Profile for this terminal.

This command will cause the operator to be prompted for each CICS-JUGGLER option.

When the **JUGL,INIT** command is issued, you will be presented with the following prompts, one line at a time. Your answers to these questions will affect how the application works on your terminal.

If no User Profiles are defined, both the JUGL,ON and JUGL,INIT commands have the same effect; however if a User Profile (see section 10 - *CUSTOMIZATION*) has been defined for this terminal (either explicitly or by default), the **JUGL,ON** command will bypass some or all of the following prompts.

After **CICS-JUGGLER** has been initiated by entering "JUGL,INIT", you will be prompted with:

==> *SELECT NUMBER OF TERMINALS*

Key in a number from 2 to 9 for the number of concurrent applications or sessions (virtual terminals) that you desire to have available on this physical terminal and press ENTER. Your choice should be based on the number of terminals you feel will be needed throughout the day. Each virtual terminal is capable of operating any CICS application. One rule of thumb is to select one more terminal than the average number of different applications that you use in a day.

You will next be prompted with:

```
==>   PRESS DESIRED TOGGLE KEY
```

This key will be used to move you sequentially from one virtual terminal to another in a "forward" direction. Press any program function (PF) or program attention (PA) key. This is now the key to press to move from the current virtual terminal to the next. (You should select a key that is not used as a PF or PA key in your applications. We suggest using a PF key for this function.)

The final prompt that you will receive is:

```
==>  ENTER PSEUDO TERMINAL ID'S IF DESIRED
        (1)      (2)      (3)      (4)      (5)
     xxxx    xxxx   xxxx   xxxx   xxxx
```

The number of IDs shown on the line will correspond to the number of virtual terminals that you selected. This is an optional entry, read the following to determine if you need to make the entry.

Certain CICS applications are terminal ID dependent, in the sense that they use your physical terminal ID as part of the processing logic. When more than one of these applications is invoked on different virtual terminals on the same physical terminal, there can be contention, which could cause unexpected results.

The Pseudo Terminal option is provided to prevent this contention by allowing the concurrent use of these types of applications. You may want to consult with your MIS system's programmer to decide if you need to use Pseudo IDs and, if so, what value should be assigned to each one. (Note for MIS personnel: see the discussion entitled *PSEUDO TERMINAL ID CONSIDERATIONS* in section 11 - *SPECIAL CONSIDERATIONS*).

To invoke the Pseudo Terminal feature, do the following:

1) The (1), (2), (3), etc. under the message refers to each virtual terminal that you have selected. The xxxx is pre-filled with the real ID of this physical terminal. You cannot change the ID of terminal number one (1). Enter a unique ID in the ID positions under the remaining virtual terminal positions. The ID must be unique. CICS-JUGGLER will tell you if it is not. If you wish for CICS-JUGGLER to generate the pseudo terminal IDs, enter "*GEN" in the field for terminal 2.

2) After you have keyed the ID's, press ENTER.

If Pseudo Terminals are not desired or needed, simply press ENTER without modifying the terminal IDs. This will cause all ID's to be the same as the physical terminal ID.

At this point, **CICS-JUGGLER** initialization is complete and the User Configuration Display will be displayed next.

JUGL,**MAIN** The MAIN command will cause CICS-JUGGLER to switch from using Temporary Storage to using MAIN storage above the 16M line for saving terminal data.

All users must issue a JUGL,OFF before the MAIN command can be issued.

JUGL,PURGE,xxxx|ALL[,FORCE]
The PURGE command can be used to force CICS-JUGGLER to deactivate from a terminal. The terminal ID to be purged is specified following the PURGE command. Or, if ALL is specified as the terminal ID, all terminals using CICS-JUGGLER will be purged.

PURGE,ALL should be used only in emergency situations when it is necessary to quickly remove all users from CICS-JUGGLER for any reason.

PURGE could be used when it is not possible to issue a JUGL,OFF at the terminal, a terminal error has placed the terminal out of service, for instance. Or perhaps the terminal is unattended and a BACKOUT command needs to be performed.

For DOS users, if ICCF is active in any session except the current session for a terminal, the purge will not take effect unless the command is followed by ',FORCE'. The PURGE,ALL function will stop with message JUGL0048 when a terminal with ICCF active is found. For VSE users, using the FORCE operand can cause an outstanding ICCF session which cannot be logged off. Nor can you log back on to ICCF using the same Operator ID until CICS is recycled.

The PURGE commands (single or ALL) perform the following functions when issued:

1). The terminal is removed from the chain of active JUGGLER terminals and the storage occupied by the chain entry is released.

2). The terminal ID in the TCTTE is reset to the real ID (if Pseudo IDs were in use).

3). Any conversational tasks which were toggled out of for that terminal are ABENDed with an AKCS abend code, and all storage occupied by the task is released.

Usually, the purge function will have no effect on the current transaction in progress on that terminal. The operator may not be aware that a purge was done until he or she attempts to use one of the toggle or window keys, at which time it will be inoperative.

JUGL,START
The START command cancels a previous STOP command and allows users to perform a JUGL,ON command to activate CICS-JUGGLER at their terminal.

[Note]: If both the JUGL,STOP command and the JUGL,BACKOUT command is issued, the JUGL,START command will not restart CICS-JUGGLER. A new copy must be performed before CICS-JUGGLER can be reactivated, in which case, the JUGL,START command will not be neccessary.

JUGL,**STOP** The STOP command will prevent anyone from performing a JUGL,ON command until a subsequent JUGL,START command is issued. The STOP command should be used anytime a BACKOUT, MAIN, or TEMP command needs to be issued.

[Note]: If both the JUGL,STOP command and the JUGL,BACKOUT command is issued, a new copy must be performed before CICS-JUGGLER can be reactivated.

JUGL,**TEMP** The TEMP command will cause CICS-JUGGLER to switch from using MAIN storage to using Temporary Storage for saving terminal data.

All users must issue a JUGL,OFF before the TEMP command can be issued.

USER EXITS AND PROGRAM INTERFACES

CICS-JUGGLER can be customized still further by writing one of the available user exit programs or using one of the interface points where CICS-JUGGLER can be called from a user program.

This section deals with all of the available user exit and program interface points. Each routine is described, along with the reason that you might want to use it, then a coded example is illustrated. Many of these example routines and programs are distributed on the installation tape. They are present in the first file, which contains all of the source code that is distributed with the product.

Most user exits and interface routines in this section are coded in Assembler language. The user should have a good working knowledge of Assembler and CICS coding conventions for both macro level and command level.

[Note]: For versions 3.2 and 3.3 of CICS, it is still possible to do all the macro-level functions described herein, and reference both the CSA and TCA. For more details on this, see *PERFORMING MACRO LEVEL FUNCTIONS IN CICS 3.2 AND 3.3,* later in this section.

WRITING A USER EXIT FOR CICS-JUGGLER

User exit programs may be written to alter the way that CICS-JUGGLER performs in certain cituations by using the Pseudo Terminal ID generation exit.

The exit program must be written in Assembler language. It must be assembled and the object module cataloged in a library or PDS where it can be included with the JUGLTBL module during Link-Edit. In addition, the presence of the exit program must be indicated in the JUGLTBL macro by means of a keyword specifying the exit program name.

You can perform any CICS functions in the user exit program, so long as they are coded in Macro-Level CICS. The exit program can not perform any Command-level functions directly, however it is possible to do a DFHPC TYPE=LINK to a Command-level program that is specified in the PPT.

You can address the terminal (TCTTE) by loading field TCAFCAAA, and obtain any information desired from it. You may need the terminal ID (TCTTETI), the Operator ID (TCTTEOI), the security keys (TCTTESK and TCTESKE) or possibly the terminal I/O area (TCTTEDA). CICS 1.7 users can obtain the Operator ID and other sign-on information by accessing the Sign-on table block addressed by field TCTTESNT. The DSECT for this area can be obtained by coding DFHSNT TYPE=DSECT.

If you need a dynamic work area for your exit program you can either perform a GETMAIN or define a TWASIZE parameter in the PCT for the JUGL transaction and use the Transaction Work Area. CICS-JUGGLER will not use a TWA itself and your exit program will be running under the same TCA as the JUGGLER program.

Each of the user exits, along with coding details, is described below.

PSEUDO TERMINAL ID GENERATION EXIT

This exit will get control when CICS-JUGGLER is ready to generate the Pseudo Terminal IDs during initialization processing as a result of a JUGL,ON or INIT command.

You activate this exit by coding PSGNXIT=xxxxxxx in the TYPE=INITIAL statement of the JUGLTBL macro (see section 10 - *CUSTOMIZATION).* The xxxxxxxx is a 1 to 8 character name of your program. It must be the same as the name on the CSECT or START statement of your program.

Upon entry to the exit program, the following register conventions are used:

R15 - Contains the entrypoint to the exit program.
R14 - Contains the return address.
R13 - Contains a 72-byte savearea address.
R1 - Contains the address of a parameter list.

The parameter list contains the following fields:

Position 0 - Address of a 4-byte field containing the real terminal ID.
Position 4 - Address of a 32-byte area where the exit program is to build the terminal IDs.
Position 8 - Address of a 1-byte hex field containing the number of IDs to build.
Position 12 - Address of the CSA.
Position 16 - Address of the TCA.

The exit program must build the terminal IDs corresponding to virtual terminals 2 thru 9 in the 32-byte area pointed-to by the second word of the parmlist. The terminal IDs must be in increments of 4 bytes each. When complete, bytes 1-4 will be the ID for session 2, bytes 5-8 will contain the ID for session 3, bytes 9-12 will contain the ID for session 4, etc.

You can build all eight pseudo terminal IDs if desired, even though the operator may be using less than nine virtual terminals. Or, if you prefer, use the 1-byte hex value pointed-to by the third word of the parmlist to construct only the number of IDs needed. Keep in mind that a JUGL,INIT command bypasses the Auto-Init table entries, so it is possible for an operator to request more sessions than their User Profile would allow.

Your exit program can decide, if desired, that this operator or this terminal should not use Pseudo IDs at all. In that case, simply move the real ID into all ID positions.

Upon return from the exit program, CICS-JUGGLER will validate that the IDs that your program built are unique among all other real and Pseudo IDs and send the normal JUGL1402 message to the terminal if they are not.

PSEUDO TERMINAL ID GENERATION EXIT EXAMPLE

Following is an example of a Pseudo ID generation user exit. This example includes MVS JCL to assemble the exit and place the object module in an object PDS.

```
//PSGNXIT    JOB    1.'ACCOUNT-DATA',MSGCLASS=x
//ASM        EXEC   PGM=IEV90,PARM='DECK'
//SYSLIB     DD     DSN=CICS.MACLIB,DISP=SHR
//           DD     DSN=SYS1.MACLIB,DISP=SHR
//SYSPRINT   DD     SYSOUT=*
//SYSLIN     DD     DUMMY
//SYSUT1     DD     UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSPUNCH   DD     DSN=CICS.OBJLIB(PSIDGEN),DISP=SHR
//SYSIN      DD     *
             TITLE  'WINDOWS PSEUDO ID GENERATION EXIT'
PSIDGEN      CSECT
             USING  *,15
             STM    14,12,12(13)                SAVE CALLER REGS
             LM     4,6,0(1)                    LOAD 1ST  3  ADDRESSES
             USING  TERMID,4                    ADDRESS REAL TERM ID
             USING  GENAREA,5                   ADDRESS ID BUILD AREA
```

```
              USING  COUNT,6                         ADDRESS ID COUNT FIELD
              MVC    GENAREA(4),TERMID               MOVE REAL ID
              MVC    GENAREA+4(28),GENAREA   PROPAGATE ALL IDS
              XR     0,0
              IC     0,COUNT                         LOAD COUNT
              LA     1,GENAREA                       R1 = SESSION 2 ID
              LA     2,=C'ABCDEFGH'                  R2 = TABLE OF MODIFIERS
GENLOOP       EQU    *
              MVC    0(1,1),0(2)                     MODIFY FIRST BYTE OF ID
              LA     1,4(1)                          BUMP TO NEXT ID
              LA     2,1(2)                          BUMP TO NEXT MODIFIER
              BCT    0,GENLOOP                       DO ALL REQUIRED IDS
              SPACE
              LM     14,12,12(13)                    RELOAD CALLER REGS
              BR     14                              RETURN TO CICS-JUGGLER
              SPACE
              LTORG
              SPACE 3
TERMID        DSECT
              DS     CL4                             REAL TERMINAL ID
GENAREA       DSECT
              DS     CL32                            AREA FOR 8 PSEUDO IDS
COUNT         DSECT
              DS     XL1                             NUMBER IF IDS REQUIRED
              END    PSIDGEN
/*
/&
```

INSTALLING THE USER EXIT PROGRAM

To install your exit program, assemble it with the DECK option and place the object in an object library, as in the examples above. For DOS this would be a CATALR into a relocatable library (name.OBJ for VSE users) and for MVS, the object module should go in an OBJLIB or MODLIB PDS.

Now code the program name in the appropriate exit operand in the User Option Table (ATCHXIT or PSGNXIT or both), then assemble and link-edit JUGLTBL. If the library module name of your exit program is the same as the CSECT or START name, the Linkage Editor will auto-link the exit program with JUGLTBL. Otherwise you must have an INCLUDE statement in the Link-Edit step to retrieve the module.

Check the output from the Linkage-Editor to make sure that you have no unresolved address constants.

Now issue a JUGL,BACKOUT command, then perform a CEMT NEWCOPY for JUGLTBL. The exit program is now installed and will take effect when CICS-JUGGLER is activated.

THE MENU GENERATION FEATURE USER EXIT

The menu generation user exit is used to edit the available selections on a menu. It is called once for each of the selections available on the menu immediately prior to displaying the menu.

The exit program may perform any of the following functions:

- allow the menu selection. (do nothing)

- blank the menu selection from the menu display
- delete the menu selection from screen, and shift all other menu selections up one line
- display the menu selection on the screen, but disable the selection
- alter the selection parameters (i.e. change the transaction code of a transaction to be initiated)
- alter the text that is displayed for a selection

A sample exit program and parameter copybook is distributed on the installation tape. The program name is MENUEXIT and the copybook is CWxx$M32 (where xx is the release of CICS-JUGGLER.

The exit program is passed a parameter list in the commarea. The parmlist contains 3 addresses as described below:

Menu Selection Definition

This is the single selection definition that applies to this call of the exit program. The format of this definition is shown below, and also in the CWxx$M32 source copybook (where xx is the release of CICS-JUGGLER).

```
MNDDSECT        DSECT
   DS     XL1   RESERVED
MNDMNUID        DS      CL8     MENU ID
   DS     XL1   RESERVED
MNDEXIT         DS      CL8     USER EXIT ID
   DS     XL1   RESERVED
MNDQUIT1        DS      CL1     ALLOW QUIT AT LEVEL ONE MENU   CONTAINS:
MNDQYES         EQU     C'/'            TO ALLOW QUIT
MNDQNO          EQU     C' '            TO NOT ALLOW QUIT
   DS     XL1   RESERVED
MNDREFS         DS      CL1     REFRESH MENU AT TRANSACTION END   CONTAINS:
MNDRYES         EQU     C'/'            TO REFRESH MENU
MNDRNO          EQU     C' '            TO NOT REFRESH MENU
   DS     XL4   RESERVED
MNDACTN         DS      CL1     ACTION CODE   CONTAINS:
MNDAOK          EQU     C'O'            ALLOW MENU SELECTION
MNDABLNK        EQU     C'B'            BLANK MENU SELECTION FROM MENU
MNDADEL         EQU     C'D'            DELETE SELECTION FROM MENU
MNDAIGN         EQU     C'I'            DISPLAY SELECTION BUT DON'T             -
                              ALLOW USER TO ACCESS IT
   DS     XL1   RESERVED
MNDENTRY        DS      CL2     SELECTION ENTRY NUMBER
   DS     XL1   RESERVED
MNDTYPE         DS      CL4     SELECTION TYPE   CONTAINS:
MNDTTRAN        EQU     C'T'            C'TRAN' FOR TRANSACTION
MNDTPROG        EQU     C'P'            C'PROG' FOR PROGRAM
MNDTQUIT        EQU     C'Q'            C'QUIT' FOR QUIT
MNDTESC         EQU     C'E'            C'ESC' FOR ESCAPE
MNDTMENU        EQU     C'M'            C'MENU' FOR MENU
MNDTSUBM        EQU     C'S'            C'SUBM' FOR SUBMENU
   DS     XL1   RESERVED

MNDID   DS      CL8     PROGRAM, TRANSACTION OR MENU ID
   DS     XL1   RESERVED
MNDPFKEY        DS      CL4     PFKEY MNEMONIC
   DS     XL1   RESERVED
MNDROW          DS      CL2     CURSOR SELECTION ROW
   DS     XL1   RESERVED
MNDCOL          DS      CL2     CURSOR SELECTION COLUMN
   DS     XL1   RESERVED
MNDINPUT        DS      CL8     SELECTION INPUT
```

```
   DS      XL1       RESERVED
MNDSTYPE        DS        CL6      START TYPE (FOR TRANSACTION OF PROGRAM)
   CONTAINS:
MNDSATCH        EQU       C'A'              C'ATTACH'  FOR ATTACH TRANSACTION
MNDSSTRT        EQU       C'S'              C'START'  FOR START TRANSACTION
MNDSLINK        EQU       C'L'              C'LINK'  FOR LINK TO PROGRAM
MNDSXCTL        EQU       C'X'              C'XCTL'  FOR XCTL TO PROGRAM
   DS      XL1       RESERVED
MNDSWCMD        DS        CL8      WINDOW COMMAND
   DS      XL1       RESERVED
MNDSDATA        DS        CL76     DATA TO BE PASSED TO PROGRAM/TRANSACTION
   DS      XL1       RESERVED
MNDWORK         DS        XL80     WORK AREA
```

Any of these fields (except the reserved fields and MENU ID) may be modified by the exit program. These fields are explained in the following paragraphs.

The MNDACTN field may be used by the exit program to alter the way the menu is displayed or executed. The exit program may pass any of the following values to the menu generator:

O Allow the menu selection.
B Blank the menu selection, so that it will not display on the menu and disable the selection so that it may not be selected.
D Delete the selection from the screen and disable the selection so that it may not be selected.. This will move all subsequent lines up to fill the deleted selection.
I Ignore the selection. This will display the selection on the menu but will not allow this option to be selected.

The MNDWORK field is not used or altered by the menu generator. It is provided as a sort of commarea between calls of the exit program.

The remainder of the fields correspond directly to the Menu Selection Definition Display. For more information on these fields, refer to section 08 - *MENU GENERATION*.

Menu Selection Text
This is the address of the actual menu option text as it appears on the menu display. You may change the text that is to display. However, this address points to an area within the entire menu display. This should be modified with caution since the modification could corrupt the remainder of the menu display. For more information, see the description of the Menu Display, below.

Menu Display This is the menu screen display in a 'flat record' type format. For a model 2 terminal, this will be an area of storage that is 1920 bytes long. All display attributes, whether extended or regular, are represented by a single-byte internal attribute code. Some of the internal attribute codes are displayable characters, but none are characters that can be keyed from a standard U. S. English keyboard.

INSTALLING THE MENU GENERATION FEATURE USER EXIT

To install the menu generation feature user exit, assemble and link the program to a CICS load library, and create a PPT entry (or for RDO a program entry). Then enter the name of the exit program on the appropriate Menu Definition Display, in the USER EXIT field.

PROGRAM INTERFACE ROUTINES

You can call CICS-JUGGLER from a user program to perform various functions. Some of these are fairly specialized functions, but most are routines that have been developed in response to the needs described by other users of the product. Depending on your environment, you may find that you need to develop one or more of these interface calls in order for CICS-JUGGLER to operate properly in all circumstances.

Some of the uses of the program interface routines are summarized as follows:

1). Activate or Deactivate CICS-JUGGLER at a terminal.

2). Perform session commands under program control.

3). Deactivate CICS-JUGGLER at a terminal when a node error occurs.

4). Deactivate CICS-JUGGLER at a terminal prior to an automatic log-off, or for any other reason.

5). Retrieve information about the current session in progress.

6). Obtain the real terminal ID from a pseudo ID.

7). Test to see if CICS-JUGGLER is active on a terminal.

8). Accessing or altering the active user table entry.

9). Change the transaction code to be initiated in a recurring transaction start table.

There are two types of interfaces to CICS-JUGGLER. The first technique is known as the APPLICATION PROGRAM INTERFACE (API) and consists of a direct program LINK, with a command, either in a terminal I/O area or in the TWA. With this method you can perform virtually any command that can be entered at a terminal by an operator.

The second type of interface is a series of special-purpose functions whereby a user program can communicate with CICS-JUGGLER by means of a direct call. These routines are called the BUILT-IN FUNCTIONS, and are described following the API technique, later in this section.

THE APPLICATION PROGRAM INTERFACE

An application interface to CICS-JUGGLER is provided whereby the JUGGLER program may be called from a command level or macro level CICS Application program and the various commands and functions of CICS-JUGGLER may be invoked under program control.

This feature enables programmers to take advantage of the features of CICS-JUGGLER when developing or enhancing their Application Systems. Examples of effective use of the Application Interface would be:

1). Automatically turning CICS-JUGGLER on for a terminal, from a sign-on program or a menu screen, for instance.
2). Automatically turning CICS-JUGGLER off for a terminal.
3) Initializing each virtual terminal with a particular Application Program.

FUNCTIONS AVAILABLE

CICS-JUGGLER, when linked-to by an application program, will perform the functions based on a command stream defined by the calling program. The command name corresponding to each function is as follows:

- Activate (JUGL,ON) for this terminal
- Deactivate (JUGL,OFF pr JUGL,PURGE) for this terminal
- Toggle from the current session to the next session
- Toggle from the current session to the previous session
- Remove CICS-JUGGLER from the system for a NEWCOPY
- Purge a single (or all terminals)
- Start CICS-JUGGLER after a previous stop command
- Stop all users from activating CICS-JUGGLER.

In effect, most of the terminal input commands to CICS-JUGGLER will function under program control, with the additional capability that they may be "streamed". That is, a series of session commands may be interspersed with links to Application Programs.

For instance, the streaming sequence to activate CICS-JUGGLER, starting a user application in three sessions, would be as follows:

1) Link to JUGGLER with JUGL,ON command. If the User Profile for that terminal or operator has all necessary options defined, the user will receive the User Configuration Display, otherwise the operator must respond to the JUGL,ON prompts.

2) Link to an application Program for virtual terminal number 1. The operator will see the display in full-screen mode from this application.

3) Link to JUGGLER with a "toggle" command to move to virtual terminal number 2.

4) Link to another (or the same) application Program. The display from that application will appear.
5) Link to JUGGLER with a "toggle" command to move to virtual terminal number 3.

6) Link to the application Program.

LINKING TO CICS-JUGGLER

To invoke the application Interface, the calling program must issue a CICS Link command to the program "JUGGLER".

In command level, the format of the Link statement is:

```
EXEC CICS LINK PROGRAM(JUGGLER) END-EXEC.
```

In macro level, the format is:

```
DFHPC TYPE=LINK,PROGRAM=JUGGLER
```

PASSING COMMANDS

Commands are passed to the JUGGLER program starting in position 1 of the Transaction Work Area (TWA). This area may be addressed in a Command Level program with the statement:

```
EXEC CICS ADDRESS TWA(ptr-ref) END-EXEC.
```

Positions 1-5 of the TWA must contain the characters '**JUGL,**'. Beginning in position 6, one of the following commands may be used:

1) BACKOUT - Remove CICS-JUGGLER from the system in order to perform a NEWCOPY.
2) ON - Activate CICS-JUGGLER on this terminal.
3) OFF - Deactivate CICS-JUGGLER on the terminal.
4) PURGE,ALL - Purge CICS-JUGGLER from all terminals.
5) PURGE,xxxx - Purge CICS-JUGGLER from terminal xxxx.
6) START - Start CICS-JUGGLER. (Used after a previous STOP command)
7) STOP - Stop all users from activating CICS-JUGGLER on their terminals.
8) TB - Perform a toggle backward to the previous session.
9) TF - Perform a toggle forward to the next session.
10) Tx - Perform a toggle directly to session x.

APPLICATION INTERFACE EXAMPLE

The following example illustrates a Command Level Cobol CICS program that exercises the various functions of the application Interface. This program receives a formatted command from the operator, then links to the JUGGLER module to perform it.

```
**********************************************************************
THIS PROGRAM CALLS JUGGLER TO TEST VARIOUS FUNCTIONS.
***********************************************************************

  IDENTIFICATION DIVISION.
  PROGRAM-ID.          JUGLTST.
  AUTHOR.              CAJ.
  INSTALLATION.        UNICOM SYSTEMS.
  DATE-WRITTEN.        11/11/85.
  DATE-COMPILED.
  SECURITY.            FREE-ACCESS.

  **********************************************************************
     ENVIRONMENT DIVISION.
  **********************************************************************

  DATA DIVISION.
  WORKING-STORAGE SECTION.


  01      INQUIRY.
          05    INQ-MSG                    PIC  X(13)  VALUE 'ENTER COMMAND'.
          05    ERROR-MSG                  PIC  X(12)  VALUE 'ERROR RESPONSE'.
          05    MSG-RESPONSE.
                10    MSG-RESPONSE-FIRST-3         PIC  X(3) VALUE SPACES.
                10    FILLER               PIC  X(11) VALUE SPACES.

  01      MSG-LENGTH                       PIC  S9(4) COMP VALUE +13.
  01      RESP-LENGTH                      PIC  S9(4) COMP VLAUE +14.
  01      ERROR-MSG-LENGTH                 PIC  S9(4) COMP VALUE +12.

  LINKAGE SECTION.

  01      DFHBLLDS.
          02    FILLER                     PIC  S9(8) COMP.
          02    TWA-PTR                    PIC  S9(8) COMP.
```

```
01      TWA-AREA.
        02     TWA-START.
               03     WND0-COMMANDS        PIC  X(19).
               03     RESPONSE REDEFINES JUGL-COMMANDS.
                      05   FILLER                       PIC  X(5).
                      05   COMMAND-IN                   PIC  X(14).
               03     FILLER REDEFINES JUGL-COMMANDS.
                      05   JUGL-RETURN-CODE  PIC  X.

******************************************************************
PROCEDURE DIVISION.
******************************************************************
*
******  ESTABLISH ADDRESSABILITY TO TWA
*
               EXEC CICS ADDRESS TWA(TWA-PTR) END-EXEC.

*
*****   REQUEST AND RECEIVE COMMAND FROM OPERATOR
*
        100-START.
               EXEC CICS SEND FROM(INQ-MSG)
               LENGTH(MSG-LENGTH) WAIT END-EXEC.

               EXEC CICS RECEIVE INTO(MSG-RESPONSE)
               LENGTH(RESP-LENGTH) END-EXEC.

*
*****   INTERROGATE AND PERFORM REQUESTED COMMAND
*
               IF MSG-RESPONSE = 'ON' GO TO JUGL-LINK.
               IF MSG-RESPONSE = 'OFF' GO TO JUGL-LINK.
               IF MSG-RESPONSE = 'T' GO TO JUGL-LINK.
               IF MSG-RESPONSE = 'WIN,IN' GO TO JUGL-LINK.
               IF MSG-RESPONSE = 'WIN,OUT' GO TO JUGL-LINK.
               IF MSG-RESPONSE-FIRST-3 = 'PAN' GO TO JUGL-LINK.
               IF MSG-RESPONSE-FIRST-3 = 'RWI' GO TO JUGL-LINK.
               IF MSG-RESPONSE-FIRST-3 = 'CWI' GO TO JUGL-LINK.
               IF MSG-RESPONSE-FIRST-3 = 'AWI' GO TO JUGL-LINK.
               IF MSG-RESPONSE = 'ALT,OFF' GO TO JUGL-LINK.
               IF MSG-RESPONSE = 'ALT,ON' GO TO JUGL-LINK.
               IF MSG-RESPONSE-FIRST-3 = 'EXP' GO TO JUGL-LINK.
               IF MSG-RESPONSE = 'CANCEL' GO TO RETURN-TO-CICS.
               GO TO 100-START.

*
*****LINK TO THE JUGGLER MODULE
*
JUGL-LINK.
               MOVE 'JUGL' TO JUGL-COMMANDS.
               MOVE MSG-RESPONSE TO COMMAND-IN.
               EXEC CICS LINK PROGRAM('WINDOWS') END-EXEC.
               IF JUGL-RETURN-CODE = HIGH-VALUES
               EXEC CICS SEND FROM(ERROR-MSG)
```

```
                  LENGTH (ERROR-MSG-LENGTH) END-EXEC
                  GO TO RETURN-TO-CICS.

  *
  *****RETURN TO CICS
  *
  RETURN-TO-CICS.
             EXEC CICS RETURN END-EXEC.
```

INVOKING APPLICATION PROGRAMS

You may invoke an application program at any time during a stream of commands to the JUGGLER program by performing a LINK command for the requested application. You should use LINK rather than START because the calling program is attached to the terminal and CICS will not START another program until the calling program returns to CICS.

The following considerations apply when invoking applications via the LINK command:

1). The calling program must have a TWASIZE in the PCT as large as any program being invoked will need.

2). If the invoked program requires a Terminal I/O area, the calling program must be written in Macro Level or running on CICS version 3 in order to build the TIOA and pass the address in TCTTEDA.

3). You may use the COMMAREA of command level to communicate with the invoked program.

4). If the invoked program produces a display to the terminal, the display will appear to the user for an instant, until the next command is passed to the JUGGLER program.

5). If the invoked program is a conversational program, the operator will have to respond to it and the application program return to CICS before the next JUGGLER command will be executed.

EXAMPLE OF APPLICATION INVOCATION

The following example is a macro level Assembler program which activates CICS-JUGGLER, invokes an application in the first session, toggles to the next session, and invokes another application:

```
         TITLE  'TESTDISP - JUGGLER TEST PROGRAM'
*
*****  COPY REQUIRED CICS COPYBOOKS
*
TCTTEAR  EQU    R9
TIOABAR  EQU    R10
         COPY   DFHCSADS
         COPY   DFHTIOA
         COPY   DFHTCTTE
         COPY   DFHTCADS
```

```
JUGLCMDS     DS      CL14
             EJECT
TESTDISP     CSECT
             USING   *,R11
             LR      R11,R14                        SET BASE REG
             L       TCTTEAR,TCAFCAAA               ADDRESS TCTTE
             MVC     JUGLCMDS(14),SPACES      CLEAR COMMAND AREA
             MVC     JUGLCMDS(7),=C'JUGL,ON'   SET COMMAND
             BAL     R2,JUGLLINK                    TURN ON JUGGLER
             SPACE
             BAL     R2,GETTIOA                     GET A TIOA
             MVC     TIOADBA(4),=C'AR10'            SET AR10 TRANCODE
             MVC     TIOADTL,=H'4'                  SET TIOA DATA LENGTH

             DFHPC TYPE=LINK,                       LINK TO ACCOUNTS             X
                     PROGRAM=ACCTRECV'              RECEIVABLE
             SPACE
             MVC     JUGLCMDS(14),SPACES            CLEAR COMMAND AREA
             MVC     JUGLCMDS(11),=C'JUGL,TOGGLE'   SET COMMAND
             BAL     R2,JUGLLINK                    TOGGLE TO SESSION 2
             SPACE
             BAL     R2,GETTIOA                     GET A TIOA
             MVC     TIOADBA(4),=C'AP10'            SET AP10 TRANCODE
             MVC     TIOADTL,=H'4'                  SET TIOA DATA LENGTH
             DFHPC TYPE=LINK,                       LINK TO ACCOUNTS             X
                     PROGRAM=ACCTPAY'               PAYABLE
             SPACE
             DFHPC TYPE=RETURN                      RETURN TO CICS
             EJECT
JUGLLINK     EQU     *
             DFHPC   TYPE=LINK,PROGRAM=JUGGLER
             BR      R2                             RETURN TO CALLER
             SPACE 2
GETTIOA      EQU     *
             MVC     TCASCNB,=H'20'                 SET LENGTH NEEDED
             DFHSC TYPE=GETMAIN,                    GET A TIOA                   X
                     CLASS=TERMINAL,                                             X
                     INITIMG=00
             L       TIOABAR,TCASCSA                ADDRESS THE AREA
             ST      TIOABAR,TCTTEDA                SET ADDRESS IN TCTTE
             BR      R2                             RETURN
             SPACE
SPACES       DC      CL14'  '
             LTORG
             END
```

TESTING SUCCESSFUL EXECUTION

When linking to JUGGLER passing commands in the Transaction Work Area, whether using Macro Level or Command Level, CICS-JUGGLER will pass back a one-byte return code in the first position of the TWA if it is unable to process the command. Either of two return codes may be set:

X'FF' (High Values) - the JUGGLER program will pass back a x'FF' (High- Values) in the first byte of the Transaction Work Area if it is unable to perform the requested command due to a syntactical or logical error in the command as it is found in the TWA.

A syntactical error is simply misspelling or erroneously positioning a command, for instance, JUGL, ONinstead of JUGL,ON.

Logical errors occur when a command is syntactically correct but the current status of the terminal will not allow the command to be executed. An examples of a logical errors would be to specify toggling to three sessions when only two sessions are available.

X'00' (Low Values) - this code is set only in response to the JUGL,OFF command. It is set if CICS-JUGGLER can not be terminated on the terminal because a task is still active in a session. This could only be a conversational transaction unless the REQUIRE CLEAR SESSIONS BEFORE OFF or REQUIRE TRANSACTION END BEFORE OFF option (see section 10 - *CUSTOMIZATION)* is in effect. If this condition occurs, CICS-JUGGLER will display message JUGL1409 on the screen prior to returning to the calling program. This situation can be avoided by passing a JUGL,PURGE command for the current terminal rather than a JUGL,OFF.

Note that if a command can not be executed, control simply returns to the calling program and no action is taken. Thus, it causes no problems as far as CICS-JUGGLER is concerned to simply ignore the error return and continue to execute a command stream.

ACTIVATING AND DEACTIVATING CICS-JUGGLER FROM A USER PROGRAM

It is often desirable to start or stop CICS-JUGGLER at a terminal under program control. Many users prefer that CICS-JUGGLER be active on the terminal as soon as the operator signs on. Others use selection menu processing for all applications and prefer to start or stop CICS-JUGGLER at a terminal by having the operator select a menu option. This can be accomplished with no user programming by specifying the AUTOSTART option in the User Profile. If you have special requirements, however, the following technique can be used.

A common practice is to imbed the activation and deactivation of CICS-JUGGLER in the sign-on / sign-off process. A user program is written using transaction code CESN (or any trancode) that first links to the CICS sign-on program, then links to JUGGLER with a JUGL,ON command. The same program could be invoked using transaction code CESF (or any sign-off trancode) that links to JUGGLER with a JUGL,OFF or JUGL,PURGE command, then links or exits to the sign-off program.

Using this method, the sign-on is always performed before CICS-JUGGLER is activated, which means that all sessions will have the same security and no subsequent sign-ons will be required in the virtual terminals. By turning JUGGLER off before signing off, you insure that no sessions are left active when the operator signs-off (this can also be accomplished using the FORCE PURGE AT SIGNON AND SIGNOFF option in the User Option Table).

By defining the CICS-JUGGLER configuration for the terminal using the customization statements in the User Option Table (see section 10 - *CUSTOMIZATION)*, you can eliminate all operator interaction with CICS-JUGGLER during initialization except for the initial JUGL,ON command. To make the activation and/or deactivation of CICS-JUGGLER totally transparent to the operator, you can invoke the JUGL,ON or JUGL,OFF command from either a Macro Level or Command Level CICS program.

LINKING FROM A MACRO LEVEL PROGRAM

In a Macro Level program, you can use two different techniques to link to the JUGGLER program passing it a command:

1). Obtain a terminal I/O area (TIOA) and construct the command at the start of the TIOA, then perform a Program Control LINK to the JUGGLER program.

2). Build the command starting in the first position of the Transaction Work Area (TWA), then LINK to the JUGGLER program.

With method 1, you can pass any valid JUGL command to CICS-JUGGLER, since the JUGGLER program does not recognize that it is being called from another program in this case. It assumes that the incoming command in the TIOA was entered at the terminal. You can not pass the toggle or control key, however. It must be one of the transaction commands listed in APPENDIX A.

With method 2, the JUGGLER program recognizes the JUGL command at the beginning of the TWA and processes that command as if it had been entered at a terminal.

Macro Level Assembler Examples

Following is an example of method 1, using Macro Level Assembler language:

```
         TITLE  'ACTIVATE CICS-JUGGLER FROM A TIOA'
TCTTEAR  EQU    9
TIOABAR  EQU    10
         COPY   DFHCSADS
         COPY   DFHTIOA
         COPY   DFHTCTTE
         COPY   DFHTCADS
         EJECT
JUGLCALL CSECT
         USING  *,11
         LR     11,14                    SET BASE REG
         L      TCTTEAR,TCAFCAAA         ADDRESS TCTTE
         MVC    TCASCNB,=H'8'            SET LENGTH NEEDED
         DFHSC TYPE=GETMAIN,             GET A TIOA                   X
                CLASS=TERMINAL,                                       X
                INITIMG=00
         L      TIOABAR,TCASCSA          ADDRESS THE AREA
         ST     TIOABAR,TCTTEDA          SET ADDRESS IN TCTTE
         MVC    TIOADBA(7),=C'JUGL,ON'   SET COMMAND
         MVC    TIOADTL,=H'7'            SET TIOA DATA LENGTH
         DFHPC TYPE=LINK,                LINK TO THE                  X
                PROGRAM=JUGGLER             JUGGLER PROGRAM
         SPACE
         DFHPC TYPE=RETURN               RETURN TO CICS
         END
```

Following is an example of method 2, using Macro Level Assembler language:

```
          TITLE  'DEACTIVATE CICS-JUGGLER'
          COPY   DFHCSADS
          COPY   DFHTCADS
          EJECT
JUGLCALL  CSECT
          USING  *,11
          LR     11,14                    SET BASE REG
          MVC    TWACOBA(8),=C'JUGL,OFF'  SET COMMAND
          DFHPC TYPE=LINK,                LINK TO THE            X
                 PROGRAM=JUGGLER            JUGGLER PROGRAM
          SPACE
          DFHPC TYPE=RETURN               RETURN TO CICS
          END
```

LINKING FROM A COMMAND LEVEL PROGRAM

In a Command Level program prior to release 3 of CICS, you can not obtain a terminal I/O area, therefore the only available method is as described in method 2, above, passing the command in the Transaction Work Area. For version 3 of CICS, you can link to the JUGGLER program passing a TIOA as described in the previous example. Following is an example of the coding required using Command Level COBOL:

Command Level COBOL Example

```
IDENTIFICATION DIVISION.
PROGRAM-ID.    JUGLCALL.

DATA DIVISION.
WORKING-STORAGE SECTION.

LINKAGE SECTION.

01        DFHBLLDS.
          02    FILLER                        PIC  S9(8) COMP.
          02    TWA-POINTER          PIC  S9(8) COMP.

01        TWA-AREA.
          05    JUGL-COMMANDS                 PIC  X(8).

   PROCEDURE DIVISION.

*
******    ESTABLISH ADDRESSABILITY TO TWA
*
          EXEC CICS ADDRESS TWA(TWA-POINTER) END-EXEC.


*
*****     LINK TO THE JUGGLER MODULE
*
```

```
           MOVE 'JUGL,ON' TO JUGL-COMMANDS.
           EXEC CICS LINK PROGRAM('WINDOWS') END-EXEC.

*
*****      RETURN TO CICS
*
           EXEC CICS RETURN END-EXEC.
```

[Note]: When defining your calling program to CICS, be sure to specify a TWASIZE of at least 32 bytes.

BYPASSING THE CICS-JUGGLER USER CONFIGURATION DISPLAY

When activating or deactivating CICS-JUGGLER from a user program, it is often desirable to suppress the User Configuration Display that is normally produced in response to the JUGL,ON or JUGL,OFF command. Without doing this, if the calling program generates a terminal display after linking to JUGGLER or if control is passed to another program, such as the sign-on program, which creates a terminal display, the JUGGLER User Configuration Display will appear to the operator as an intermittent flash between displays.

To suppress the User Configuration Display, select the BYPASS STATUS SCREEN option in the User Option Table. See section 10 - *CUSTOMIZATION* for a more complete discussion.

With the BYPASS STATUS SCREEN option in effect, the User Configuration Display can still be invoked by entering the JUGL,INQ or the JUGL (no operand) command. It will not display in response to a JUGL,ON or JUGL,OFF command, whether invoked from a user program or from the terminal, nor will the Auto-Purge messages generated by the FORCE PURGE AT SIGNON AND SIGNOFF option display.

BUILT-IN FUNCTIONS

There are five interface routines that can be executed by a direct call to CICS-JUGGLER. These interfaces are used primarily by a number of vendor packages in order to operate properly with CICS-JUGGLER, primarily to handle the pseudo IDs correctly. They can be used by any user program as long as the linkage standards are strictly adhered to. The built-in functions can only be called from an ASSEMBLER program.

The five built-in function routines are:
1). Test if CICS-JUGGLER is active on a terminal.
2). Purge CICS-JUGGLER from a terminal.
3). Obtain the real terminal ID from a pseudo ID.
4). Alter the transaction to be initiated in a recurring session.
5). Obtain addressability to the active user table.

OBTAINING THE BUILT-IN FUNCTION ENTRYPOINT

All of the built-in functions are accessed through a table of entrypoint addresses located at the top of the JUGGLER module. Thus, the technique for using a built-in function interface is to obtain the program entrypoint for JUGGLER, then index into this table to get the entrypoint for the desired routine, then perform a BALR or BASR to the function.

There are two methods of obtaining the program entrypoint for the JUGGLER module. Either can be done from a command-level program, but the second method requires addressability to the CSA, which is still possible in all releases of CICS.

Method 1: Perform a program LOAD. In either macro-level or command-level, you can issue a program control LOAD and obtain the entrypoint for program JUGGLER. If your calling program is not part of a CICS task that will go to normal transaction end when finished, you must do a DELETE (macro-level) or RELEASE (command-level) when finished to reduce the in-use count.

Method 2: You can go through the CSA to obtain the entrypoint for JUGGLER as well as determining if CICS-JUGGLER has been activated at all. The advantage of this method is it bypasses the CICS call which means the overhead is less. If your program is a global CICS exit with lots of activity, this method should be used. Note that the technique documented here is only valid for CICS-JUGGLER release 3.2 or above. Older releases must use the technique documented for that version, which will not work for release 3.2.

The coding for the direct-access method of obtaining the JUGGLER entrypoint is as follows:

```
L      R15,CSATCRWE              Load ZARQ address
CLI    CSACIREL,X'32'            CICS release 3.2 or greater?
BNL    *+8                              Br if yes
ICM    R15,8,=X'00'              Clear high-byte in address
SH     R15,=H'16'                Back-up 16 bytes
CLC    =CL8'WINDOWS',0(R15)      Is Windows hooked?
BNE    EXIT                      No, can't be active
L      R15,12(R15)               Load Windows entrypoint
```

You can eliminate the test for CICS release 3.2 and 3.3 if your calling program is not release independent. For CICS 3.1 and below, the high-order byte of the terminal control read-write routine contains a non-zero value which must be cleared if your calling program could be running in 31-bit addressing mode.

[Note]: For versions 3.2 and 3.3 of CICS, it is still possible to reference the CSA. For details on obtaining the CSA address, see *PERFORMING MACRO LEVEL FUNCTIONS IN CICS 3.2 AND 3.3*, later in this section.

THE BUILT-IN FUNCTION ENTRYPOINT LIST

Once you have the program entrypoint for JUGGLER, you must back-up 32 bytes to the beginning of the address list of built-in function entrypoints. The first five addresses in this list are the entrypoints for interfaces one through five. The next three words contain zeros. The five routines in the list are:

1). Test if CICS-JUGGLER is active on the terminal, return no other information.

2). Purge CICS-JUGGLER from the terminal, aborting any conversational tasks that may be inactive.

3). Obtain the real terminal ID from a pseudo ID.

4). Alter the next transaction to be automatically initiated in a recurring session.

5). Access the CICS-JUGGLER active user table for this terminal.

Each of these routines are described in detail following.

LINKAGE CONVENTIONS

Linkage to all five built-in functions is effectively the same, although interface number 4 requires more information to be passed. All five routines require the following register contents upon entry:

R1 - Contains the address of a 4-character terminal ID.
R13 - Contains the address of a 72-byte task savearea (Do not use the CSA).
R14 - Contains the return address.
R15 - Contains the built-in function entrypoint.

Upon return, all registers contain the same values as when the call was issued, except register 1 and sometimes register 15, as described in the individual routines following. Register 1 will always contain x'00000000' if CICS-JUGGLER is not active on the requested terminal. If it is active, register 1 will contain the information requested by the chosen interface, if applicable.

An example of the coding required to access built-in function number 3, obtaining the real terminal ID, is as follows, assuming R15 contains the entrypoint for the JUGGLER module.

```
SH      R15,=H'32'              Back-up to entrypoint list
L       R15,((3-1)*4)(R15)      Load third entrypoint
LA      R13,SAVEAREA            Point R13 to savearea
LA      R1,EIBTRMID             Point R1 to the terminal ID
BALR    R14,R15                 Perform Interface 3
LTR     R1,R1                   Is Windows active ?
BZ      EXIT                      Br if no
```

In this example, R1 now contains the real terminal ID.

THE BUILT-IN FUNCTIONS, CODING DETAILS AND EXAMPLES

To use any of the built-in functions, first obtain the program entrypoint for the JUGGLER module as described in *OBTAINING THE BUILT-IN FUNCTION ENTRYPOINT* above.

The following examples assume that address is in R15:

This interface simply returns a zero or non-zero value in R1. Zero means CICS-JUGGLER is not active on the terminal, non-zero means it is.

Coding required:

```
SH      R15,=H'32'              Back-up to entrypoint list
L       R15,((1-1)*4)(R15)      Load first entrypoint
LA      R13,SAVEAREA            Point R13 to savearea
LA      R1,EIBTRMID             Point R1 to the terminal ID
BALR    R14,R15                 Perform Interface 1
LTR     R1,R1                   Is Windows active ?
BZ      EXIT                      Br if no
```

This interface will return a zero value in R1 if CICS-JUGGLER is not active on the terminal. Otherwise an internal PURGE commands is issued, which deactivates CICS-JUGGLER from the terminal after first terminating any conversational tasks that may have been toggled out-of. The current session is not disturbed and the terminal ID is reset to the real ID.

Coding required:

```
SH      R15,=H’32’                   Back-up to entrypoint list
L       R15,((2-1)*4)(R15)           Load second entrypoint
LA      R13,SAVEAREA                 Point R13 to savearea
LA      R1,EIBTRMID                  Point R1 to the terminal ID
BALR    R14,R15                      Perform Interface 2
```

This interface will return a zero value in R1 if CICS-JUGGLER is not active on the terminal. Otherwise, R1 contains the real terminal ID (the actual ID, not the address of the ID). In addition, R15 contains the following values ...

BYTE 1 - Binary zero
BYTE 2 - C'Y' if pseudo IDs are in use.
BYTE 3 - Binary value from 1 to 9 representing the number of virtual terminals in use.
BYTE 4 - Binary value from 1 to 9 representing the current session number.

Coding required:

```
SH      R15,=H’32’                   Back-up to entrypoint list
L       R15,((3-1)*4)(R15)           Load third entrypoint
LA      R13,SAVEAREA                 Point R13 to savearea
LA      R1,EIBTRMID                  Point R1 to the terminal ID
BALR    R14,R15                      Perform Interface 3
LTR     R1,R1                        Is Windows active ?
BZ      EXIT                           Br if no
```

This is a special-purpose interface used in application integration. It can be used to make a decision, while in one session, as to what transaction to initiate in another session using a recurring application start-up table. In most cases, the same functionality can be obtained by accessing the active user table with interface 5, determining the pseudo ID of the desired session and issuing a START for a transaction in that session. When the operator subsequently toggles to the session, the started transaction will initiate.

The problem with using START, however, is that it may be possible to issue several START commands before a toggle is done into the target session. In other words, the user program must keep track of what STARTS have been issued for which sessions to avoid unnecessary queueing up of several pending transactions.

By designating a recurring session, one which has an associated transaction which is to initiate every time the session is entered, CICS-JUGGLER will do the start or attach only when the session is activated. With interface 4, you can alter both the transaction and the data to be passed to the transaction based on some criteria in the current session.

For details of the coding involved, see *CHANGING THE TRANSACTION CODE IN A RECURRING SESSION*, later in this section.

This interface will return a zero value in R1 if CICS-JUGGLER is not active on the terminal. Otherwise, R1 contains the real address of the CICS-JUGGLER active user table entry for that terminal. In addition, R15 contains the same information returned in interface 3, as follows:

BYTE 1 - Binary zero
BYTE 2 - C'Y' if pseudo IDs are in use.
BYTE 3 - Binary value from 1 to 9 representing the number of virtual terminals in use.
BYTE 4 - Binary value from 1 to 9 representing the current session number.

The active user table is a table which is maintained of all terminal operators who have activated CICS-JUGGLER on their terminal. Much of the information in the active user table is useful when you are modifying application programs to run in a more integrated fashion in a windowing environment. For details of the format and use of the active user table, see *ACCESSING OR ALTERING THE ACTIVE USER TABLE ENTRY* later in this section.

Coding required:

```
SH      R15,=H’32’            Back-up to entrypoint list
L       R15,((5-1)*4)(R15)    Load fifth entrypoint
LA      R13,SAVEAREA          Point R13 to savearea
LA      R1,EIBTRMID           Point R1 to the terminal ID
BALR    R14,R15               Perform Interface 5
LTR     R1,R1                 Is Windows active ?
BZ      EXIT                     Br if no
USING   WINTABLE,R1           Address the active user table
```

USES FOR THE BUILT-IN FUNCTIONS

There are many uses of the built-in functions described above. Some are listed in the following examples along with the coding required:

DEACTIVATING CICS-JUGGLER WHEN A NODE ERROR OCCURS

The node error program in CICS (DFHZNEP) controls the action to be taken when a terminal error occurs. The *CICS CUSTOMIZATION GUIDE* contains instructions for modifying this program to conform to your environmental specifications.

It is good practice, and sometimes critically important, to add some logic to DFHZNEP to deactivate CICS-JUGGLER at a terminal any time a terminal error occurs where DFHZNEP allows or forces the terminal operator to be logged off. Failure to deactivate CICS-JUGGLER when an error resulting in a log-off occurs can produce the following unsatisfactory conditions:

1) There is a potential security breach here. If the terminal is logged off because of an error, CICS-JUGGLER does not have a chance to go through its session termination logic, therefore the session termination options are not performed. If another operator logs on to the terminal and presses the toggle key, they will continue in whatever transaction was in progress in that virtual terminal prior to the terminal error occurring. This may be a transaction that they should not be authorized to use.

Note that this condition can also be controlled with the FORCE PURGE AT SIGNON AND SIGNOFF option, but many people prefer the technique of deactivating CICS-JUGGLER from DFHZNEP, in order to clean everything up when the error occurs, rather than waiting till the next signon.

2) With some security packages, this condition can result in a storage violation in CICS. This is because CICS-JUGGLER has saved an address pointer from the TCT which points to a security system control block. When the forced log-off occurs, the security package releases its control blocks for that terminal. Upon re-entering a previously saved session in CICS-JUGGLER, the old address of the control block is restored in the TCT, which usually will no longer point to the same or even a valid security control block.

   Security packages that are known to result in this situation are ACF2, RACF and TOP-SECRET. As a general rule, if you are using any package other than normal (non-RACF) CICS security, you should install this interface.

CODING THE DFHZNEP INTERFACE

There are two sample programs in the source file of the installation tape called JUGLNEPC and JUGLNEPM. JUGLNEPC is a command level DFHZNEP example, and JUGLNEPM is macro-level. You may use the appropriate program or simply insert the necessary code in your DFHZNEP program.

Locate the point in DFHZNEP where the terminal is to be disconnected. At this point, or anywhere in the main logic of DFHZNEP, insert the following instructions:

Note that you should test for the call to DFHZNEP that occurs at session start-up and skip the purge call. Otherwise you can have problems at log-on time.

```
         CLI     TWAEC,TCZOPSIN                    Session just opened?
         BE      NEP0400                           Yes, skip purge
         EXEC    CICS,HANDLE,CONDITION,PGMIDERR(NEP0400).
         EXEC    CICS,LINK,PROGRAM(‘JUGLZNEP’),COMMAREA(TWATCTA).
NEP0400  EQU     *
         .
         .
         DFHEISTG
         DFHEIEND
         END     DFHZNENA                          See note below
```

[Note]: In the CICS 3.2 and 3.3 default version of DFHZNEP, which is called DFHZNEPX, it was necessary to add this label to the END statement for the linkage editor to compute the correct entrypoint to the program. Check your link-edit output - the entrypoint should be at offset 48.

```
         CLI     TWAEC,TCZOPSIN                    Session just opened?
         BE      NEP0400                             Yes, skip purge
         DFHPC TYPE=LINK,PROGRAM=JUGLZNEP,COND=YES  PURGE  JUGLZNEP
NEP0400  EQU     *
```

DEACTIVATING PRIOR TO AUTOMATIC LOG-OFF OR SIGN-OFF

Many security packages and VTAM session managers can perform an automatic log-off or automatic sign-off at a terminal after a specified period of inactivity. Or, you may have an in-house program or CICS modification which performs this activity. It is usually desirable to deactivate CICS-JUGGLER when this occurs.

In many cases, the action taken to force the terminal to log-off generates a node error in CICS and the DFHZNEP program is invoked. If this is the case, the interface logic described above in *DEACTIVATING CICS-JUGGLER WHEN A NODE ERROR OCCURS* will suffice. If there is any question, a CICS auxiliary trace of the automatic log-off will show you if DFHZNEP is getting invoked.

If DFHZNEP is not getting invoked, your security package or session manager may have a user exit point where you could place this interface code. If that is not the case, and you feel that it is necessary to handle this condition, please notify both Unicom Systems and the vendor of the product in question. There are several vendor packages on the market which currently interface to CICS-JUGGLER for one reason or another, and most vendors are quite willing to insert this interface logic into their product. Two security packages which currently call CICS-JUGGLER prior to performing an automatic log-off are ALERT-CICS, and CA-SENTINAL.

OBTAINING THE REAL TERMINAL ID FROM A PSEUDO ID

When pseudo terminal IDs are in use, it is often desirable to obtain the real terminal ID, when all that is known is the pseudo ID.

Interface 3 is currently used by several vendor packages which need to maintain information about the terminal by the real ID, and cannot process a pseudo ID. Among these products are DATAPACKER, SUPER-OPTIMIZER, ALERT-CICS, and CTOP-III. If you are using ACF2, there is a user exit available in it called the TERMINAL IDENTIFICATION EXIT, into which this interface can be inserted. For release 5.0 of ACF2 and above, this interface is necessary for proper operation.

CODING THE ACF2 TERMINAL ID RETRIEVAL EXIT

For ACF2 users, there is a sample program in the source file of the installation tape called JUGLACF2. You may use this program as your terminal identification exit or simply insert the necessary code in your program.

The interface code for this routine is exactly the same as interface 3, described *BUILT-IN FUNCTIONS*, above. Simply load the address of the known terminal ID from wherever it exists (EIBTRMID if this is a command level program) into register 1 and call the JUGGLER program as illustrated. The real terminal ID will be returned in register 1.

ACCESSING OR ALTERING THE ACTIVE USER TABLE ENTRY

The active user table is a table that is maintained by CICS-JUGGLER of all terminal operators who have activated CICS-JUGGLER on their terminal. Much of the information in the active user table is useful when you are modifying application programs to run in a more integrated fashion in a windowing environment.

There is a general purpose interface routine that allows you to call the JUGGLER program and get addressability to the active user table entry for the current terminal in use. Once addressability is obtained, you can use any of the information in the table to make decisions in your program, or in some cases, you may modify certain information to cause CICS-JUGGLER to act differently.

THE ACTIVE USER TABLE

The exact format and contents of the active user table entry are contained in the source module called JUGLINT5, which is distributed on the installation tape.

In general, the active user table contains the following information:
1) The real terminal ID and all pseudo IDs in use.
2) The aid byte of the toggle-forward key.
3) The aid byte of the toggle-backward key.
4) The aid bytes of all direct session keys
5) The total number of sessions allocated to the terminal.
6) The current session number.
7) The current transaction in progress in each session.
8) The current program in progress in each session.
9) The cursor position of the last terminal input message.

[Note]: The Active User Table Entry is located above the 16M line; therefore, your program must be operating in 31-bit addressing mode to access it.

CODING THE ACTIVE USER TABLE ACCESS INTERFACE

There is a sample program in the source file of the installation tape called JUGLINT5. You may use this routine to insert the necessary code in your program:

```
         EXEC  CICS,LOAD,PROGRAM(‘WINDOWS’),
             ENTRY(R15).                Get Windows entrypoint
         SH    R15,=H’32’               Back-up to entrypoint list
         L     R15,((5-1)*4)(R15)       Load fifth entrypoint
         LR    R2,R13                   Save current R13
         LA    R13,SAVEAREA             Point R13 to savearea
         L     R1,TWATCTA               Load TCTTE address
         BALR  R14,R15                  Perform Interface 5
         LR    R13,R2                   Restore R13
EXIT     EQU   *
         .
         .
         .
         DFHEISTG
SAVEAREA DS    18F
*
*  Upon return, R1 contains the address of the active user table entry
*
*  And R15 contains the following values ...
*        BYTE 1  -  Binary zero
*        BYTE 2  -  C'Y' if pseudo IDs are in use.
*        BYTE 3  -  Binary value from 1 to 9 representing the number of virtual terminals in use.
*        BYTE 4  -  Binary value from 1 to 9 representing the current session number.
```

USES FOR ACCESSING THE ACTIVE USER TABLE

Some of the possible uses for obtaining the information in the active user table might be as follows:

1) Obtaining the pseudo ID of another session in order to start a transaction destined for that session.

2) Determining what transaction is active in another session in order to allow or disallow certain activity in this session.

3) Using the last input cursor position to "pick" an item from the incoming terminal I/O area, to use as the key of a transaction to be routed to another session.

ALTERING THE ACTIVE USER TABLE ENTRY

You can change some of the information in the active user table under program control. However, you must be very careful not to change anything else, as doing so could cause an abnormal termination or other unpredictable results. The fields that may be changed are:

1) The aid byte of the toggle-forward key.
2) The aid byte of the toggle-backward key.
3) The aid bytes of all direct session keys
4) The total number of sessions allocated. (You can decrease this but not increase it).

You can disable the function of one or more of the hot keys by moving a x'00' to the corresponding aid byte for that key. This might be useful if you wanted to control the point at which you will allow an operator to toggle out of an application program.

CHANGING THE TRANSACTION CODE IN A RECURRING SESSION

Transaction start tables (Application Start-up Tables in the JAUX transaction) are normally used to initiate a transaction in a session upon first toggling into that session. If RECUR is specified, however, the transaction will be initiated every time the session is entered via a toggle or window-switch function. This feature allow you to customize and integrate your applications by causing things to happen automatically when a session is entered.

It is sometimes desirable to alter the transaction to be initiated in a recurring session, based upon some logic of a control program running in another session. This can be accomplished with one of the built-in functions, interface number 4. To use this feature, do the following:

1). Define an application start-up table connected to the profile to be used.

2). For the session or sessions to be altered, designate the transaction code to be started first. This can be a dummy trancode if desired, as your program can set the correct one.

3). Code RECUR on all sessions where applicable.

4). If you intend to pass data to the transaction that will change with each invocation, fill the DATA field with dummy information. Completely fill the field so there will be room to alter it with data of different lengths.

5). In your control program which makes the decision as to what transaction should be initiated next in this session, define a parameter list as described below, then call the JUGGLER program using built-in function interface 4, passing the address of the parameter list.

6). This causes JUGGLER to alter the application start-up entry for the designated session with the information passed in the parmlist. The next time the operator toggles into this session, the new transaction will be initiated with any data to be passed to it.

CODING THE AUTO-START TRANSACTION-CHANGE INTERFACE

The following instructions illustrate this procedure:

```
*
*****       Definition of parameter list to pass with interface 4
*
BIF4PARM    DS     0F
BIF4TERM    DS     CL4                  Terminal ID
BIF4SESS    DS     XL1                  Session number to be altered
BIF4TRAN    DS     CL4                  New transaction code to initiate
BIF4METH    DS     CL1                  Method of initiation ...
*                                               A = Attach
*                                               S = Start
BIF4DLEN    DS     XL1                  Length of data to be passed ...
*                                               x'00' = none
*                                               x'FF' = leave as is
*                                               X'01'-X'28' (data length)
BIF4DATA    DS     CL56                 Data to be passed to transaction, if any
            .
            .
            .
*
*****       Set up parameter list to transaction to be invoked
*
            MVC    BIF4TERM,EIBTRMID
            MVC    BIF4SESS,...
            MVC    BIF4TRAN,...
            MVC    BIF4METH,...
            MVC    BIF4DATA,...
                   .
                   .
                   .
*
*****       Required coding to call interface 4
*
            EXEC   CICS,LOAD,PROGRAM('WINDOWS'),                        X
                ENTRY(R15).                 Get Windows entrypoint
            .
            .
            or ...
            .
            .
            L      R15,CSATCRWE         Load ZARQ address
            CLI    CSACIREL,X'32'       CICS release 3.2 or greater?
            BNL    *+8                          Br if yes
            ICM    R15,8,=X'00'         Clear high-byte in address
            SH     R15,=H'16'           Back-up 16 bytes
            CLC    =CL8'WINDOWS',0(R15) Is Windows hooked?
```

```
BNE     EXIT                            No, can’t be active
L       R15,12(R15)                     Load Windows entrypoint
.
.
.
SH      R15,=H’32’                      Back-up to entrypoint list
L       R15,((4-1)*4)(R15)              Load fourth entrypoint
LA      R13,SAVEAREA                    Point R13 to savearea
LA      R1,BIF4PARM                     Load parmlist address
BALR    R14,R15                         Perform Interface 4
LTR     R1,R1
BZ      EXIT                            Exit if Windows not active
```

SAMPLE PROGRAMS ON THE INSTALLATION TAPE

Several programs and routines are distributed on file number one of the installation tape. For DOS/VSE users, these programs are cataloged as sub-library type 'Z'.

Each program is written in Assembler language. The on-line programs and routines use macro-level CICS. Each program contains documentation at the beginning of the source member that describes its function and usage.

To make use of any of these programs, move the desired member to your editor library to make any desired modifications, then assemble and link-edit the member into the desired core-image/load library.

The programs and routines on the tape are:

PERFORMING MACRO LEVEL FUNCTIONS IN CICS 3.2 AND 3.3

1) No file control activity. The DFHFC macros and all associated control blocks have been removed. File I/O must be performed in command level.

2) No GETMAIN CLASS=TEMPSTRG. You must obtain USER storage for DFHTS functions.

3) Must be in 31-bit addressing mode. Most control blocks and nucleus modules are located above the line.

ACCESSING THE TCA AND CSA

The following instructions will locate the TCA in MVS/XA or MVS/ESA:

```
L       R13,X'021C'(R0)                 Load TCA address
L       R13,X'D0'(R13)                  Locate the CSA
L       R13,X'14'(R13)                  Locate the CSA
L       R13,8(R13)                      Locate the CSA
USING   DFHCSADS,R13                    Address the CSA
L       R12,CSACDTA                     Load current TCA address
USING   DFHTCADS,R12                    Address it
```

[Note]: Source copybooks and macro-level macros are located in CICS321.SDFHMAC.

AUXILIARY FUNCTIONS MESSAGES "WA" PREFIX

Messages produced by the Auxiliary functions transaction are prefixed by "WA" followed by a 5-position message number, then the message text. The full message will be listed, followed by an explanation and appropriate action to take.

WA00001. CICS-JUGGLER PASSWORD ERROR, CODE=x

This message indicates that the product has expired or cannot be used because of a password error or installation error. The product is now inactive and cannot be used.

The "x" in the above message will be one of the following error codes:

1 - The current date is greater than the password expiration date.
2 - Module "STSPASS" could not be found in a library in the search string.
3 - The product ID could not be located in STSPASS.
4 - The password in STSPASS is invalid.
5 - Module "STS0100" could not be found in a library in the search string.
6 - STSPASS contains a CPU ID that is invalid for the machine that it is installed on.
7 - The password entered is not valid for the requested product.
9 - An undetermined error has been encountered. Please call Unicom Systems Technical Support at 1-(818)-838-0606.

ACTION: Correct the error if possible, or call Unicom Systems Technical Support at 1-(818)-838-0606.

WA00002. FILE "xxxxxxxx" IS NOT OPEN.

A file defined to CICS-JUGGLER (in the Option table) is not available for the requested function.

ACTION: If the function is to be performed, the file must be made available.

WA00003. FILE "xxxxxxxx" HAS NOT BEEN DEFINED.

The file denoted by xxxxxxxx is not defined in the CICS FCT.

ACTION: Ensure that the file name is spelled correctly and that the file is available to CICS-JUGGLER.

WA00004. SCREEN "xxxxx" IS NOT ON FILE.

The JUGLFIL file contains records that are for CICS-JUGGLER internal use. Amongst these are records that are used to map some of the CICS-JUGGLER screen displays. This message notifies that CICS-JUGGLER was unable to locate a necessary screen record.

ACTION: All neccessary screen records are shipped with the product and are pulled from the installation tape during the installation process. If a screen record has been deleted, you may run the installation step that initializes the JUGLFIL VSAM file with the MODE=REINSTALL keyword.

WA00005. PROGRAM "xxxxxxxx" IS NOT AVAILABLE.

CICS-JUGGLER could not find the program xxxxxxxx. Either the PPT entry is disabled or not present, or the program is not in a CICS load file.

ACTION: Ensure that the program is available to CICS-JUGGLER.

WA00030. PCT ENTRY FOR TRANSACTION xxxx MUST HAVE A TWA SIZE OF AT LEAST 32 BYTES

The PCT entry for the transaction indicated by xxxx must have a TWA size of at least 32 bytes.

ACTION: Using RDO or the DFHPCT macro, increase the TWA size.

WA01003. JUGGLER IS NOT AVAILABLE.

CICS-JUGGLER could not find the program "JUGGLER". Either the PPT entry is disabled or not present, or the program is not in a CICS load file.

ACTION: Ensure that the program is available to CICS-JUGGLER.

WA02001. INVALID SELECTION. PLEASE SELECT AGAIN.

The selection was invalid.

ACTION: Review the valid selections in the Technical Reference Guide, then respond accordingly.

WA02004. VERSION 3.0 USER OPTION TABLE HAS BEEN CREATED.

This message is issued in response to a dynamic creation of the user option table.

ACTION: No action is required.

WA06001. MAKE CHANGE AND PRESS "ENTER" TO UPDATE THE USER OPTION TABLE.

The dynamic User Option table is being viewed.

ACTION: If you wish to change the Option table, change the appropriate field and press ENTER.

WA06002. USER OPTION TABLE HAS BEEN CHANGED.

This message is merely to confirm that a requested change to the User Option table was completed successfully.

ACTION: No action is required.

WA06003. USER OPTION TABLE IS BEING MODIFIED ON ANOTHER TERMINAL

A request to edit the User Option table has failed because the table is being edited on another terminal.

ACTION: Wait until the other user has finished editing the table.

WA07001. PROFILE ID "xxxx" ALREADY EXISTS.

An attempt to create a Profile with the ID of xxxx has failed because another Profile with the same ID already exists.

ACTION: Choose another ID for the new Profile.

WA07002. OPTION TABLE HAS CHANGED SINCE PREVIOUS INPUT. CHANGES IGNORED.

Steps have been taken to prevent more than one person from changing customization options at the same time; however it is possible to defeat this safety measure. If CICS-JUGGLER detects a change in this table (that did not originate from this terminal) while this terminal was editing this table, this message will be displayed.

ACTION: Exit from this table to the main menu, and reenter this table. Then perform the desired changes.

WA07004. THE FIRST PROFILE TABLE ENTRY HAS BEEN DISPLAYED.

While browsing backward through the Profile entries, the first entry on file has been displayed.

ACTION: To browse through more entries you must browse in a forward direction.

WA07005. THE LAST PROFILE TABLE ENTRY HAS BEEN DISPLAYED.

While browsing forward through the Profile entries, the last entry on file has been displayed.

ACTION: To browse through more entries you must browse in a backward direction.

WA07006. PROFILE "xxxx" HAS BEEN DELETED.

In response to a request to delete a Profile record, the record was successfully deleted.

ACTION: No action is required.

WA07007. "xxxx" IS THE DEFAULT PROFILE. IT CANNOT BE DELETED.

In response to a request to delete a Profile record, the request failed because the Default Profile may not be deleted.

ACTION: No action is required.

WA07008. PROFILE "xxxx" HAS BEEN ADDED.

This message merely confirms that the Profile xxxx was successfully added to the table.

ACTION: No action is required.

WA07009. PROFILE "xxxx" HAS BEEN CHANGED.

This message merely confirms that the Profile xxxx was successfully altered.

ACTION: No action is required.

WA07010. PROFILE "xxxx" WAS NOT FOUND.

In response to a Find function, CICS-JUGGLER was unable to locate the record in the User Option table being searched.

ACTION: If multiple files are being used for CICS-JUGGLER then the record may exist in another file.

WA07011. PROFILE "xxxx" IS REFERENCED BY AN AUTO-INIT ENTRY. IT CANNOT BE DELETED

Deletion of the Profile xxxx failed because the Profile is referenced by at least one Auto-Init record.

ACTION: Delete or modify all Auto-Init entries that point to this profile. Then this profile may be deleted.

WA07012. PROFILE ID IS REQUIRED.

While adding a new Profile , the record cannot be added until a new Profile ID is entered.

ACTION: Key a valid profile ID and press ENTER.

WA08004. THE FIRST AUTO-INIT TABLE ENTRY HAS BEEN DISPLAYED.

While browsing backward through the Auto-Init entries, the first entry on file has been displayed.

ACTION: To browse through more entries you must browse in a forward direction.

WA08005. THE LAST AUTO-INIT TABLE ENTRY HAS BEEN DISPLAYED.

While browsing forward through the Auto-Init entries, the first entry on file has been displayed.

ACTION: To browse through more entries you must browse in a forward direction.

WA08009. AUTO-INIT TABLE HAS BEEN CHANGED.

This message merely confirms that the Auto-Init table was successfully altered.

ACTION: No action is required.

WA09001. APPL START ID "xxxxxxxx" ALREADY EXISTS.

An attempt to create an Application Start entry with the ID of xxxxxxxx has failed because another profile with the same ID already exists.

ACTION: Choose another ID for the new Application Start entry.

WA09004. THE FIRST APPLICATION START TABLE ENTRY HAS BEEN DISPLAYED.

While browsing backward through the Application Start entries, the first entry on file has been displayed.

ACTION: To browse through more entries you must browse in a forward direction.

WA09005. THE LAST APPLICATION START TABLE ENTRY HAS BEEN DISPLAYED.

While browsing forward through the Application Start entries, the first entry on file has been displayed.

ACTION: To browse through more entries you must browse in a backward direction.

WA09006. APPLICATION START TABLE "xxxx" HAS BEEN DELETED.

In response to a request to delete an Application Start record, the record was successfully deleted.

ACTION: No action is required.

WA09008. APPLICATION START TABLE "xxxxxxxx" HAS BEEN ADDED.

This message merely confirms that the Application Start record xxxxxxxx was successfully added to the table.

ACTION: No action is required.

WA09009. APPLICATION START TABLE "xxxxxxxx" HAS BEEN CHANGED.

This message merely confirms that the Application Start record xxxxxxxx was successfully altered.

ACTION: No action is required.

WA09010. APPLICATION START TABLE "xxxx" WAS NOT FOUND.

In response to a Find function, CICS-JUGGLER was unable to locate the record in the User Option table being searched.

ACTION: If multiple files are being used for CICS-JUGGLER then the record may exist in another file.

WA09011. APPL START ID "xxxxxxxx" IS REFERENCED BY THE PROFILE TABLE. CAN'T DELETE.

Deletion of the Application Start record xxxxxxxx failed because the Application Start entry is referenced by at least one Profile record.

ACTION: Delete or modify all Profile entries so that no Profiles point to this Application Start entry. Then this record may be deleted.

WA09012. ID IS REQUIRED.

While adding a new record, the record cannot be added until a new ID is entered.

ACTION: Key a valid ID and press ENTER.

WA10004. THE FIRST TABLE ENTRY HAS BEEN DISPLAYED.

While browsing backward through the table, the first entry on file has been displayed.

ACTION: To browse through more entries you must browse in a forward direction.

WA10005. THE LAST TABLE ENTRY HAS BEEN DISPLAYED.

While browsing forward through the table, the last entry on file has been displayed.

ACTION: To browse through more entries you must browse in a backward direction.

WA10009. TABLE HAS BEEN CHANGED.

This message merely confirms that the table was successfully altered.

ACTION: No action is required.

WA11004. AT LEAST ONE FILE TABLE ENTRY IS REQUIRED..

The File table specifies the file that CICS-JUGGLER uses for functions such as mapping the screens of several CICS-JUGGLER screen displays.

ACTION: CICS-JUGGLER must have at least one file.

WA11009. FILE TABLE HAS BEEN CHANGED.

This message merely confirms that the File table was successfully altered.
ACTION: No action is required.

MESSAGE BROADCASTING MESSAGES "WB" PREFIX

Messages produced by JUGLMSG (the Message Broadcasting facility) are prefixed by "WB" followed by a 5-position message number, then the message text. The full message will be listed, followed by an explanation and appropriate action to take.

<u>WB01005. PROGRAM "xxxxxxxx" IS NOT AVAILABLE.</u>

CICS-JUGGLER could not find the program xxxxxxxx. Either the PPT entry is disabled or not present, or the program is not in a CICS load file.

ACTION: Ensure that the program is available to CICS-JUGGLER.

<u>WB02004. THE FIRST MESSAGE DEFINITION HAS BEEN DISPLAYED.</u>

While browsing backward through the message definitions, the first definition on file has been displayed.

ACTION: To browse through more entries you must browse in a forward direction.

<u>WB02005. THE LAST MESSAGE DEFINITION HAS BEEN DISPLAYED.</u>

While browsing forward through the message definitions, the last definition on file has been displayed. This message also appear on the Status Display if no active messages are found.

ACTION: To browse through more entries you must browse in a backward direction.

<u>WB02008. MESSAGE "xxxxxxxx" HAS BEEN ADDED.</u>

This message merely confirms that the message definition xxxxxxxx was successfully added to the file.

ACTION: No action is required.

<u>WB02009. MESSAGE "xxxxxxxx" HAS BEEN CHANGED.</u>

This message merely confirms that the message definition xxxxxxxx was successfully altered.

ACTION: No action is required.

<u>WB02010. MESSAGE "xxxxxxxx" WAS NOT FOUND.</u>

In response to a Find function, CICS-JUGGLER was unable to locate the message definition xxxxxxxx in the file being searched.

ACTION: If multiple files are being used for CICS-JUGGLER then the definition may exist in another file.

WB04001. MESSAGE ID "xxxxxxxx" HAS NOT BEEN DEFINED.

The attempted function failed because the message definition could not be located in the file being searched.

ACTION: If multiple message files are in use, the message definition may reside in another file. In which case, you should ensure that the file is defined to CICS-JUGGLER. For more information see section 10 - *CUSTOMIZATION*, under *THE FILE TABLE*. If multiple message files are not in use, then the message definition does not exist.

WB04002. MESSAGE "xxxxxxxx" HAS BEEN SENT.

This message is issued in response to a Send command and confirms that the message has been successfully sent.

ACTION: No action is required.

WB04003. THERE IS NO TEXT FOR MESSAGE "xxxxxxxx"

An attempt to send a message failed because the text record for the message was not found.

ACTION: Create the message.

WB04004. MESSAGE "xxxxxxxx" IS CURRENTLY ACTIVE.

An attempt to send a message failed because the message has been previously sent and not purged, so that the message is still active.

ACTION: The message must be purged before it may be sent again.

WB04005. PROGRAM "xxxxxxxx" IS NOT AVAILABLE.

CICS-JUGGLER could not find the program xxxxxxxx. Either the PPT entry is disabled or not present, or the program is not in a CICS load file.

ACTION: Ensure that the program is available to CICS-JUGGLER.

WB04006. NO AVAILABLE TERMINALS WERE FOUND FOR MESSAGE "xxxxxxxx".

An attempt to send a message failed because CICS-JUGGLER did not find any terminals as specified in the message destination fields of the message definition, or because CICS-JUGGLER has not been initialized in this CICS.

ACTION: Review section 09 - *MESSAGE BROADCASTING* under *FIELDS OF THE MESSAGE DEFINITION DISPLAY*.

WB05001. MESSAGE ID "xxxxxxxx" HAS BEEN PURGED.

This message should not occur, if it does, it is due to an improper deletion of a message broadcast temp storage record.

ACTION: No action is required. The message cannot be retrieved.

WB05003. THERE ARE NO OUTSTANDING MESSAGES.

This message appears upon an attempt to view one's messages, if there are no messages that have been sent the user.

ACTION: No action is required.

WB05004. CICS-JUGGLER CANNOT RUN ON THIS TERMINAL.

A message has been broadcast to a terminal (or user) on which CICS-JUGGLER cannot be activated, due to the coding of the customization table (usually the terminal exclusion table). In order to deliver the message, CICS-JUGGLER must be active or be able to automatically activate on the terminal.

ACTION: If this terminal (or user) is to be eligible to receive message broadcasts, the terminal (or user) must also be able to activate CICS-JUGGLER.

WB08001."xxxxxxxx" IS NOT AN ACTIVE MESSAGE.

Upon attempting to display the Terminal Status of a message definition, CICS-JUGGLER noticed that the message was not active, so this message was issued to inform that there are not any terminals to display.

ACTION: No action is required.

WB08004. THE FIRST TERMINAL HAS BEEN DISPLAYED.

While browsing backward, the first terminal was reached.

ACTION: To browse through more entries you must browse in a forward direction.

WB08005. THE LAST TERMINAL HAS BEEN DISPLAYED.

While browsing forward, the last terminal was reached.

ACTION: To browse through more entries you must browse in a backward direction.

WB09006. MESSAGE "xxxxxxxx" HAS BEEN DELETED.

This message merely confirms that the message definition xxxxxxxx was successfully deleted from the file.

ACTION: No action is required.

WB09007. KEY DATA AND PRESS "ENTER" TO ADD MESSAGE.

In response to New function (creating a new message definition), this message is prompting you to key the desired data then press ENTER. The message will then be added.

ACTION: Key desired data then press ENTER.

WB09011. MESSAGE ID "xxxxxxxx" ALREADY EXISTS.

A request to add a new message definition failed because an old definition with the same message ID already exists.

ACTION: Key a unique message ID then press ENTER.

WB09012. KEY NEW MESSAGE ID AND PRESS "ENTER" TO COPY MESSAGE.

While copying a message definition, this message appears to prompt for the message ID of the new message definition.

ACTION: Key the new message definition ID then press ENTER. The new message definition will then be added.

JUGLHELP MESSAGES "WH" PREFIX

Messages produced by JUGLHELP (the HELP-WINDOWS feature) are prefixed by "WH" followed by a 5-position message number, then the message text. The full message will be listed, followed by an explanation and appropriate action to take.

WH01001. NO HELP TEXT IS AVAILABLE. PRESS CLEAR TO RESTORE TRANSACTION.

No help text that matched the current display was found.

ACTION: Press CLEAR to restore the current transaction.

WH01002. UNEXPECTED NO RECORD FOUND - xxxxxxxxxxxxxxxxxx

An unexpected no-record-found condition has occurred. The x's in the message are the text record key in question.

ACTION: This is probably an internal program error. Notify Unicom Systems technical support department.

WH01004. TERMINAL HAS BEEN SET TO xxxxxxxxxx TRANSLATE MODE

This message indicates successful completion of the UC command to reset the upper or lower case translation. The xxxxxxxxx in the message will be UPPER CASE or LOWER CASE.

ACTION: No action is necessary. You may proceed with any on-line activity.

WH01005. "xxxxxxxxx" IS INVALID IN THIS CONTEXT

The xxxxxxxxx in the message is an extraneous or undefined parameter for the command being issued.

ACTION: Correct the command and re-enter.

WH01007. DEFINE MODE IS xxx

This message is issued in response to the JHLP,DEFINE command. The xxx in the message is either ON or OFF, and specifies whether this terminal is eligible for defining help.

ACTION: No action is required.

WH01008. JUGGLER MUST BE ACTIVE ON THIS TERMINAL FOR DEFINE MODE

An attempt to activate define mode for this terminal has failed because CICS-JUGGLER is not active on this terminal.

ACTION: You must first activate CICS-JUGGLER on this terminal. See the JUGL,ON command.

WH01009. CUSTOMER IS NOT LICENSED TO USE THE HELP-WINDOW FEATURE

The HELP-WINDOWS feature is an additional paid feature of CICS-JUGGLER.

ACTION: If you are licensed to use the HELP-WINDOWS feature, contact Unicom Systems Technical support. If you wish to be licensed, contact your sales representative.

WH02001. INVALID COMMAND CODE - xxxxxxxx

The command indicated by xxxxxxx is not a recognized command or keyword.

ACTION: Correct the command and re-enter.

WH02002. KEYWORD xxxxxxxx REQUIRES =DATA

The keyword indicated by xxxxxxx of the command is syntactically incorrect. This keyword must be followed by and equals sign (=) and an operand.

ACTION: Correct the keyword and re-enter.

WH02003. xxxxxxxx IS AN INVALID PF KEY MNEMONIC

The keyword indicated by xxxxxxx is preceded by 'PF' or 'PA' but is not a valid PF key mnemonic code, or it specifies a PF or PA key which is not valid in HELP-WINDOWS.

ACTION: Correct the keyword and re-enter.

WH02004. KEYWORD xxxxxxxx CANNOT BE FOLLOWED BY =DATA

The keyword indicated by xxxxxxx of the command is syntactically incorrect. This keyword must not be followed by and equals sign (=) and an operand.

ACTION: Correct the keyword and re-enter.

WH02005. NO DATA FOLLOWING = FOR KEYWORD xxxxxxx

The keyword indicated by xxxxxxx of the command is syntactically incorrect. There is no data following the equal sign (=) or the first character after the equal sign is a space.

ACTION: Correct the keyword and re-enter.

WH02006. DATA (xxxxxxx) FOR KEYWORD yyyyyyy MUST BE NUMERIC

The data following the equal sign indicated by xxxxxxxx in the message for the keyword indicated by yyyyyyy is syntactically incorrect. The data must be all numeric.

ACTION: Correct the keyword and re-enter.

WH02007. DATA (xxxxxxx) FOR KEYWORD yyyyyyy IS TOO LONG

The data following the equal sign indicated by xxxxxxxx in the message for the keyword indicated by yyyyyyy is syntactically incorrect. The number of characters in the data field exceeds the maximum allowable for this keyword.

ACTION: Correct the keyword and re-enter.

WH11001. CROSS-DOCUMENT FIELD IS ALREADY ON FILE

The cross-document record that you are attempting to add already exists in the file.

ACTION: Change the system or field name and re-enter.

WH11002. CROSS-DOCUMENT FIELD IS NOT ON FILE

The cross-document record that you are attempting to retrieve does not exist in the file.

ACTION: Correct the record name and re-enter. If you think the record is there, try the NEXT command, then browse forward or backward until you find it, or use the directory to locate it.

WH11003. PRESS "ENTER" TO UPDATE OR PRESS THE "QUIT" KEY

This message confirms entry into update mode for a cross-document record. You can now make changes to the record.

ACTION: Make any desired changes and press ENTER to complete the update, or press the QUIT key to exit update mode.

WH11004. PRESS "ENTER" TO DELETE OR PRESS THE "QUIT" KEY

This message confirms entry into delete mode for a cross-document record. The requested record is displayed on the screen.

ACTION: Press ENTER to complete the delete, or press the QUIT key to exit deletion mode.

WH11005. ENTER DATA AND PRESS "ENTER" TO ADD OR PRESS THE "QUIT" KEY

This message confirms entry into add mode for a cross-document record. You can now enter all data fields for the record.

ACTION: Enter all required fields and press ENTER to complete the add, or press the QUIT key to exit add mode.

WH11006. CROSS-DOCUMENT HAS BEEN ADDED

This message confirms successful completion of a record add for a cross-document record.

ACTION: No action is required, continue with the next maintenance function.

WH11007. CROSS-DOCUMENT RECORD HAS BEEN CHANGED

This message confirms successful completion of a record update for a cross-document record.

ACTION: No action is required, continue with the next maintenance function.

WH11008. CROSS-DOCUMENT RECORD HAS BEEN DELETED

This message confirms successful completion of a record delete for a cross-document record.

ACTION: No action is required, continue with the next maintenance function.

WH11009. END OF X-DOCS USE "PREV" OR "QUIT" KEY

While browsing forward through cross-document records, the end of these record has been reached.

ACTION: You can browse backward with a 'P' command, request help with the help key, or press the QUIT key to exit.

WH11010. ENTER KEY FIELDS TO RETRIEVE RECORD FOR DELETION

A DELETE command has been issued for a cross-document record and the system and field names were not entered, nor were they present in the conversation.

This message can also occur if a name is filled-in from the conversation and the resultant record key does not exist.

ACTION: Enter the necessary key fields to retrieve the desired record to be deleted.

WH11011. SYSTEM HAS NOT BEEN DEFINED

The cross-document record that you are attempting to add contains a system identifier for which no system record exists. You can not add a lower-level record without a complete hierarchy.

ACTION: Correct the system name and re-enter, or add the necessary system record.

WH11012. DOCUMENT HAS NOT BEEN DEFINED

The DOCUMENT field of the cross-document record names a document identifier for which no document record exists.

ACTION: Correct the document name and re-enter, or add the necessary document record.

WH11013. SECTION HAS NOT BEEN DEFINED

The SECTION field of the cross-document record names a section identifier for which no section record exists.

ACTION: Correct the section number and re-enter, or add the necessary section record.

WH11014. SUBJECT HAS NOT BEEN DEFINED

The SUBJECT field of the cross-document record names a subject identifier for which no subject record exists.

ACTION: Correct the subject name and re-enter, or add the necessary subject record.

WH11015. NO RECORD FOR THIS KEY. ENTER VALID KEY AND PRESS "ENTER"

An INQUIRE or UPDATE command has been issued for a cross-document record and the system and field names were not entered, nor were they present in the conversation.

This message can also occur if a name is filled-in from the conversation and the resultant record key does not exist.

ACTION: Enter the necessary key fields to retrieve the desired record.

WH11016. KEY FIELDS HAVE BEEN CHANGED, RECORD HAS NOT BEEN UPDATED

An UPDATE command was issued, but prior to pressing ENTER, one or more of the identifiers in the command was changed.

This message indicates that the record update did not take place.

ACTION: No action is necessary, continue with any activity.

WH12001. THIS RECORD IS ALREADY ON FILE. CORRECT KEY OR PRESS THE "QUIT" KEY

The transaction key record that you are attempting to add already exists in the file.

ACTION: Change the system or field name and re-enter, or press the QUIT key to exit add mode.

WH12002. TRAN KEY IS NOT ON FILE

The transaction key record that you are attempting to retrieve does not exist in the file.

ACTION: Correct the record name and re-enter. If you think the record is there, try the NEXT command, then browse forward or backward until you find it, or use the directory to locate it.

WH12003. NO RECORD FOR THIS KEY, ENTER VALID KEY AND PRESS "ENTER"

An INQUIRE or UPDATE command has been issued for a transaction key record and the transaction ID, screen and field names were not entered, nor were they present in the conversation.

This message can also occur if a name is filled-in from the conversation and the resultant record key does not exist.

ACTION: Enter the necessary key fields to retrieve the desired record.

WH12004. PRESS "ENTER" TO UPDATE

This message confirms entry into update mode for a transaction key record. You can now make changes to the record.

ACTION: Make any desired changes and press ENTER to complete the update, or press the QUIT key to exit update mode.

WH12005. PRESS "ENTER" TO DELETE OR THE "EXIT" KEY

This message confirms entry into delete mode for a transaction key record. The requested record is displayed on the screen.

ACTION: Press ENTER to complete the delete, or press the QUIT key to exit deletion mode.

WH12006. ENTER DATA AND PRESS "ENTER" TO ADD OR THE "EXIT" KEY

This message confirms entry into add mode for a transaction key record. You can now enter all data fields for the record.

ACTION: Enter all required fields and press ENTER to complete the add, or press the QUIT key to exit add mode.

WH12007. TRAN KEY HAS BEEN ADDED

This message confirms successful completion of a record add for a transaction key record.

ACTION: No action is required, continue with the next maintenance function.

WH12008. TRAN KEY HAS BEEN CHANGED

This message confirms successful completion of a record update for a transaction key record.

ACTION: No action is required, continue with the next maintenance function.

WH12009. TRAN KEY HAS BEEN DELETED

This message confirms successful completion of a record delete for a transaction key record. Note that pressing ENTER after a deletion will result in a no-record-found condition. Browsing backward will produce the same result. The only file activity that can be done without changing the command line is to browse forward.

ACTION: You may browse forward with the NEXT key, request help with the help key, or press the QUIT key to exit.

WH12010. END OF TRAN KEYS. USE "PREV" OR "EXIT" KEY

This message indicates that the end of transaction key records has been reached while browsing forward with the NEXT command.

ACTION: You may browse backward with the PREV key, request help with the help key, or press the QUIT key to exit.

WH12011. ENTER KEY FIELDS TO RETRIEVE RECORD FOR DELETION

A DELETE command has been issued for a transaction key record and the transaction ID, screen and field names were not entered, nor were they present in the conversation.

This message can also occur if a name is filled-in from the conversation and the resultant record key does not exist.

ACTION: Enter the necessary key fields to retrieve the desired record to be deleted.

WH12012. SYSTEM HAS NOT BEEN DEFINED

The SYSTEM field for the cross-document identity in the transaction key record contains a system identifier for which no system record exists.

ACTION: Correct the system name and re-enter, or add the necessary system record.

WH12013. X-DOC FIELD IS INVALID. PRESS "ENTER" TO CONFIRM OR "EXIT"

The cross-document identity in the transaction key record does not exist.

ACTION: Correct the cross-document name or system and re-enter, or go ahead and press enter to add this transaction key record with an invalid cross-document identifier.

WH12014. THE "FROM" RECORD COULD NOT BE FOUND. PRESS "EXIT" TO END COPY

The starting record of a COPY function does not exist as entered.

ACTION: Restart the COPY command and enter a valid starting record identifier or press the QUIT key to exit the copy function.

WH12015. TRAN KEY TABLE EXHAUSTED. INCREASE THE TKEYPARM IN HWINTBL

This message should not occur.

ACTION: First try backing-out CICS-JUGGLER, then reinitialize JUGGLER and retry the operation that failed. If the message still occurs, contact Unicom Systems Technical Support.

WH13001. CROSS-DOCUMENT "xxxxxxxxxx" IS NOT DEFINED

The cross-document identity in the transaction key record or as the object of an INCLUDE statement does not exist. The xxxxxxxx in the message identifies the cross-document name.

ACTION: Either add the missing cross-document record or change the reference in the transaction key or the INCLUDE statement.

WH13002. TRANSACTION KEY "xxxxxxxxxx" IS NOT DEFINED

The transaction key denoted by xxxxxxxxxx does not exist.

ACTION: Retry the operation using the Transaction Key directory.

WH13003. THE LAST PAGE OF THIS SECTION/SUBJECT HAS BEEN DISPLAYED

This message indicates that the end of text records for this section or subject has been reached while browsing forward with the NEXT command.

ACTION: No action is required, continue with the next maintenance function.

WH13004. THE FIRST PAGE OF THIS SECTION/ SUBJECT HAS BEEN DISPLAYED

This message indicates that the beginning of records for this section or subject has been reached while browsing backward with the PREVIOUS command.

ACTION: No action is required, continue with the next maintenance function.

WH13005. TEXT RECORD IS NOT ON FILE

The text record requested with a LIST or GET command is not on file.

ACTION: Correct the record number and re-enter, or use the NEXT and PREVIOUS commands to browse the file.

WH13006. <INCLUDE> STATEMENT IS RECURSIVE AND HAS BEEN IGNORED

An INCLUDE statement has been found which references a cross-document record which points to the text records currently being processed.

ACTION: No action is necessary. You may want to locate this INCLUDE statement an correct the command.

WH13007. <INCLUDE> STATEMENT EXCEEDS MAXIMUM ALLOWABLE INCLUDES

The level of nested INCLUDE statements has exceeded the maximum allowed.

ACTION: No action is necessary, however, the text referenced by this INCLUDE statement will not be processed.

WH13008. <INCLUDE> STATEMENT REFERENCES AN INVALID TEXT RECORD

An INCLUDE statement has been found which references a cross-document record which points to a text record that is not on file.

ACTION: No action is necessary. You may want to locate this INCLUDE statement and correct the text reference fields of the cross-document record.

WH13009. <INCLUDE> STATEMENT REFERENCES AN UNDEFINED CROSS-DOCUMENT

An INCLUDE statement has been found which references a cross-document record which is not on file.

ACTION: No action is necessary. You may want to locate this INCLUDE statement an correct the cross-document name, or add the referenced cross-document record.

WH13010. <INCLUDE> STATEMENT IS SPECIFIED INCORRECTLY

An INCLUDE statement has been found which is syntactically incorrect.

ACTION: No action is necessary. You may want to locate this INCLUDE statement an correct it.

WH13011. ALL TEXT FOR FIELD "xxxxxxxxx" HAS BEEN DISPLAYED

This message indicates that the end of text records for this cross-document record has been reached while using the GET command. The xxxxxxx in the message is the cross-document name.

ACTION: No action is required, continue with the next maintenance function.

WH13012. USER EXIT "xxxxxxxx" IN <USERXIT> COMMAND CANNOT BE FOUND

A USERXIT command has been found while processing text records in help access mode. The designated program could not be loaded into memory. The xxxxxxx in the message is the program name.

ACTION: Verify that that the exit program is present in one of the load libraries available for this CICS system. Also verify that a PPT entry is present with the same program name. If problem persists, notify Unicom Systems technical support.

WH13013. RETURNED PARAMETERS FROM USER EXIT "xxxxxxxx" ARE INVALID, RCODE=y

Upon return from a user exit program specified with a USERXIT command in the text, the return parameters set by the exit program are incorrect. The xxxxxxx in the message is the program name. The 'y' in the message is a code indicating the type of error. Meanings of the code are:

1 Invalid status code set.
2 The returned cross-document record is invalid.
3 The returned text key is invalid.

ACTION: Review the specifications for writing a user exit program in the *TECHNICAL REFERENCE manual*. Correct the errors in the program and re-initialize it. If you need help determining the error, notify Unicom Systems technical support.

WH14001. TEXT RECORD NUMBER IS REQUIRED

A text record addition was attempted and no text record number was entered, nor is one present in the conversation.

ACTION: Enter the REC= keyword to supply the text record number and re-enter the command.

WH14002. DOCUMENT MUST BE REORGANIZED BEFORE THIS RECORD CAN BE INSERTED

During text record addition, the next text record after the one previously added contains a record number which is only one greater than the previous number. You cannot add any more records at this point without re-numbering the section.

ACTION: If more text records must be added at this point, use the RENUMBER command to reorganize the text records for this section/subject.

WH14003. PRESS ENTER TO ADD RECORD OR THE "QUIT" KEY

This message confirms entry into add mode for a text record. You can now enter all data fields for the record.

ACTION: Enter all required text data and press ENTER to complete the add, or press the QUIT key to exit add mode.

WH14004. TEXT RECORD NUMBER ALREADY EXISTS

The text record assigned to the record that you are attempting to add already exists in the file.

ACTION: Change the record number, or press PF3 to exit add mode.

WH14005. MAKE CHANGES AND PRESS "ENTER" TO UPDATE RECORD

This message confirms entry into edit mode for a text record. You can now make changes to the record, if desired.

ACTION: Make any desired changes and press ENTER to complete the update, or perform any other function.

WH14006. REQUESTED RECORD IS NOT ON FILE

The text record that you are attempting to retrieve does not exist in the file.

ACTION: Correct the record name and re-enter. If you think the record is there, try the NEXT command, then browse forward or backward until you find it, or use the directory to locate it.

WH14007. PRESS "ENTER" TO DELETE RECORD OR THE "QUIT" KEY

This message confirms entry into delete mode for a text record. The requested record is displayed on the screen.

ACTION: Press ENTER to complete the delete, or press PF3 to exit deletion mode.

WH14012. TEXT RECORD HAS BEEN ADDED

This message confirms successful completion of a record add for a text record.

ACTION: No action is required, continue with the next maintenance function.
WH14013. TEXT RECORD HAS BEEN CHANGED

This message confirms successful completion of a record update for a text record.

ACTION: No action is required, continue with the next maintenance function.

WH14013. TEXT RECORD HAS BEEN CHANGED

This message confirms successful completion of a record update for a text record.

ACTION: No action is required, continue with the next maintenance function.

WH14014. TEXT RECORD HAS BEEN DELETED

This message confirms successful completion of a record delete for a text record.

ACTION: No action is required, continue with the next maintenance function.

WH14015. THE LAST TEXT RECORD IN THIS SECTION/SUBJECT HAS BEEN DISPLAYED

This message indicates that the end of text records for this section or subject has been reached while browsing forward with the NEXT command.

ACTION: No action is required, continue with the next maintenance function.

WH14016. THE FIRST TEXT RECORD IN THIS SECTION/SUBJECT HAS BEEN DISPLAYED

This message indicates that the beginning of text records for this section or subject has been reached while browsing backward with the PREVIOUS command.

ACTION: No action is required, continue with the next maintenance function.

WH14023. PRESS ENTER TO RENUMBER TEXT RECORDS OR THE "QUIT" KEY

This message confirms entry into renumber mode for a series of text records.

ACTION: Press ENTER to start the renumber process, or press the QUIT key to exit renumber mode.

WH14024. TEXT RECORDS HAVE BEEN RENUMBERED

This message indicates successful completion of the RENUMBER command.

ACTION: No action is necessary, continue with any function.

WH14025. TEXT TEXT HAS BEEN FOUND, PRESS "ENTER" TO CONTINUE SEARCH

This message indicates that the data entered as the operand of a SEARCH command has been located. The text record containing the search data is displayed.

ACTION: Press ENTER to continue searching forward in the text from this point, otherwise continue with any function.

WH14026. ALL TEXT HAS BEEN SEARCHED

This message indicates that the end of text records for this section/subject was encountered before the data entered as the operand of a SEARCH command could be found. The search data does not exist in this text in the form that it was entered.

If the search data was previously found, it means that there are no more occurrences of the search data after the last one that was found.

ACTION: Reposition to the beginning of the text for this section/subject and restart the search with a different search string, if desired, otherwise continue with any function.

WH14027. TEXT RECORD REQUIRES ALTERNATE SCREEN SIZE CAPABILITY

The text record being retrieved was created on a terminal with a larger screen size than the terminal attempting to display it and the ALTSCRN option was set when the text record was created. Or, the TCT entry for this terminal does not specify the same alternate screen size as the terminal which created this text record.

ACTION: You cannot view this text record as it is from this terminal. Either use a terminal with the correct alternate screen size or update the text record, removing the ALTSCRN option.

WH14028. xxxxxxxx HEADER RECORD MISSING, CANNOT ADD TEXT

A required hierarchical control record is not present for the system, document, section, subject hierarchy as specified. Text addition cannot be done until the complete hierarchy is present. The xxxxxxxx in the message indicates the type of control record that is missing.

ACTION: Add the required control record, then restart the text record addition function.

WH14029. TRANSACTION KEY "xxxxxxxxxxxxxxxxx" IS NOT VALID

The transaction key denoted by xxxxxxxxxx does not exist.

ACTION: Retry the operation using the Transaction Key directory.

WH18002. THE LAST RECORD HAS BEEN DISPLAYED

This message indicates that all records have been displayed for this directory display.

ACTION: No action is necessary, continue with any function.

WH18003. TRAN KEY "xxxxxxxxxxxxxxxxx" WAS NOT FOUND

While attempting to retrieve text from the transaction key directory, the designated transaction key record could not be found.

ACTION: This is probably a system error although it could be caused by incorrect input on the command line. Attempt the operation again and if the problem persists, notify Unicom Systems technical support..

WH24001. CROSS-DOCUMENT FIELD IS TOO LONG

The cross-document name exceeds the maximum length allowed. The maximum length is 20 characters

ACTION: Correct the field name command and re-enter.

WH24002. AN ERROR HAS BEEN ENCOUNTERED IN THE TEXT COMMAND AREA

The edit sub-command in the text command area is not a valid sub-command.

ACTION: Correct the sub-command and re-enter.

WH24003. <SPACE NN> COMMAND IS SPECIFIED INCORRECTLY

A text command identified as SPACE was found but cannot be processed as it is syntactically incorrect.

ACTION: Correct the SPACE command and re-enter.

WH24004. NO STARTING POSITION HAS BEEN DEFINED FOR CROSS-DOCUMENT

A STOPFLD statement has been encountered naming a cross-document for which no STRTFLD was present.

ACTION: Enter the STRTFLD statement or change the name of the field in the STOPFLD statement if it is incorrect.

WH24005. <STRTFLD> HAS ALREADY BEEN DEFINED FOR THIS CROSS-DOCUMENT

A STRTFLD statement has been encountered naming a cross-document which already has a STRTFLD in the file.

ACTION: Change the name of the field in the STRTFLD statement to a different cross-document name, or delete the duplicate STRTFLD statement.

WH24006. <STOPFLD> HAS ALREADY BEEN DEFINED FOR THIS CROSS-DOCUMENT

A STOPFLD statement has been encountered naming a cross-document which already has a STOPFLD in the file.

ACTION: Change the name of the field in the STOPFLD statement to a different cross-document name, or delete the duplicate STOPFLD statement.

WH24007. SYSTEM ID IS INVALID FOR <STRTFLD>

A STRTFLD statement has been encountered which names a system ID as well as a cross-document field name. The named system ID does not exist.

ACTION: Change the name of the system in the STRTFLD statement to a different name, or add a valid system record with the requested name.

WH24008. SYSTEM ID IS INVALID FOR <STOPFLD>

A STOPFLD statement has been encountered which names a system ID as well as a cross-document field name. The named system ID does not exist.

ACTION: Change the name of the system in the STOPFLD statement to a different name, or add a valid system record with the requested name.

WH24009. SPLIT COMMAND IS NOT VALID FOR TRANSACTION KEY TEXT

You cannot use the SPLIT command when adding or editing fast-path help text. Fast-path text consists of one text record which is attached to the transaction key record. If you need additional text, either change to structured text or add another transaction key record with a higher sequence number.

ACTION: Remove the SPLIT command and take appropriate action as described above.

WH24010. <STRTFLD> IS INVALID FOR TRANSACTION KEY TEXT

You cannot use the STRTFLD or STOPFLD command when adding or editing fast-path help text. Fast-path text consists of one text record which is attached to the transaction key record. Therefore it is not necessary to define a cross-document record.

ACTION: Remove the STRTFLD command and continue with text maintenance.

WH24011. INSERT WOULD EXCEED TWO PAGES, COMMAND IGNORED

An INSERT sub-command as been used when editing fast-path help text. There is insufficient room in the text record to add insert all of the data that was copied or moved. Fast-path text consists of one text record which is attached to the transaction key record. The maximum size is the size of one VSAM record. If you need additional text, either change to structured text or add another transaction key record with a higher sequence number.

ACTION: Remove the INSERT command and take appropriate action as described above.

WH24012. CROSS-DOCUMENT FIELD HAS NOT BEEN DEFINED

The cross-document name specified on an INCLUDE statement could not be located as coded.

ACTION: Verify that the cross-document record exists within the specified system. Correct the INCLUDE statement or add the cross-document record.

WH24013. EXIT NAME MUST BE SPECIFIED IN <USERXIT> COMMAND

A USERXIT statement was found in the text which did not specify a program name.

ACTION: Enter the program name following the USERXIT command. Enclose the entire statement in <> signs.

WH24014. EXIT PARAMETERS IN <USERXIT> COMMAND ARE TOO LONG

The PARM statement following the program name in a USERXIT command exceeds the allowable maximum.

ACTION: Correct the PARM statement. Review the specifications for the USERXIT command if needed.

WH24015. ENDING BLOCK COMMAND NOT ENTERED FOR "xx"

A block deletion, move or copy function was specified by keying a DD, MM or CC in the sub-command area of the editor. The ending DD, MM or CC was omitted when ENTER was pressed.

ACTION: Key the ending block command or remove the starting block command. Note that you cannot perform block functions that span more than one text record.

WH24016. FIELD NAME IS REQUIRED IN STRTFLD COMMAND

A STRTFLD statement was found in the text which did not specify a name.

ACTION: Enter the name following the STRTFLD command. Enclose the entire statement in <> signs.

WH24017. FIELD NAME CANNOT CONTAIN "," OR "="

The field name cannot contain the indicated characters.

ACTION: Correct the field name and press ENTER.

WH25001. TEXT RECORD REORGANIZATION IS IN PROGRESS. PASS x

This is an informational message to report the status of a text record reorganization.

ACTION: No action is required.

WH25002. THERE ARE NO RECORDS IN THIS SECTION/SUBJECT TO RENUMBER

While attempting to perform a text record reorganization, the function was halted because there are no records to reorganize.

ACTION: No action is required

WH25003. RENUMBER COULD NOT BE COMPLETED BECAUSE X-DOC RECORD WAS MISSING

While attempting to perform a text record reorganization, the function was halted because a cross document record is missing.

ACTION: Manually add the cross document record.

WH25004. RENUMBER COULD NOT BE COMPLETED BECAUSE TEXT RECORD WAS MISSING

While attempting to perform a text record reorganization, the function was halted because a cross document record pointed to a text record that no longer exists.

ACTION: Manually delete or correct the bogus cross document record.

WH25005. RENUMBER COULD NOT BE COMPLETED BECAUSE TEXT RECORD WAS DUPLICATED

During the processing of a RENUMBER command, a text record could not be added because the generated record number already exists. The records are partially renumbered at this point.

ACTION: This is probably a system error. The situation should not occur because unique record numbers are calculated during the renumber operation. Notify Unicom Systems technical support.

JUGLINIT MESSAGES "WI" PREFIX

Messages produced by JUGLINIT (the JNIT transaction) are prefixed by "WI" followed by a 5-position message number, then the message text. The full message will be listed, followed by an explanation and appropriate action to take.

WI01005. PROGRAM "xxxxxxxx" IS NOT AVAILABLE.

CICS-JUGGLER could not find the program xxxxxxxx. Either the PPT entry is disabled or not present, or the program is not in a CICS load file.

ACTION: Ensure that the program is available to CICS-JUGGLER.

WI02001. THIS TERMINAL IS NOT USING CICS-JUGGLER.

While attempting to display the CICS-JUGGLER profile of the user of this terminal, the function was not completed because this terminal is not using CICS-JUGGLER.

ACTION: Press PF3 to exit.

WI02002. THIS PROFILE IS PROTECTED. WINDOW CONFIGURATION CANNOT BE SAVED.

From the User Configuration Display, an attempt to save changes made to the configuration has failed because the corresponding Profile is protected.

ACTION: The changes cannot be saved from this screen. If the changes are to be permanent, the User Profile must be altered.

WI02003. PROFILE HAS BEEN DELETED FROM OPTION TABLE. IT CANNOT BE SAVED.

From the User Configuration Display, an attempt to save changes made to the configuration has failed because the corresponding Profile no longer exists.

ACTION: The profile must be re added in order to save the changes.

WI02004. WINDOW CONFIGURATION HAS BEEN SAVED.

This message indicates that the attempt to save changes made to the configuration has successfully completed.

ACTION: No action is required.

WI04002. "xxxxxxxx" IS NOT A VALID JUGGLER COMMAND.

While editing the CICS-JUGGLER PF key assignments, an invalid CICS-JUGGLER command was attempted to be assigned to a PF key.

ACTION: Review the valid CICS-JUGGLER commands, and correct the error.

MENU GENERATION MESSAGES "WM" PREFIX

Messages produced by JUGLMENU are prefixed by "WM" followed by a 5-position message number, then the message text. The full message will be listed, followed by an explanation and appropriate action to take.

WM02001. MENU "xxxxxxxx" HAS NOT BEEN DEFINED.

The attempted function failed because the menu definition could not be located in the file being searched.

ACTION: If multiple menu files are in use, the menu definition may reside in another file. In which case, you should ensure that the file is defined to CICS-JUGGLER. For more information see section 10 - *CUSTOMIZATION*, under *THE FILE TABLE*. If multiple menu files are not in use, then the menu definition does not exist.

WM03002. PROGRAM "xxxxxxxx" IS NOT AVAILABLE.

CICS-JUGGLER could not find the program xxxxxxxx. Either the PPT entry is disabled or not present, or the program is not in a CICS load file.

ACTION: Ensure that the program is available to CICS-JUGGLER.

WM03003. TRANSACTION "xxxx" IS NOT AVAILABLE.

CICS-JUGGLER could not find the transaction xxxx. Either the PCT entry is disabled or not present.

ACTION: Ensure that the PCT entry is available to CICS-JUGGLER.

WM03004. JUGGLER COMMAND "xxxxxxxx" COULD NOT BE EXECUTED.

The selected menu option has a 'Window Command' associated with it. The command was not executed because the command is syntactically incorrect or an invalid function at this point in windowing operation.

ACTION: Correct the Window Command in the Menu Definition.

WM04004. THE FIRST MENU DEFINITION HAS BEEN DISPLAYED.

While browsing backward through the menu definitions, the first definition on file has been displayed.

ACTION: To browse through more entries you must browse in a forward direction.

WM04005. THE LAST MENU DEFINITION HAS BEEN DISPLAYED.

While browsing forward through the menu definitions, the last definition on file has been displayed.

ACTION: To browse through more entries you must browse in a backward direction.

WM05006. MENU "xxxxxxxx" HAS BEEN DELETED.

This message merely confirms that menu definition xxxxxxxx was successfully deleted from the file.

ACTION: No action is required.

WM05007. KEY DATA AND PRESS "ENTER" TO ADD MENU.

In response to New function (creating a new menu definition), this message is prompting you to key the desired data then press ENTER. The menu will then be added.

ACTION: Key desired data then press ENTER.

WM05008. MENU "xxxxxxxx" HAS BEEN ADDED.

This message merely confirms that the menu definition xxxxxxxx was successfully added to the file.

ACTION: No action is required.

WM05009. MENU "xxxxxxxx" HAS BEEN CHANGED.

This message merely confirms that the menu definition xxxxxxxx was successfully altered.

ACTION: No action is required.

WM05010. MENU "xxxxxxxx" WAS NOT FOUND.

In response to a Find function, CICS-JUGGLER was unable to locate the menu definition xxxxxxxx in the file being searched.

ACTION: If multiple files are being used for CICS-JUGGLER then the definition may exist in another file.

WM05011. MENU ID "xxxxxxxx" ALREADY EXISTS.

A request to add a new menu definition failed because an old definition with the same menu ID already exists.

ACTION: Key a unique menu ID then press ENTER.

WM05012. KEY NEW MENU ID AND PRESS "ENTER" TO COPY MENU.

While copying a menu definition, this message appears to prompt for the menu ID of the new definition.

ACTION: Key the new menu definition ID then press ENTER. The new menu definition will then be added.

WM05013. NO MENU SELECTION AT CURSOR. PRESS "ENTER" TO ADD.

While in the Menu Editor, the SELECT key was pressed. There was not a cursor selectable menu selection at the position of the cursor when the SELECT key was pressed.

ACTION: If you wish to create a cursor selectable menu selection at the position of the cursor, key all desired fields in the select window, then press ENTER.

JUGLMRO MESSAGES "WMR" PREFIX

Messages produced by the JUGLMRO program, which runs in a remote region in an MRO/ISC environment are prefixed by "WMR".

WMR0100. JUGLMRO SUCCESSFULLY ACTIVATED

This message indicates that JUGLMRO is successfully installed and ready for operation.

ACTION: No action is required, you can begin operation of CICS-JUGGLER now.

WMR0200. NO PSEUDO IDS FOUND IN AUTOTBL, JUGLMRO IS NOT NEEDED

This message appears on the system console during CICS startup for a remote region CICS. It indicates that one of the following conditions exists:

1). No User Option file (JUGLFIL) was found in the Core-image/Load library.

2). The User Option Table does not contain any Auto-Init table statements.

3). All Auto-Init table statements have PSEUDO IDS=NO coded.

ACTION: If Pseudo Terminal IDs are not in use, there is no need to activate the JUGLMRO program in the remote region. If they are in use, you must have an Auto-Start table present in the remote region which defines the Pseudo IDs to be used. See *RUNNING CICS-JUGGLER IN AN MRO/ISC ENVIRONMENT* in section 12 - *UNIQUE ENVIRONMENTS AND SPECIAL SITUATIONS*, for more information.

No, action is needed at this point. CICS will be properly initialized and you may use CICS-JUGGLER, but Pseudo Terminal IDs will not work properly for any transactions in this remote region.

TEXT EDITOR MESSAGES "WN" PREFIX

Messages produced by the Menu or Message text editor are prefixed by "WN" followed by a 5-position message number, then the message text.

WN19001. TEXT HAS EXCEEDED MAXIMUM RECORD LENGTH

The menu or message is to large.

ACTION: If this is a message and you have multiple consecutive blank lines, try using the SPACE command. For more information, see section 08, under *THE MENU TEXT EDITOR.*

WN19002. AN ERROR HAS BEEN ENCOUNTERED IN THE TEXT COMMAND AREA.

An invalid command has been issued.

ACTION: Review the appropriate command in section 08, under *THE MENU TEXT EDITOR.*

WN19001. INSERT WOULD EXCEED TWO PAGES. COMMAND IGNORED.

The attempted insert could not be completed because it was too large.

ACTION: Try copying/moving a smaller number of lines.

WN19001. ENDING BLOCK COMMAND NOT ENTERED FOR "xx".

A block command is in progress but has not been performed because the block command does not have a matching command.

ACTION: Review the appropriate command in section 08, under *THE MENU TEXT EDITOR.*

CICS-JUGGLER MESSAGES "JUGL" PREFIX

Messages produced by JUGGLER are prefixed by "JUGL" followed by a 4-position message number, then the message text.

For this document, the full message will be listed, followed by an explanation of appropriate action to take.

JUGL0001. ERROR RETURN FROM TEMP-STORAGE IO, CODE=xx.

A non-zero code was returned by CICS from a Temporary Storage data access. This problem can occur if a CICS task running under **CICS-JUGGLER** causes a storage violation, or if auxiliary Temporary Storage is in use and the control interval size of the Temporary Storage dataset is smaller than the largest terminal screen size plus 500 bytes.

ACTION: If problem persists, notify Unicom Systems.

JUGL0002. xxxx CAN'T BE STARTED WHILE yyyy IS ACTIVE IN SESSION z

This message can occur if the Single Occurring Transaction Table is installed and one of the transactions in the table is attempted to start in more than one virtual terminal. It can also occur if the transaction designated by yyyy in the message is defined as an 'excluded' transaction in the User Option Table (For more information, please refer to *SINGLE OCCURRING TRANSACTIONS TABLE*, *SINGLE OCCURRING PROGRAMS TABLE*, or *TRANSACTION EXCLUSION TABLE* in section 10 - *CUSTOMIZATION*).

In the message,

xxxx	=	The transaction code entered.
yyyy	=	The transaction or program name in conflict in the other session.
z	=	The virtual terminal number (session number) where the conflicting application is running.

ACTION: You can not start this transaction without first terminating the conflicting transaction in the other virtual terminal.

JUGL0003. ACTIVE TASK xxxx ON LOGICAL TERMINAL t MUST BE ENDED

This message occurs in response to a JUGL,OFF command. It states that you must terminate task xxxx in session t before the JUGL,OFF command may be performed. (This is a customization option, for more information see *FIELDS OF THE USER OPTIONS TABLE* in section 10 - *CUSTOMIZATION).*

ACTION: Toggle to session t and exit the task, then issue the JUGL,OFF command.

JUGL0004. DO YOU WANT TO TERMINATE CICS-JUGGLER ?

The operator has signed off via the CSSF (or some other) transaction. **CICS-JUGGLER** intercepts the sign-off transactions and gives this query. It is intended to serve as a reminder that **CICS-JUGGLER** is still active on this terminal.

ACTION: If you want to terminate **CICS-JUGGLER** on this terminal, press ENTER. (the "Y" response is pre-coded on the screen). Otherwise key an "N" and press ENTER.

JUGL0005. CICS-JUGGLER PASSWORD ERROR, CODE=x

This message indicates that the product has expired or cannot be used because of a password error or installation error. The product is now inactive and cannot be used.

The "x" in the above message will be one of the following error codes:

1 - The current date is greater than the password expiration date.
2 - Module "STSPASS" could not be found in a library in the search string.
3 - The product ID could not be located in STSPASS.
4 - The password in STSPASS is invalid.
5 - Module "STS0100" could not be found in a library in the search string.
6 - STSPASS contains a CPU ID that is invalid for the machine that it is installed on.
7 - The password entered is not valid for the requested product.
9 - An undetermined error has been encountered. Please call Unicom Systems Technical Support at 1-(818)-838-0606.

ACTION: Correct the error if possible, or call Unicom Systems Technical Support at 1-(818)-838-0606.

JUGL0006. RELEASE LEVEL OF CICS IS INCOMPATIBLE WITH CICS-JUGGLER

This message appears if the wrong version of **CICS-JUGGLER** is installed; the CICS 1.6 version on a 1.7 CICS, for instance.

ACTION: Re-install CICS-JUGGLER.

JUGL0007. ffffffff ERROR ON aaaaaa, CODE=bb-cccccccc, VSAM CODE=dd

During a file control access to the VSAM file ffffffff, an error was returned from CICS. In the message,

ffffffff	=	The name of the VSAM file.
aaaaaa	=	The request macro being issued.
bb	=	The return code from CICS.
cccccccc	=	The interpreted CICS code.
dd	=	The VSAM error code if cccccccc is 'ILLOGIC'.

The following action is taken by CICS-JUGGLER on any error return during access to JUGLFIL:

1). The message is displayed on the terminal.
2). The message is displayed on the console.
3). A transaction dump is taken.

ACTION: For an explanation of the CICS return codes, refer to the *CICS APPLICATION PROGRAMMERS REFERENCE MANUAL* under *FILE CONTROL RESPONSE CODES*.

Attempt to correct the problem with the file and continue operation. Or, issue the JUGL,TEMP command to switch to using Temporary Storage.

JUGL0008. FILE 'xxxxxxxx' IS NOT OPEN

The VSAM file xxxxxxxx has been closed to CICS users.

ACTION: The file must be opened.

JUGL0009. CICS-JUGGLER MUST BE TERMINATED TO LOG-OFF

You can not sign off of CICS without first terminating **CICS-JUGGLER**. This is an option specified in the User Option Table.

ACTION: Enter JUGL,OFF to terminate **CICS-JUGGLER**, then do the sign off.

JUGLO011. TRANSACTION xxxx NOT ALLOWED IN WINDOW MODE

You can not issue this transaction code while in window mode. You must be in full-screen mode. The message indicates that this transaction code is defined in the TYPE=WTXNTBL of the User Option Table (see *CUSTOMIZATION*).

This message will also appear if you enter a CSSF (or equivalent sign-off) transaction code while in window mode, since sign-off is not allowed from within a window.

ACTION: Press the 'WINDOW' key or use the "X" command to return to full-screen mode and then enter the transaction.

JUGL0012. JUGGLER CAN NOT BE INITIALIZED, WINDOW HAS BEEN USED.

The CICS-JUGGLER product and the CICS-JUGGLER product can not run together on the same system. If CICS-JUGGLER is used before a JUGL transaction is entered, this message will result. If a JUGL transaction is performed first, the WNDO transaction is automatically disabled.

ACTION: You should remove the WNDO transaction from the CICS-PCT and bring CICS down and up.

JUGL0013. USER EXIT INTERFACE NOT PRESENT, NEED EXITS = YES IN DFHSIT.

In attempting to enable the After-Input TCP/ZCP user exit to CICS during CICS-JUGGLER initialization, it was determined that the CICS User Exit Interface is not enabled.

ACTION: You must specify EXITS=YES in the System Initialization Table or as a SIT override parameter in the CICS start-up JCL, then bring CICS down and up.

JUGL0014. ERROR HAS OCCURRED DURING EXIT ENABLE, EIBRCODE = X'nnnn'

In attempting to enable the After-Input TCP/ZCP user exit to CICS during CICS-JUGGLER initialization an error was returned by CICS from the ENABLE command. X'nnnn' is the 2-byte hexadecimal code posted in the EIBRCODE field by CICS. See the *CICS CUSTOMIZATION GUIDE* under *ENABLING A GLOBAL USER EXIT* for an explanation of these codes.

JUGL0015. DFHPCP MODULE IS IN SVA/LPA OR ALTERED BY ANOTHER PRODUCT

The DFHPCPxx module can not reside in the SVA or LPA. See *CICS REQUIREMENTS* in section 09 - *INSTALLATION,* for further explanation.

ACTION: Remove the DFHPCP module from the DOS SVA or MVS LPA using the procedure described in *CICS-REQUIREMENTS* and bring CICS down and up.

JUGL0016. GUARDIAN INCOMPATIBILITY, CALL UNICOM SYSTEMS FOR PTF.

This message indicates that the release of GUARDIAN that you are using is incompatible with CICS-JUGGLER.

ACTION: Notify Unicom Systems Technical Support (see Appendix C).

JUGL0017. xxxxxxxx RECORD SIZE IS TOO SMALL

The record size (RECSZ) operand in the IDCAMS DEFINE for the VSAM file xxxxxxxx must be at least 5100.

ACTION: Redefine the VSAM file with the proper record size, then bring CICS down and back up.

JUGL0018. CICS-JUGGLER CURRENTLY UNAVAILABLE

A JUGL,STOP command has been issued to stop all users from doing any JUGL transactions until some sort of system activity has been completed. No JUGL transactions will be accepted until a JUGL,START command is issued.

Note that if you are trying to do a JUGL,START command after doing a JUGL,BACKOUT command, you must do a CEMT NEWCOPY command on the JUGGLER and JUGLMAIN modules.

JUGL0019. JUGL,BACKOUT MUST BE DONE TO START JUGL

The CICS-JUGGLER package has been used before any JUGL transaction was issued. The two packages can not both run at the same time. If you want to activate JUGGLER, you must do a WNDO,BACKOUT command.

ACTION: Make all CICS-JUGGLER users do a WNDO,OFF (or purge all users), then do a WNDO,BACKOUT command, then do any JUGL transaction.

JUGL0020. MULTIPLE INTERACTIVE INTERFACE SESSIONS NOT ALLOWED

This message occurs when, after toggling into a virtual terminal other than session number 1, the operator pressed PF3 to return to the Interactive Interface selection panel and multiple Interactive Interface support has not been specified for this terminal (VSE INTERACTIVE INTERFACE=NO in the User Options Table). The operator must start applications using transaction codes in all sessions except number 1.

ACTION: If you desire to support multiple Interactive Interface sessions for this terminal, refer to *RUNNING CICS-JUGGLER WITH THE DOS VSE INTERACTIVE INTERFACE* in section 12 - *UNIQUE ENVIRONMENTS AND SPECIAL SITUATIONS* for the coding required in the User Option Table to support this feature.

JUGL0021. CANNOT ACTIVATE, UNABLE TO CONSTRUCT JUGLTBL

During CICS-JUGGLER activation, CICS-JUGGLER was not able to build an internal image of the customization table and was unable to activate.

ACTION: This message should not occur, however if it does, perform a NEWCOPY of all CICS-JUGGLER programs, then try to activate CICS-JUGGLER. If the problem still occurs, call Unicom Systems Technical Support.

JUGL0022. xxxx CAN'T BE STARTED WITH CICS-JUGGLER

The transaction code represented by xxxx in the message is present in the TransacTion Stop table in the User option table, which means that this transaction can not be issued on this terminal if CICS-JUGGLER is active.

ACTION: You must do a JUGL,OFF command in order to use this transaction code.

JUGL0023. YOU MUST END CONVERSATIONAL MODE, PRESS ENTER TO RESTORE

An attempt to toggle to another session has failed because you were trying to toggle out of a conversation task.

ACTION: You must end conversational mode before toggling.

JUGL0024. JUGL,BACKOUT MUST BE DONE TO USE xxxx

The transaction code represented by xxxx in the message is present in the Transaction Backout table. This means that this transaction can not be issued on any terminal if CICS-JUGGLER is active on any terminal.

ACTION: You must do a JUGL,BACKOUT command in order to use this transaction code.

JUGL0025. JUGLMAIN PROGRAM NOT FOUND, CANNOT ACTIVATE

CICS-JUGGLER could not find the JUGLMAIN program. Either the PPT entry is disabled or not present, or JUGLMAIN is not in a CICS load file.

ACTION: Ensure that the JUGLMAIN program is available to CICS-JUGGLER.

JUGL0026. FILE 'xxxxxxxx' STRNO IS LESS THAN 3, CANNOT USE

This message can occur at CICS-JUGGLER initialization if the VSAM file in use has an inadequate number of strings (STRNO) defined. The STRNO value in the FCT entry must be at least '3'. CICS-JUGGLER is unable to use the file as defined, and therefore is unable to activate.

ACTION: In order to use this VSAM file, the STRNO value in the FCT must be increased to 3 or greater. You can not do any JUGL command until this is corrected.

JUGL0027. FILE 'xxxxxxxx' RECFORM MUST BE VARIABLE, CANNOT USE

This message can occur at CICS-JUGGLER initialization if the VSAM file in use has an incorrect record format (RECFORM) defined. The RECFORM value in the FCT entry must be 'VARIABLE'. CICS-JUGGLER is unable to use the file as defined, and therefore is unable to activate.

ACTION: In order to use this VSAM file, the RECFORM parameter in the FCT must be changed to 'VARIABLE'. You can not do any JUGL command until this is corrected.

JUGL0028. PROGRAM 'JUGLENAB' NOT IN PPT OR NOT IN CICS LOAD FILE

CICS-JUGGLER could not find the JUGLENAB program. Either the PPT entry is disabled or not present, or JUGLENAB is not in a CICS load file.

ACTION: Ensure that the JUGLENAB program is available to CICS-JUGGLER.

JUGL0029. YOU MUST SUCCESSFULLY PERFORM A SIGNON BEFORE PROCEEDING

The action that you have selected requires that you be signed-on.

ACTION: Sign on.

JUGL0030. PROGRAM JUGLAUXL HAS NOT BEEN INSTALLED, UNABLE TO INITIALIZE

CICS-JUGGLER could not find the program "JUGLAUXL". Either the PPT entry is disabled or not present, or the program is not in a CICS load file.

ACTION: Ensure that the program is available to CICS-JUGGLER.

JUGL0031. TRANSACTION xxxx MUST HAVE A TWASIZE OF AT LEAST 32 BYTES

The PCT entry for the transaction indicated by xxxx must have a TWA size of at least 32 bytes.

ACTION: Using RDO or the DFHPCT macro, increase the TWA size.

JUGL0052. CANNOT START INTERACTIVE INTERFACE SESSION
JUGL0053. CANNOT START INTERACTIVE INTERFACE SESSION
JUGL0054. CANNOT START INTERACTIVE INTERFACE SESSION
JUGL0055. CANNOT START INTERACTIVE INTERFACE SESSION
JUGL0056. CANNOT START INTERACTIVE INTERFACE SESSION
JUGL0057. CANNOT START INTERACTIVE INTERFACE SESSION
JUGL0058. CANNOT START INTERACTIVE INTERFACE SESSION

All seven of the above listed messages contain the same message text. The message number defines the condition which was encountered.

These messages originated in the JUGLVSP program which controls the method of operation when using multiple Interactive Interface selection panels in a DOS VSE environment. In all cases, the message is produced for terminals with VSE INTERACTIVE INTERFACE=YES or VSE INTERACTIVE INTER-FACE=SIGNON in the User Option Table when a JUGL,ON command is issued and it is not possible to start multiple Interactive Interface sessions for some reason.

Following is an explanation of the condition that was encountered according to the message number:

JUGL0052	-	Unable to locate the IESO transaction work area.
JUGL0053	-	Unable to locate the system vector.
JUGL0054	-	The terminal is not present in the SP anchor table.
JUGL0055	-	No user status record present.
JUGL0056	-	Unrecognized user status record.
JUGL0057	-	Unable to locate panel hierarchy.
JUGL0058	-	Unable to locate the IESO task.

ACTION: This condition could occur if you are not running DOS VSE but have installed the JUGLVSP program. JUGLVSP only pertains to VSE systems. If this is the case, remove JUGLVSP from the system.

It could also occur if you have activated CICS-JUGGLER using a different Pseudo Terminal ID than that which is coded in the Auto-Init table entry for this terminal. If you used JUGL,INIT, for example and entered a different Pseudo ID or entered no Pseudo IDs.

If either of the above are true, the error will not impact CICS-JUGGLER operation but you will not be able to start multiple Interactive Interface sessions on this terminal.

If neither of the above is true, call Unicom Systems Technical Support and report the message number to a technical support representative.

JUGL0059. JUGLVSP SUCCESSFULLY ACTIVATED

This message is displayed on the system console in a DOS VSE environment at CICS start-up if the JUGLVSP program is installed. It indicates that support for multiple Interactive Interface sessions has been successfully established.

ACTION: No action is required.

JUGL0060. USER OPTION TABLE MUST BE PRESENT TO ACTIVATE JUGLVSP

This message is displayed on the system console in a DOS VSE environment at CICS start-up if the JUGLVSP program is installed. It indicates that the User Option Table, JUGLTBL, could not be found.

ACTION: If you intend to support multiple Interactive Interface sessions, refer to *RUNNING CICS-JUGGLER WITH THE DOS VSE INTERACTIVE INTERFACE* in section 12 - *UNIQUE ENVIRONMENTS AND SPECIAL SITUATIONS* for the coding required in the User Option Table to support this feature. Otherwise, remove JUGLVSP from the system.

JUGL0061. MULTIPLE INTERACTIVE INTERFACE SUPPORT IS NOT ACTIVE

This message is displayed on the system console in a DOS VSE environment at CICS start-up if the JUGLVSP program is installed. It indicates that support for multiple Interactive Interface sessions could not be established for one of the following reasons:

1). All terminals in the User Option Table specify VSE INTERACTIVE INTERFACE=NO.

2). No Auto-Init table entries are present.

ACTION: If you intend to support multiple Interactive Interface sessions, refer to *RUNNING CICS-JUGGLER WITH THE DOS VSE INTERACTIVE INTERFACE* in section 12 - *UNIQUE ENVIRONMENTS AND SPECIAL SITUATIONS* for the coding required in the User Option Table to support this feature. Otherwise, remove JUGLVSP from the system.

JUGL0062 .NON-EXISTENT TERMINAL IN AUTOTBL, ID=xxxx

This message is displayed on the system console in a DOS VSE environment at CICS start-up if the JUGLVSP program is installed. It indicates that an Auto-Init table entry was found which specifies a terminal ID which is not defined to the CICS system. In the message, xxxx is the terminal ID in question.

ACTION: No action is required, the invalid terminal ID will be ignored.

JUGL0063. CORRUPTED CHAIN DETECTED, CD=xx, SNAP DUMP TAKEN

This is a should-not-occur message, indicating that CICS-JUGGLER has detected a storage violation in its internal control blocks.

ACTION: Contact Unicom Systems Technical Support 1-(818)-838-0606. He will probably ask you to print the dump and send it to him.

JUGL1201. COPY FUNCTION COMPLETED

In response to a copy function, this message appears to confirm that the copy function was completed successfully.

ACTION: No action is required.

JUGL1202. NOTHING TO PASTE

In response to a paste function, CICS-JUGGLER was unable to perform the paste because no previous copy function was completed.

ACTION: You must first perform a copy function.

JUGL1401. MAXIMUM NUMBER OF TERMINALS HAS BEEN REACHED.

This message can occur if more terminals are using **CICS-JUGGLER** then the available dynamic storage will allow or if the User Option table is coded to limit the number of active users and that limit has been reached.

ACTION: Terminate **CICS-JUGGLER** on some other terminal and try again.

JUGL1402. xxxx IS ALREADY IN USE.

This message appears when attempting to activate Pseudo Terminals and the terminal ID which was entered is either a real physical terminal ID known to CICS or is a Pseudo Terminal ID in use on this or some other terminal.

ACTION: Enter another code for the Pseudo Terminal ID. Note that you may not duplicate codes.

JUGL1403. THIS TERMINAL CAN NOT USE CICS-JUGGLER

This terminal has been excluded from using **CICS-JUGGLER** in the User Option Table.

ACTION: Notify your supervisor or use another terminal.

JUGL1404. THIS OPERATOR CAN NOT USE CICS-JUGGLER

The 3-character Operator ID assigned to this operator sign-on has been excluded from using **CICS-JUGGLER** in the User Option Table.

ACTION: Notify your supervisor or use another operator sign-on.

JUGL1405. AUTO-START TABLE REQUIRED BUT NOT PRESENT

AUTOTBL=FORCE has been specified in the User Option Table but no valid TYPE=AUTOTBL statement could be found.

ACTION: Correct and re-assemble the User Option Table and retry.

JUGL1406. NUMBER OF TERMINALS MUST BE FROM 2 TO x.

During initialization of **CICS-JUGGLER** at a specific physical terminal, in response to the query "SELECT NUMBER OF TERMINALS", the operator entered a number outside the range of 2 to x (where x is the number coded for MAX LOGICAL TERMINALS in the User Options Table).

ACTION: Enter the number again.

JUGL1407. ENTER IS AN INVALID TOGGLE/WINDOW KEY.

In response to the query "SELECT TOGGLE KEY", the operator pressed the ENTER key instead of a Program Function (PF) or Program Attention (PA) key.

ACTION: Press any PF or PA key.

JUGL1408. ERROR RETURN FROM TEMP-STORAGE IO, CODE=xx.

A non-zero code was returned by CICS from a Temporary Storage data access. This problem can occur if a CICS task running under **CICS-JUGGLER** causes a storage violation, or if auxiliary Temporary Storage is in use and the control interval size of the Temporary Storage dataset is smaller than the largest terminal screen size plus 500 bytes.

ACTION: If problem persists, notify Unicom Systems.

JUGL1409. ACTIVE TASK xxxx ON LOGICAL TERMINAL n MUST BE ENDED.

The operator has attempted to terminate **CICS-JUGGLER** while a conversational task is waiting for a response in another logical terminal. xxxx is the task ID, n is the logical terminal number. If the "Clear all terminals before logoff" option is specified in the user option table, the message will appear if any logical terminal contains a screen display at the time JUGL,OFF is entered.

ACTION: return to the indicated logical terminal (n) and respond to the task waiting to be ended. Then enter JUGL,OFF again to terminate **CICS-JUGGLER**.

JUGL1410. PF/PA KEY IS CURRENT TOGGLE KEY
JUGL1410. PF/PA KEY IS CURRENT WINDOW KEY

In response to the query "SELECT WINDOW KEY", the operator pressed the same PF or PA key used for the "toggle" key.

ACTION: Select another PF or PA key.

JUGL1411. CICS-JUGGLER IS CURRENTLY UNAVAILABLE

A JUGL,STOP command has been issued to prevent anyone from using CICS-JUGGLER. The JUGL,ON or JUGL,INIT command can not be performed until a JUGL,START is issued.

ACTION: Wait until CICS-JUGGLER has been made available and try again.

JUGL1412. ID xxxx FOR TERMINAL yyyy ALREADY IN USE

During CICS-JUGGLER initialization when using the PLT Start-up option, the Pseudo Terminal ID indicated by xxxx in the message is defined in the Auto-Start Table for terminal yyyy. Another terminal in the Auto-Start table has defined the same Pseudo Terminal ID.

ACTION: CICS-JUGGLER will not be started on this terminal. Initialization will continue with the next terminal in the Auto-Start table.

JUGL1413. ERROR IN PROCESSING AUTO START ENTRY FOR TERMINAL xxxx

During CICS-JUGGLER initialization when using the PLT Start-up option, a coding error has been found in the Auto-Start Table for terminal xxxx. The previous error message on the console indicates the error condition.

ACTION: CICS-JUGGLER will not be started on this terminal. Initialization will continue with the next terminal in the Auto-Start table.

JUGL1414. ** CICS-JUGGLER WILL EXPIRE IN xx DAYS **

The trial period for CICS-JUGGLER is about to expire. The xx in the message indicates the number of days until it will be automatically deactivated.

ACTION: Notify the MIS System Programmer or call Unicom Systems.

JUGL1415. AUTOTBL KEY CONFLICTS WITH PRINT KEY - xxxx

The Toggle-Forward, Toggle-Backward or Direct Session key in the User Profile designated by xxxx in the message is the same key defined to CICS as the terminal print key (for printing terminal screens on an on-line printer).

This is an informational message indicating that the designated key could not be activated due to the conflict.

ACTION: Change the key in the Auto-Start table to another PF or PA key, or change the SITPRINT operand in the CICS System Initialization Table to specify a different PRINT key.

JUGL1416. xxxxxx KEY CONFLICTS WITH SYSTEM PRINT KEY

The PF or PA key pressed for the TOGGLE key or the WINDOW key is the same key defined to CICS as the terminal print key (for printing terminal screens on an on-line printer). This key can not be used for the TOGGLE key.

ACTION: Select another PF/PA key, or change the SITPRINT operand in the CICS System Initialization Table to specify a different PRINT key.

JUGL1417. USER NOT AUTHORIZED FOR THIS COMMAND

If any of the following are coded YES on the User Option table, then the command(s) cannot be entered with the JUGL transaction code:

BACKOUT, CPROFF, CPRON, DEBUG, INIT, PURGE, START, STOP, SYS, TEMP, MAIN and WIN=

You may use the command(s) with the JUSC transaction code if you are authorized for the transaction.

ACTION: Try the command using JUSC. If CICS will not let you issue that transaction, you cannot issue this command.

JUGL1418. INVALID WINDOW COMMAND

The transaction command following the JUGL transaction code is not a valid CICS-JUGGLER command.

ACTION: Refer to Appendix A for a list of valid transaction commands.

JUGL1419. CICS-JUGGLER xxxxxx, TERMINAL yyyy, USER zzz, TIME hh:mm:ss

This message appears in the CICS statistics log each time a JUGL command is issued at any terminal. MESSAGE LOG=xxxx (where xxx is the destination ID) must be specified in the User Option Table to obtain these messages.

In the message, xxxxxx is the command (ON, OFF, PURGE, etc.), yyyy is the terminal ID where the command was given, zzz is the Operator ID at that terminal and hh:mm:ss is the current time of day.

ACTION: No action is required, the message log is for informational purposes only.

JUGL1420. "WIN=" COMMAND NOT VALID FOR POPUP JUGGLER.

The "WIN=" command may only be used for Standard or Variable window modes.

ACTION: Popup window mode allows you to configure the windows however you wish.

JUGL1421. "xxxx" IS NOT A VALID PROFILE ID.

In response to the JUGL,ON,xxxx command, the command was not able to complete properly because the profile xxxx does not exist.

ACTION: Re-enter the command with a valid profile ID.

JUGL1501. ALL USERS MUST JUGL, OFF TO ISSUE THIS COMMAND.

In order to issue any of the following special purpose transaction commands there must not be any active terminal using CICS-JUGGLER:

```
JUGL,TEMP
JUGL,MAIN
JUGL,BACKOUT
```

ACTION: Make all terminal operations issue a JUGL,OFF command. You may use the JUGL,STOP as command to prevent anyone from doing and the JUGL,ON on and/or you may use the JUGL,PURGE or PURGE,ALL commands to force users off. (See section 13 - *SPECIAL-PURPOSE COMMANDS* for further information).

JUGL1502. VSAM FILE 'xxxxxxxx' NOT FOUND IN FCT.

The JUGL,VSAM command has been issued and JUGLFIL has not been defined in the File Control Table.

ACTION: Define the JUGLFIL FCT entry as described in the installation process. If Unicom System's product CICS-FCTD is present you may use it to dynamically define the file without recycling CICS.

JUGL1503. CICS-JUGGLER NOW USING TEMP-STORAGE
JUGL1503. CICS-JUGGLER NOW USING MAIN-STORAGE

This is a confirmation message in response to the JUGL,TEMP or JUGL,MAIN command indicating successful completion.

ACTION: Proceed with any CICS activity.

JUGL1504. TERMINAL xxxx CAN NOT BE LOCATED.

In response to the JUGL,PURGE, xxxx COMMAND. The terminal ID (real or pseudo) designated by xxxx could not be found.

ACTION: Re-enter the command with a valid terminal ID. This may be a real ID in the TCT or a pseudo ID of an active CICS-JUGGLER user.

JUGL1505. PURGE FOR TERMINAL xxxx SUCCESSFUL.

This is a confirmation message in response to the JUGL,PURGE command indicating successful completion. The designated terminal has been removed from the active users of CICS-JUGGLER. This is equivalent to the operator of that terminal doing a JUGL,OFF command.

ACTION: Proceed with any CICS activity.

JUGL1506. TERMINAL xxxx IS NOT USING JUGGLER.

In response to the JUGL,PURGE, xxxx command, the terminal designated by xxxx is not an active CICS-JUGGLER user.

ACTION: There is no need to purge this user. Proceed with any CICS activity.

JUGL1507. CICS-JUGGLER SUCCESSFULLY QUIESCED.

In response to the JUGL,STOP command, this message indicates that no one will be allowed to activate CICS-JUGGLER on their terminal until a JUGL,START command is issued. Users that are already activated will not be affected.

ACTION: Proceed with whatever system action required the JUGL,STOP command, then issue a JUGL,START command.

JUGL1508. CICS-JUGGLER MAY NOW BE NEW-COPIED.

This is a confirmation message in response to the JUGL,BACKOUT command. CICS-JUGGLER has removed its intercept points in CICS and removed the "in-use" condition.

ACTION: You may now issue a CEMT NEWCOPY command for the module "JUGGLER" or "JUGLTBL" or both, if desired. Note that if you previously issued a JUGL,STOP command, you must perform a NEWCOPY on the JUGGLER module. Message JUGL0040 will result when attempting to perform a JUGL,START command after doing the BACKOUT command.

JUGL1509. CICS-JUGGLER SUCCESSFULLY RESTARTED

In response to the JUGL,START command, this message indicates that JUGL,ON or JUGL,INIT commands will now be honored.

ACTION: Proceed with any CICS activity.

JUGL1510. CICS-JUGGLER IS CURRENTLY UNAVAILABLE

A JUGL,STOP command has been issued to prevent anyone from using CICS-JUGGLER. The JUGL,ON or JUGL,INIT command can not be performed until a JUGL,START is issued.

ACTION: Wait until CICS-JUGGLER has been made available and try again.

JUGL1511. PURGE FOR ALL TERMINALS SUCCESSFUL

This is a confirmation message in response to the JUGL,PURGE,ALL command indicating successful completion. All terminals have been removed from the active users of CICS-JUGGLER.

ACTION: Proceed with any CICS activity.

JUGL1512. DEBUG FOR TERMINAL xxxx NOW ON
JUGL1512. DEBUG FOR TERMINAL xxxx NOW OFF

This a confirmation message in response to the JUGL,DEBUG command indicating successful completion of the ON or OFF specification for the designated terminal.

ACTION: If you just turned DEBUG on, CICS-JUGGLER will begin writing transaction dumps to the Dump Dataset for every JUGL transaction until a DEBUG OFF command is issued. This will fill up the Dump Dataset if you leave it on very long.

JUGL1513. ALL SESSIONS HAVE BEEN CLEARED

This message appears any time the CLEAR key is depressed in full-screen mode if CLEAR ALL SESSIONS ON CLEAR has been selected in the User Option Table. It indicates that the transactions in all virtual terminals for this physical terminal have been terminated and their screens erased.

ACTION: Continue with any CICS activity. Note that you cannot get rid of this message by pressing CLEAR again. You may start another transaction by entering the transaction code and pressing ERASE EOF before pressing ENTER.

JUGL1514. UNABLE TO PURGE xxxx, ICCF IS ACTIVE

This message can occur when a JUGL,PURGE command is issued, either for a specific terminal or for all terminals. In the message, xxxx will be the specific terminal ID or the work 'ALL'.

The PURGE could not be accomplished because ICCF is active in a virtual terminal other than the current session at that terminal. CICS-JUGGLER will not purge a terminal if ICCF is active.

ACTION: If the PURGE was for ALL terminals, press ENTER at this point and you will be presented with the JUGL,SYS display. The first terminal on the display is the terminal with ICCF active. All other terminals down to that one have been purged.

You can follow the terminal ID or the word 'ALL' in the PURGE command with the word FORCE, if desired. This will go ahead and purge the terminal even though ICCF is active. In DOS VSE systems, this causes an outstanding ICCF session which can not be restarted. You will not be able to log on to ICCF with that Operator ID until CICS is recycled. For non-SP systems, the next time ICCF is entered on that terminal you will receive an ICCF forced-logoff message.

If you do not want to FORCE the PURGE, terminate ICCF on the terminal normally, then re-issue the PURGE command, if needed.

JUGL1515. TASK xxxx PURGED DUE TO JUGL,TIMEOUT

This message occurs when a conversational transaction has been automatically purged by CICS-JUGGLER. Conversational transactions will be purged if the TIME OUT SELECTION TYPE option is specified in the User Option Table and the transaction indicated by xxxx in the message has been inactive for the specified time interval. This message replaces the session display in the virtual terminal where transaction xxxx was initiated, and will be seen when the operator toggles into that virtual terminal.

ACTION: No action is required. You may start the same or any other transaction in this virtual terminal, if desired.

JUGL1517. INVALID JUGGLER COMMAND

The transaction command following the JUGL transaction code is not a valid CICS-JUGGLER command.

ACTION: Refer to Appendix A for a list of valid transaction commands.

JUGL1518. PSEUDO TERMINAL IDS ARE NOW ON
JUGL1518. PSEUDO TERMINAL IDS ARE NOW OFF

This message occurs in response to the JUGL,ALT command. The message indicates that the action was successful and the Pseudo Terminal ID feature of CICS-JUGGLER is now either ON or OFF.

ACTION: No action is required.

JUGL1520. CICS-JUGGLER AUTO-PURGED,

If FORCE PURGE AT SIGNON AND SIGNOFF is specified in the User Option Table, this message will appear any time a sign-on is performed when CICS-JUGGLER is active at the terminal. The message indicates that CICS-JUGGLER has been automatically purged. All sessions have been terminated and any conversational tasks abnormally terminated.

ACTION: No action is required.

JUGL1521. PROGRAM "xxxxxxxx" IS NOT AVAILABLE

CICS-JUGGLER could not find the program xxxxxxxx. Either the PPT entry is disabled or not present, or the program is not in a CICS load file.

ACTION: Ensure that the program is available to CICS-JUGGLER.

JUGL8001. CURSOR IMPROPERLY POSITIONED

When using the VIEW function of the JUGL,SYS display, the cursor was not positioned to the first character of the transaction code in the terminal session to be displayed.

ACTION: Re-position the cursor by using the TAB or NEW LINE key, then press ENTER again.

JUGL8002. INCOMPATIBLE SCREEN SIZE

When using the VIEW function of the JUGL,SYS display, the screen display in the terminal session to be displayed is larger than the screen of your terminal. This indicates that the object session is using alternate screen size on a different model terminal than yours.

ACTION: This terminal session screen can not be viewed on your terminal. You must go to a terminal with the same or larger screen size.

JUGL8003. TERMINAL IS NOT ACTIVE

The terminal session designated by the cursor position on the JUGL,SYS display is not currently an active CICS-JUGGLER session. There are two possible causes for this situation:

1). The cursor was positioned on a virtual terminal number greater than the maximum virtual terminals in use by that physical terminal (selecting session 5 when the terminal only has four virtual terminals, for instance).

2). Between the time that you did the JUGL,SYS command, then positioned the cursor and pressed ENTER, the operator at that terminal did a JUGL,OFF command.

ACTION: Press ENTER to refresh the SYS display. If the terminal still shows on the display, the problem is caused by number 1, above. Re-position the cursor to an active session to try again.

APPENDIX A - SUMMARY OF CICS-JUGGLER COMMANDS

The types of commands in CICS-JUGGLER are:

API commands	(A)pplication (P)rogram (I)nterface commands are issued from a program to integrate transactions with CICS-JUGGLER.
Control Character commands	These are commands that are preceded with the Control Character. These may be entered from any display in CICS, and will be intercepted by CICS-JUGGLER.
Transaction commands	These are preceded by the JUGL transaction code and must be entered in the upper left-hand corner of the screen.
Window Mode commands	These are entered while in window mode, in the command area above the window.

COMMAND	**MODE**	**DESCRIPTION**
BACKOUT	Transaction API	Remove CICS-JUGGLER from the system for new copy.
~C	Control Char.	Copy for cut and paste procedure.
CWINx	API	Clear window number x.
DEBUG,xxxx,OFF	Transaction	Turn off DEBUG mode for terminal xxxx.
DEBUG,xxxx,ON	Transaction	Turn on DEBUG mode for terminal xxxx.
E	Window	Pull down Exit Menu.
~H	Control Char.	Invoke help. Same as pressing the Help key in an unprotected field.
HELP	Transaction	Invoke CICS-JUGGLER Help Index.
INIT	Transaction	Activate **CICS-JUGGLER** on a terminal, bypassing the Auto-Start Table.
INQ	Transaction	Display User Configuration Screen. Same as JUGL with no operands.
~K	Control Char.	Popup the KEYS window, showing all hot key allocations.
~M	Control Char.	Popup the Control Window.
MAIN	Transaction	Switch from using Temporary Storage to using MAIN storage for saving screen data.
~MSG	Control Char.	Receive message. Valid for the CICS-JUGGLER Message Broadcasting Facility only.

OFF	Transaction API	Terminate **CICS-JUGGLER** on this terminal.
ON	Transaction API	Activate **CICS-JUGGLER** on this terminal (using the Auto-Init table).
~P	Control Char.	Paste for cut and paste procedure.
PURGE,ALL	Transaction API	Remove all terminals from using CICS-JUGGLER.
PURGE,xxxx	Transaction API	Remove a terminal xxxx from CICS-JUGGLER use.
~S	Control Char.	Stack (copy the selected data onto the end of the clipboard) for the cut and paste procedure.
~ST	Control Char.	Popup the Control Window to select a session for toggle.
START	Transaction API	Restart CICS-JUGGLER operation after a previously issued STOP command.
STOP	Transaction API	Stop anyone from performing a JUGL,ON or JUGL,INIT command. Active users will not be affected.
SYS	Transaction	Display the System Screen showing all users.
~TB	Control Char.	Toggle backward to the next lower numbered session.
TB	API	Toggle backward to the next lower numbered session.
TEMP	Transaction	Switch from using MAIN storage to Temporary Storage for saving screen data.
~TF	Control Char.	Toggle forward to the next higher numbered session.
TF	API	Toggle forward to the next higher numbered session.
~Tn	Control Char.	Toggle directly to session n.
~UNP	Control Char.	Unprotect all fields on the screen for following cut and paste procedure.

APPENDIX B - SYSTEM REQUIREMENTS

This appendix describes the system environment in which **CICS-JUGGLER** will operate, the program modules involved, the CICS table entries required and the fixed and dynamic storage utilization involved when **CICS-JUGGLER** is active.

DOS/VSE, VSE/SP, VSE/ESA, CICS 1.6 and above
MVS/XA, MVS/ESA, CICS 1.7 and above

Any terminal using IBM 3270 data-stream technology. This includes personal computers accessing the mainframe via the IRMA board or some other interface. The terminal may be in a local or remote environment. BTAM, VTAM and TCAM terminals are supported.

Core-Image/Load modules -
JUGGLER, JUGLENAB, JUGLMAIN, JUGLAUXL, JUGLINIT, JUGLVSP, JUGLMENU, JUGLMRO, JUGLVSP, JUGLMSG, JUGLIVP, JUGLHELP, JUGLREFM, JUGLZNEP STS0100, STSPASS, STSCORE, JUGLHDMO

Source modules -
JUGLTBL, JUGLACF2, JUGLCSSF, JUGLCSSN, JUGLINT3, JUGLINT4, JUGLINT5, JUGLNEPC, JUGLNEPM, JUGLPURG, JUGLRSDD, JUGLRSDM.

PCT - JUGL, JNIT, JAUX, JTMO, JUSC, JMSG, JMNU, JUON, JIVP, JHLP, HDMO, STSC.
PPT - JUGGLER, JUGLMAIN, JUGLINIT, JUGLAUXL, JUGLENAB, JUGLVSP (VSE only), JUGLMSG, JUGLMENU, JUGLIVP, JUGLHELP, JUGLHDMO, STS0100, STSPASS, STSCORE
FCT - VSAM file JUGLFIL.

Resident CICS storage -
JUGGLER - 50K
Shared Storage - approximately 300 bytes per physical terminal using **CICS-JUGGLER**. This storage is allocated above the line at JUGL,ON time and freed at JUGL,OFF time. For MVS users, this storage is taken from the private area above the line, outside of the CICS DSA. For DOS users, partition GETVIS can optionally be used, or it comes from the CICS DSA shared subpool.
Temporary Storage
None if Main File option is in effect. (Screen size plus 90 bytes) times the total number of virtual terminals in the system at any one time. This is the sum of all virtual terminals of every physical terminal which has **CICS-JUGGLER** activated. When **CICS-JUGGLER** is terminated at a physical terminal, the temporary storage for that terminal is released.
VSAM File space
Approximately 40 tracks on a 3380 drive, used for screen control records, on-line customization records and help records. If the Help-windows feature is in use, the amount of VSAM space is a function of the amount of help text created.

APPENDIX C - INSTALLATION DOCUMENTATION

The following is documentation that is printed at the beginning of the installation procedure. It is also included here for reference purposes.

```
***************************************************************************
*                C I C S   R E Q U I R E M E N T S
***************************************************************************
```

Table Entries: PCT - JUGL, JNIT, JAUX, JTMO, JUSC, JMSG, JMNU, JUON, WDMO
JHLP, HDMO, STSC

PPT - JUGGLER, JUGLMAIN, JUGLINIT, JUGLAUXL, JUGLENAB,
JUGLVSP, JUGLMSG, JUGLMENU, JUGLIVP, JUGLHELP,
HELPDEMO, STS0100, STSPASS, STSCORE

FCT - JUGLFIL

STORAGE REQUIREMENTS:

Resident CICS Storage: JUGGLER - 50K
JUGLTBL - APPROX 400 BYTES

Shared Storage Approximately 300 bytes per physical terminal using CICS-JUGGLER. This storage is allocated at JUGL,ON time and freed at JUGL,OFF.

For MVS, this storage is taken from the private area above the line, outside of the CICS DSA.

For DOS, partition GETVIS can optionally be used, or it comes from the CICS DSA shared subpool.

Temp or MAIN Storage: (Screen size plus 90 bytes) times the total number of virtual terminals in the system at any one time. This is the sum of all virtual terminals of every physical terminal which has CICS-JUGGLER activated. When CICS-JUGGLER is terminated at a physical terminal, the temporary storage for that terminal is released.

VSAM File Space: Approximately 40 tracks on a 3380 drive, used for screen control records, on-line customization records, and help records.

If the HELP-WINDOWS feature is in use, the amount of VSAM space is a function of the amount of help text created.

```
****************************************************************************
*  3    I N S T A L L A T I O N   S T E P S
****************************************************************************

****************************************************************************
*  3.1   PREPARATION OF THE INSTALLATION JCL
****************************************************************************
```

Use whatever means are provided by your system to get the punched JCL member from the initial Link/Print/Punch process. This is the output from the JCL keyed from Step 2 in the CICS-JUGGLER reference guide.

DOS users who have a facility such as "GETP" in ICCF for retrieving members from the punch queue can make use of the JCL=Install or JCL=Reinstall parameter.
MVS users can punch the installation JCL into a PDS member by supplying the PDS name, member name, and blocksize of the PDS in Step 2 of the installation guide. If for some reason you cannot or do not want to use the installation JCL from Step 2 of the installation instructions, use the printed output from Step 2 to create the JCL in any fashion desired.

After creating the JCL, by what ever means, modify it to fit your systems specific requirements. The optional entries are noted and filled with question marks. When the modified JCL is ready, submit it along with the necessary tape mounts, modify the CICS tables as instructed and CICS-JUGGLER will be ready for execution.

DOS INSTALLATION JCL

```
****************************************************************************
*   3.2   INSTALLATION JCL LISTING
****************************************************************************

<<STRTPCH DOS>
// JOB STSINST                               PRINT INSTRUCTIONS AND PUNCH JCL
* *****************************************************************
*          STEP 1. -     CREATE JUGGLER INSTALL TAPE
* *****************************************************************
// ASSGN SYS011,???                          INPUT MASTER PRODUCT TAPE
// ASSGN SYS012,???                          (SCRATCH TAPE FOR OUTPUT)
// LIBDEF *,SEARCH=?.?                       (LIBRARY.SUBLIBRARY)              /* NOTE 1 */
// EXEC STSINST
PRODUCT=JUGGLER
MODE=CREATE                                  (CREATE A PRODUCT INSTALL TAPE)
OPSYS=DOS                                    (DOS, MVS, MVS/SP)
CICS=???                                     (160, 170 OR 210 FOR DOS)
LINES=56                                     (CAN BE FROM 1 TO 99)
JCL=??????????                               (INSTALL|REINSTALL)
/*
<STRTDEF>
* *****************************************************************
*          STEP 2. -     DEFINE JUGGLER VSAM FILE
* *****************************************************************
// EXEC IDCAMS,SIZE=AUTO
DEFINE CLUSTER -
   (NAME(CICS.JUGGLER.CONTROL.FILE) -
   RECSZ(2080 8100) -
   CISZ(8192) -
   KEYS(40 0) -
   FSPC(10 10) -
   SHR(2) -
   VOLUME(??????)) -                                                           /* NOTE  3 */
DATA -
   (NAME(CICS.JUGGLER.CONTROL.FILE.DATA) -
   CISZ(8192) -
   TRK(40 5)) -
INDEX -
   (NAME(CICS.JUGGLER.CONTROL.FILE.INDEX) CISZ(256) -
   TRK(1 1))
```

```
/*
* ***************************************************************
*        STEP 3. -     LNKEDT THE JUGGLER PHASES
* ***************************************************************
// ASSGN SYSIPT,???                                                          /* NOTE 4 */
// MTC REW,SYSIPT
// LIBDEF PHASE,CATALOG=?.?                  (LIBRARY.SUBLIB)                /* NOTE 2 */
// OPTION CATAL
  INCLUDE
// EXEC LNKEDT
/*
* ***************************************************************
*        STEP 4. -     LOAD THE JUGGLER SOURCE
* ***************************************************************
// MTC REW,SYSIPT
// MTC FSF,SYSIPT,1
// EXEC LIBR,PARM='A S=?.?'                  (LIBRARY.SUBLIB)                /* NOTE 5 */
/*
// RESET SYSIPT
* ***************************************************************
*        STEP 5. -     INITIALIZE THE JUGGLER VSAM FILE
* ***************************************************************
// ASSGN SYS011,???                                                    /*NOTE 4*/
// MTC REW,SYS011
// MTC FSF,SYS011,2
// DLBL JUGLFIL,'CICS.JUGGLER.CONTROL.FILE',,VSAM                            /* NOTE 6 */
// LIBDEF *,SEARCH=?.?                       (LIBRARY.SUBLIB)                /* NOTE 1 */
// EXEC STSINST
PRODUCT=JUGGLER
OPSYS=DOS
MODE=INSTALL|REINSTALL                                                       /* NOTE 7 */
JUGLFILE=JUGLFIL|BYPASS                                                      /* NOTE 8 */
/*
<TOF.
************************************************************************
*  3.3   JCL NOTE EXPLANATIONS
************************************************************************
```

NOTE 1) This is the sublibrary that STSINST is link-edited into.

2) Depending on your VSE system :

L.S = Library.Sublibrary to contain the phases.
CL = Core image Library to contain the phases.

3) Enter the volume the JUGLFIL is to reside on.

4) SYSIPT should be assigned to the tape drive that the product install tape is mounted. (output from the "Create" run).

5) Depending on your VSE system:

L.S = Library.Sublibrary to contain source modules.
SL = Source library to contain source modules.

6) Should be the cluster name used in the define step. Code cluster name according to user standards.

7) MODE=Install for a new installation of the CICS-JUGGLER Product.
MODE=Reinstall for the re-installation of the CICS-JUGGLER product.

8) JUGLFILE=??????? - Must be the 7-byte VSAM filename.
JUGLFILE=BYPASS - Will bypass the file.

MVS INSTALLATION JCL

```
*************************************************************************
*   3.2   MVS INSTALLATION JCL LISTING
*************************************************************************

//STSINST       JOB 1,UNICOM-SYSTEMS,MSGCLASS=X,CLASS=A
//*
//*             CREATE THE JUGGLER PRODUCT INSTALL TAPE
//*
//STEP1         EXEC    PGM=STSINST
//SYSPRINT      DD      SYSOUT=*
//STEPLIB       DD      DSN=CICS???,DISP=SHR                            /* NOTE 1 */
//MPRDIN        DD      DSN=MASTER.TAPE,UNIT=TAPE,                      /* NOTE 2 */
//                      VOL=SER=MASTER,LABEL=(2,NL),
//                      DCB=BLKSIZE=32000
//MOBJOT        DD      DSN=&&MOBJOT,DISP=(,PASS),                      /* NOTE 3 */
//                      UNIT=SYSDA,SPACE=(CYL,(2,2,0))
//MMACOT        DD      DSN=&&MMACOT,DISP=(,PASS),                      /* NOTE 3 */
//                      UNIT=SYSDA,SPACE=(CYL,(2,2,0))
//MTXTOT        DD      DSN=&&MTXTOT,DISP=(,PASS),                      /* NOTE 3 */
//                      UNIT=SYSDA,SPACE=(CYL,(2,2,0))
//SYSIN         DD      *
PRODUCT=JUGGLER                                 PRODUCT=JUGGLER
MODE=CREATE                                     (CREATE A PRODUCT INSTALL TAPE)
OPSYS=MVS                                       (OPERATING SYSTEM)
CICS=???                                        (170, 210, 211, 212, 311, 321, 330)
LINES=56                                        (CAN BE 1 TO 99)
JCL=?????????           (INSTALL | REINSTALL)
/*
//*
//*             LINK EDIT THE JUGGLER MODULES
//*
//STEP2         EXEC    PGM=IEWL,PARM='LIST,LET,XREF'*
//SYSPRINT      DD      SYSOUT=*
//SYSLIB        DD      DSN=CICS???,DISP=SHR                            /* NOTE 4 */
//SYSLIN        DD      DSN=&&MOBJOT,DISP=(OLD,DELETE)                  /* NOTE 3 */
//SYSUT1        DD      UNIT=SYSDA,SPACE=(1024,(20,20))
//SYSLMOD       DD      DSN=CICS???,DISP=SHR                            /* NOTE 5 */
/*
//*
//*              LOAD THE OPTIONAL SOURCE
//*
//STEP3         EXEC PGM=IEBUPDTE
//SYSPRINT      DD      SYSOUT=*
//SYSUT1        DD      DSN=????????????,DISP=SHR                       /* NOTE 6 */
```

```
//SYSUT2        DD      DSN=????????????,DISP=SHR                               /* NOTE 6 */
//SYSIN         DD      DSN=&&MMACOT,DISP=(OLD,DELETE)                          /* NOTE 3 */
/*
//*
//*             DEFINE THE JUGGLER VSAM FILE
//*
//STEP4         EXEC    PGM=IDCAMS
//SYSPRINT      DD      SYSOUT=*
//SYSIN         DD      *
  DEFINE CLUSTER -
    (NAME(CICS.JUGGLER.CONTROL.FILE) -
     INDEXED -
     KEYS(40 0) -
     RECSZ(2080 8100) -
     SHR(2) -
     FSPC(10 10) -
     VOL(??????) -                                                              /* NOTE 7 */
   DATA -
     (NAME(CICS.JUGGLER.CONTROL.FILE.DATA) -
     CISZ(8192)) -
     TRK(40 5)) -
   INDEX -
    (NAME(CICS.JUGGLER.CONTROL.FILE.INDEX) -
    TRK(1 1))
/*
//*
//*             LOAD THE JUGGLER CONTROL FILE
//*
//STEP5         EXEC    PGM=STSINST
//SYSPRINT      DD      SYSOUT=*
//STEPLIB       DD      DSN=CICS???,DISP=SHR                                    /* NOTE 1 */
//JUGLFIL       DD      DSN=JUGGLER.CONTROL.FILE,DISP=SHR                       /* NOTE 8 */
//MPRDIN        DD      DSN=&&MTXTOT,DISP=(OLD,DELETE)                          /* NOTE 3 */
//SYSIN         DD      *
PRODUCT=JUGGLER                                 PRODUCT=JUGGLER
MODE=INSTALL|REINSTALL                          CREATE JUGL FILE                /* NOTE 9 */
OPSYS=MVS                                       OPERATING SYSTEM
LINES=56                                        CAN BE 1-99
JUGLFILE=JUGLFIL|BYPASS                                                         /* NOTE 10*/
/*
```

CICS TABLE ENTRIES FOR CICS RELEASE 1.7 AND 2.1

```
//*
//*              USE THE FOLLOWING TO DEFINE THE PPT, PCT AND FCT ENTRIES.
//*
//STEP06       EXEC   PGM=DFHCSDUP
//SYSPRINT     DD     SYSOUT=*
//STEPLIB      DD     DSN=CICS???.LOADLIB,DISP=SHR                          /* NOTE 1 */
//DFHCSD       DD     DSN=CICS???.DFHCSD,DISP=SHR,DISP=SHR
//SYSIN        DD     *
DEFINE PROGRAM(STSCORE)
    GROUP(JUGLGRP)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(STSPASS)
    GROUP(JUGLGRP)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(STS0100)
    GROUP(JUGLGRP)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(JUGGLER)
    GROUP(JUGLGRP)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(JUGLAUXL)
    GROUP(JUGLGRP)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(JUGLENAB)
    GROUP(JUGLGRP)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(JUGLINIT)
    GROUP(JUGLGRP)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(JUGLMAIN)
    GROUP(JUGLGRP)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(JUGLHELP)
    GROUP(JUGLGRP)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(JUGLMENU)
    GROUP(JUGLGRP)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(JUGLMSG)
    GROUP(JUGLGRP)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(JUGLIVP)
    GROUP(JUGLGRP)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(JUGLZNEP)
    GROUP(JUGLGRP)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(JUGLHDMO)
    GROUP(JUGLGRP)
    LANGUAGE(ASSEMBLER)
DEFINE TRANSACTION(STSC)
    GROUP(JUGLGRP)
    PROGRAM(STSCORE)
    TWASIZE(00032)
DEFINE TRANSACTION(JAUX)
    GROUP(JUGLGRP)
    PROGRAM(JUGLAUXL)
    TWASIZE(00032)
DEFINE TRANSACTION(JMNU)
    GROUP(JUGLGRP)
    PROGRAM(JUGLMENU)
    TWASIZE(00032)
DEFINE TRANSACTION(JMSG)
    GROUP(JUGLGRP)
    PROGRAM(JUGLMSG)
    TWASIZE(00032)
DEFINE TRANSACTION(JUGL)
    GROUP(JUGLGRP)
    PROGRAM(JUGGLER)
    TWASIZE(00032)
DEFINE TRANSACTION(JNIT)
    GROUP(JUGLGRP)
    PROGRAM(JUGLINIT)
    TWASIZE(00032)
DEFINE TRANSACTION(JUSC)
    GROUP(JUGLGRP)
    PROGRAM(JUGGLER)
    TWASIZE(00032)
DEFINE TRANSACTION(JTMO)
    GROUP(JUGLGRP)
    PROGRAM(JUGGLER)
    TWASIZE(00032)
DEFINE TRANSACTION(JUON)
    GROUP(JUGLGRP)
    PROGRAM(JUGGLER)
    TWASIZE(00032)
DEFINE TRANSACTION(JIVP)
    GROUP(JUGLGRP)
    PROGRAM(JUGLIVP)
    TWASIZE(00032)
DEFINE TRANSACTION(JHLP)
    GROUP(JUGLGRP)
    PROGRAM(JUGLHELP)
    TWASIZE(00032)
```

```
DEFINE TRANSACTION(HDMO)
    GROUP(JUGLGRP)
    PROGRAM(HELPDEMO)
    TWASIZE(00032)
/*

***********************************************************************
*   4.4    CICS-JUGGLER FCT ENTRY - CICS RELEASE 1.7 AND 2.1
***********************************************************************
DFHFCT  TYPE=DATASET,                                     JUGGLER VSAM FILE      X
        DATASET=JUGLFIL,                                  /* NOTE 2 */           X
        DISP=SHR,                                                                X
        ACCMETH=(VSAM),                                                          X
        FILSTAT=(ENABLED,OPENED),                                                X
        RECFORM=(VARIABLE,BLOCKED),                                              X
        SERVREQ=(UPDATE,ADD,DELETE,BROWSE,READ),                                 X
        STRNO=3
```

CICS TABLE ENTRIES FOR CICS RELEASE 3.1

```
//*
//*              RUN THE CSD UTILITY TO DEFINE JUGLGRP FOR CICS 3.1
//*
//STEP6          EXEC PGM=DFHCSDUP
//SYSPRINT       DD      SYSOUT=*
//STEPLIB        DD      DSN=CICS???.LOADLIB,DISP=SHR       /* NOTE 1 */
//DFHCSD         DD      DSN=CICS???.DFHCSD,DISP=SHR
//SYSIN          DD      *
DEFINE FILE(JUGLFIL)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER    CONTROL
FILE)
    DSNAME(CICS.JUGGLER.CONTROL.FILE)
    LSRPOOLID(1)
    DSNSHARING(ALLREQS)
    STRINGS(003)
    STATUS(ENABLED)
    OPENTIME(FIRSTREF)
    DISPOSITION(SHARE)
    DATABUFFERS(00004)
    INDEXBUFFERS(00003)
    RECORDFORMAT(V)
    ADD(YES)
    BROWSE(YES)
    DELETE(YES)
    READ(YES)
    UPDATE(YES)
DEFINE PROGRAM(STSCORE)
    GROUP(JUGLGRP)
    DESCRIPTION(UNICOM              MEMORY
DISPLAY/ALTER)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(STSPASS)
    GROUP(JUGLGRP)
    DESCRIPTION(UNICOM             PRODUCT
PASSWORD TABLE)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(STS0100)
    GROUP(JUGLGRP)
    DESCRIPTION(UNICOM             PRODUCT
PASSWORD CONTROL)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(JUGGLER)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER    RESIDENT
MODULE)
    LANGUAGE(ASSEMBLER)
    RESIDENT(YES)
    CEDF(NO)
DEFINE PROGRAM(JUGLAUXL)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER
CUSTOMIZATION)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(JUGLENAB)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER            EXIT
ENABLER)
    LANGUAGE(ASSEMBLER)

DEFINE PROGRAM(JUGLINIT)
```

```
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER USER
CONFIGURATION)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(JUGLMAIN)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER MAINLINE)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(JUGLMENU)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER MENU
GENERATION)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(JUGLMSG)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER MESSAGE
BROADCAST)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(JUGLIVP)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER INSTALL
VERIFICATION)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(JUGLHELP)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER HELP
PROCESSOR)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(JUGLZNEP)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER DFHZNEP
INTERFACE)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(JUGLHDMO)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER HELP DEMO
DISPLAY)
    LANGUAGE(ASSEMBLER)
DEFINE TRANSACTION(STSC)
    GROUP(JUGLGRP)
    DESCRIPTION(UNICOM MEMORY
DISPLAY/ALTER)
    PROGRAM(STSCORE)
    TWASIZE(00032)
DEFINE TRANSACTION(JAUX)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER
CUSTOMIZATION)
    PROGRAM(JUGLAUXL)
    TWASIZE(00032)
DEFINE TRANSACTION(JMNU)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER MENU
GENERATION)
    PROGRAM(JUGLMENU)
    TWASIZE(00032)
DEFINE TRANSACTION(JMSG)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER MESSAGE
BROADCAST)
    PROGRAM(JUGLMSG)
    TWASIZE(00032)
DEFINE TRANSACTION(JUGL)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER PRIMARY
TRANCODE)
    PROGRAM(JUGGLER)
    TWASIZE(00032)
DEFINE TRANSACTION(JNIT)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER USER
CONFIGURATION)
    PROGRAM(JUGLINIT)
    TWASIZE(00032)
DEFINE TRANSACTION(JUSC)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER SECURED
TRANCODE)
    PROGRAM(JUGGLER)
    TWASIZE(00032)
DEFINE TRANSACTION(JTMO)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER TIMEOUT
TRANCODE)
    PROGRAM(JUGGLER)
    TWASIZE(00032)
DEFINE TRANSACTION(JHLP)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER HELP
DIRECTORY TRANCODE)
    PROGRAM(JUGLHELP)
    TWASIZE(00032)
DEFINE TRANSACTION(JUON)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER AUTO-ON
TRANCODE)
    PROGRAM(JUGGLER)
    TWASIZE(00032)
DEFINE TRANSACTION(JIVP)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER INSTALL
VERIFICATION)
    PROGRAM(JUGLIVP)
    TWASIZE(00032)
DEFINE TRANSACTION(HDMO)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER HELP DEMO
TRANCODE)
```

```
     PROGRAM(HELPDEMO)
     TWASIZE(00032)
/*
```

CICS TABLE ENTRIES FOR CICS RELEASE 3.2

```
//*
//*                RUN THE CSD UTILITY TO DEFINE JUGLGRP FOR CICS 3.2
//*
//STEP6            EXEC   PGM=DFHCSDUP
//SYSPRINT         DD     SYSOUT=*
//STEPLIB          DD     DSN=CICS???.SDFHLOAD,DISP=SHR                          /* NOTE 1 */
//DFHCSD           DD     DSN=CICS???,DISP=SHR
//SYSIN            DD     *
DEFINE FILE(JUGLFIL)
     GROUP(JUGLGRP)
     DESCRIPTION(CICS-JUGGLER CONTROL
FILE)
     DSNAME(CICS.JUGGLER.CONTROL.FILE)
     LSRPOOLID(1)
     DSNSHARING(ALLREQS)
     STRINGS(003)
     STATUS(ENABLED)
     OPENTIME(FIRSTREF)
     DISPOSITION(SHARE)
     DATABUFFERS(00004)
     INDEXBUFFERS(00003)
     RECORDFORMAT(V)
     ADD(YES)
     BROWSE(YES)
     DELETE(YES)
     READ(YES)
     UPDATE(YES)
DEFINE PROGRAM(STSCORE)
     GROUP(JUGLGRP)
     DESCRIPTION(UNICOM MEMORY
DISPLAY/ALTER)
     LANGUAGE(ASSEMBLER)
     DATALOCATION(BELOW)
DEFINE PROGRAM(STSPASS)
     GROUP(JUGLGRP)
     DESCRIPTION(UNICOM PRODUCT
PASSWORD TABLE)
     LANGUAGE(ASSEMBLER)
     DATALOCATION(BELOW)
DEFINE PROGRAM(STS0100)
     GROUP(JUGLGRP)
     DESCRIPTION(UNICOM PRODUCT
PASSWORD CONTROL)
     LANGUAGE(ASSEMBLER)
     DATALOCATION(BELOW)
DEFINE PROGRAM(JUGGLER)
     GROUP(JUGLGRP)
     DESCRIPTION(CICS-JUGGLER RESIDENT
MODULE)
     LANGUAGE(ASSEMBLER)
     DATALOCATION(ANY)
     RESIDENT(YES)
     CEDF(NO)
DEFINE PROGRAM(JUGLAUXL)
     GROUP(JUGLGRP)
     DESCRIPTION(CICS-JUGGLER
CUSTOMIZATION)
     LANGUAGE(ASSEMBLER)
     DATALOCATION(ANY)
DEFINE PROGRAM(JUGLENAB)
     GROUP(JUGLGRP)
     DESCRIPTION(CICS-JUGGLER EXIT
ENABLER)
     LANGUAGE(ASSEMBLER)
     DATALOCATION(ANY)
DEFINE PROGRAM(JUGLINIT)
     GROUP(JUGLGRP)
     DESCRIPTION(CICS-JUGGLER USER
CONFIGURATION)
     LANGUAGE(ASSEMBLER)
     DATALOCATION(ANY)
DEFINE PROGRAM(JUGLMAIN)
     GROUP(JUGLGRP)
     DESCRIPTION(CICS-JUGGLER MAINLINE)
     LANGUAGE(ASSEMBLER)
     DATALOCATION(ANY)
DEFINE PROGRAM(JUGLMENU)
     GROUP(JUGLGRP)
```

```
    DESCRIPTION(CICS-JUGGLER MENU
GENERATION)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
DEFINE PROGRAM(JUGLMSG)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER MESSAGE
BROADCAST)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
DEFINE PROGRAM(JUGLIVP)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER INSTALL
VERIFICATION)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
DEFINE PROGRAM(JUGLHELP)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER HELP
PROCESSOR)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
DEFINE PROGRAM(JUGLZNEP)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER DFHZNEP
INTERFACE)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
DEFINE PROGRAM(HDMO)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER HELP DEMO
DISPLAY)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
DEFINE TRANSACTION(STSC)
    GROUP(JUGLGRP)
    DESCRIPTION(UNICOM MEMORY
DISPLAY/ALTER)
    PROGRAM(STSCORE)
    TWASIZE(00032)
    TASKDATALOC(BELOW)
DEFINE TRANSACTION(JAUX)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER
CUSTOMIZATION)
    PROGRAM(JUGLAUXL)
    TWASIZE(00032)
    TASKDATALOC(ANY)
DEFINE TRANSACTION(JHLP)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER HELP
DIRECTORY TRANCODE)
    PROGRAM(JUGLHELP)
    TWASIZE(00032)
    TASKDATALOC(ANY)
DEFINE TRANSACTION(JMNU)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER MENU
GENERATION)
    PROGRAM(JUGLMENU)
    TWASIZE(00032)
    TASKDATALOC(ANY)
DEFINE TRANSACTION(JMSG)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER MESSAGE
BROADCAST)
    PROGRAM(JUGLMSG)
    TWASIZE(00032)
    TASKDATALOC(ANY)
DEFINE TRANSACTION(JUGL)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER PRIMARY
TRANCODE)
    PROGRAM(JUGGLER)
    TWASIZE(00032)
    TASKDATALOC(ANY)
DEFINE TRANSACTION(JNIT)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER USER
CONFIGURATION)
    PROGRAM(JUGLINIT)
    TWASIZE(00032)
    TASKDATALOC(ANY)
DEFINE TRANSACTION(JUSC)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER SECURED
TRANCODE)
    PROGRAM(JUGGLER)
    TWASIZE(00032)
    TASKDATALOC(ANY)
DEFINE TRANSACTION(JTMO)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER TIMEOUT
TRANCODE)
    PROGRAM(JUGGLER)
    TWASIZE(00032)
    TASKDATALOC(ANY)
DEFINE TRANSACTION(JUON)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER AUTO-ON
TRANCODE)
    PROGRAM(JUGGLER)
    TWASIZE(00032)
    TASKDATALOC(ANY)
DEFINE TRANSACTION(JIVP)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER INSTALL
VERIFICATION)
```

```
     PROGRAM(JUGLIVP)
     TWASIZE(00032)
     TASKDATALOC(ANY)

DEFINE TRANSACTION(HDMO)
     GROUP(JUGLGRP)
     DESCRIPTION(CICS-JUGGLER HELP DEMO
TRANCODE)
     PROGRAM(HELPDEMO)
     TWASIZE(00032)
     TASKDATALOC(ANY)
/*
```

CICS TABLE ENTRIES FOR CICS RELEASE 3.3

```
//*
//*
//STEP6        EXEC   PGM=DFHCSDUP
//SYSPRINT     DD     SYSOUT=*
//STEPLIB      DD     DSN=CICS???.SDFHLOAD,DISP=SHR     /* NOTE 1 */
//DFHCSD       DD     DSN=CICS???.DFHCSD,DISP=SHR
//SYSIN        DD     *
DEFINE FILE(JUGLFIL)
     GROUP(JUGLGRP)
     DESCRIPTION(CICS-JUGGLER     CONTROL
FILE)
     DSNAME(CICS.JUGGLER.CONTROL.FILE)
     LSRPOOLID(1)
     DSNSHARING(ALLREQS)
     STRINGS(003)
     STATUS(ENABLED)
     OPENTIME(FIRSTREF)
     DISPOSITION(SHARE)
     DATABUFFERS(00004)
     INDEXBUFFERS(00003)
     RECORDFORMAT(V)
     ADD(YES)
     BROWSE(YES)
     DELETE(YES)
     READ(YES)
     UPDATE(YES)
DEFINE PROGRAM(STSCORE)
     GROUP(JUGLGRP)
     DESCRIPTION(UNICOM              MEMORY
DISPLAY/ALTER)
     LANGUAGE(ASSEMBLER)
     DATALOCATION(BELOW)
```

```
     EXECKEY(CICS)
DEFINE PROGRAM(STSPASS)
     GROUP(JUGLGRP)
     DESCRIPTION(UNICOM              PRODUCT
PASSWORD TABLE)
     LANGUAGE(ASSEMBLER)
     DATALOCATION(BELOW)
     EXECKEY(CICS)
DEFINE PROGRAM(STS0100)
     GROUP(JUGLGRP)
     DESCRIPTION(UNICOM              PRODUCT
PASSWORD CONTROL)
     LANGUAGE(ASSEMBLER)
     DATALOCATION(BELOW)
     EXECKEY(CICS)
DEFINE PROGRAM(JUGGLER)
     GROUP(JUGLGRP)
     DESCRIPTION(CICS-JUGGLER     RESIDENT
MODULE)
     LANGUAGE(ASSEMBLER)
     DATALOCATION(ANY)
     EXECKEY(CICS)
     RESIDENT(YES)
     CEDF(NO)
DEFINE PROGRAM(JUGLAUXL)
```

```
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER
CUSTOMIZATION)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
    EXECKEY(CICS)
DEFINE PROGRAM(JUGLENAB)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER EXIT
ENABLER)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
    EXECKEY(CICS)
DEFINE PROGRAM(JUGLINIT)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER USER
CONFIGURATION)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
    EXECKEY(CICS)
DEFINE PROGRAM(JUGLMAIN)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER MAINLINE)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
    EXECKEY(CICS)
DEFINE PROGRAM(JUGLMENU)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER MENU
GENERATION)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
    EXECKEY(CICS)
DEFINE PROGRAM(JUGLMSG)
     GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER MESSAGE
BROADCAST)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
    EXECKEY(CICS)
DEFINE PROGRAM(JUGLIVP)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER INSTALL
VERIFICATION)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
    EXECKEY(CICS)
DEFINE PROGRAM(JUGLHELP)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER HELP
PROCESSOR)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
    EXECKEY(CICS)
DEFINE PROGRAM(JUGLZNEP)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER DFHZNEP
INTERFACE)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
    EXECKEY(CICS)
DEFINE PROGRAM(JUGLHDMO)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER HELP DEMO
DISPLAY)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
    EXECKEY(CICS)
DEFINE TRANSACTION(STSC)
    GROUP(JUGLGRP)
    DESCRIPTION(UNICOM MEMORY
DISPLAY/ALTER)
    PROGRAM(STSCORE)
    TWASIZE(00032)
    TASKDATALOC(BELOW)
    TASKDATAKEY(CICS)
DEFINE TRANSACTION(JAUX)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER
CUSTOMIZATION)
    PROGRAM(JUGLAUXL)
    TWASIZE(00032)
    TASKDATALOC(ANY)
    TASKDATAKEY(CICS)
DEFINE TRANSACTION(JMNU)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER MENU
GENERATION)
    PROGRAM(JUGLMENU)
    TWASIZE(00032)
    TASKDATALOC(ANY)
    TASKDATAKEY(CICS)
DEFINE TRANSACTION(JMSG)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER MESSAGE
BROADCAST)
    PROGRAM(JUGLMSG)
    TWASIZE(00032)
    TASKDATALOC(ANY)
    TASKDATAKEY(CICS)
DEFINE TRANSACTION(JUGL)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER PRIMARY
TRANCODE)
    PROGRAM(JUGGLER)
    TWASIZE(00032)
    TASKDATALOC(ANY)
    TASKDATAKEY(CICS)
```

```
DEFINE TRANSACTION(JNIT)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER         USER         /*
CONFIGURATION)
    PROGRAM(JUGLINIT)
    TWASIZE(00032)
    TASKDATALOC(ANY)
    TASKDATAKEY(CICS)
DEFINE TRANSACTION(JUSC)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER    SECURED
TRANCODE)
    PROGRAM(JUGGLER)
    TWASIZE(00032)
    TASKDATALOC(ANY)
    TASKDATAKEY(CICS)
DEFINE TRANSACTION(JTMO)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER    TIMEOUT
TRANCODE)
    PROGRAM(JUGGLER)
    TWASIZE(00032)
    TASKDATALOC(ANY)
    TASKDATAKEY(CICS)
DEFINE TRANSACTION(JUON)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER    AUTO-ON
TRANCODE)
    PROGRAM(JUGGLER)
    TWASIZE(00032)
    TASKDATALOC(ANY)
    TASKDATAKEY(CICS)
DEFINE TRANSACTION(JIVP)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER     INSTALL
VERIFICATION)
    PROGRAM()
    TWASIZE(00032)
    TASKDATALOC(ANY)
    TASKDATAKEY(CICS)
DEFINE TRANSACTION(JHLP)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER         HELP
DIRECTORY TRANCODE)
    PROGRAM(JUGLHELP)
    TWASIZE(00032)
    TASKDATALOC(ANY)
    TASKDATAKEY(CICS)
DEFINE TRANSACTION(HDMO)
    GROUP(JUGLGRP)
    DESCRIPTION(CICS-JUGGLER HELP DEMO
TRANCODE)
    PROGRAM(HELPDEMO)
    TWASIZE(00032)
    TASKDATALOC(ANY)
    TASKDATAKEY(CICS)
```

```
***************************************************************************
*   3.3    JCL NOTE EXPLANATIONS
***************************************************************************
```

NOTE 1 Must be the library that "STSINST" phase was link edited to in the initial JCL process from the installation guide.

2 This tape drive should have the mounted master product install tape received from Unicom Systems.

3 These are temporary files that are written to disk and are deleted after use.

4 This is a CICS library that would be used to compile and link-edit a command level program.

5 This is the load library where CICS programs are kept. It must be in the concatenation for DFHRPL in the CICS startup JCL.

6 SYSUT1 and SYSUT2 should point to a source PDS for macros. If the macros already exist, you should delete them before the Install/Reinstall process.

7 VOL=(??????) - DASD volume where the JUGLFIL is to reside.

8 JUGLFILE=???????   - Must be the 7-byte VSAM filename.
JUGLFILE=BYPASS   - Will bypass the file.

9 MODE=Install for a new installation of the JUGGLER product.
MODE=Reinstall for the re-installation of the JUGGLER product.

10 JUGLFIL=BYPASS|???????
BYPASS = Bypass writing to the JUGLFIL.
?????? = Name of the JUGLFIL for your installation. STSINST will add records to this file or replace existing records depending upon the INSTALL|REINSTALL parameter.

APPENDIX D - REPORTING PROBLEMS

If a problem is encountered with **CICS-JUGGLER** which can not be resolved by the information in this document, you may call Unicom Systems at (405) 947-8080. You will be referred to a technical support representative who will take the information about the problem and resolve it then and there on the phone, if possible. Should a support person not be available when you call, you will be called back as soon as possible, usually the same day.

If the problem can not be resolved on the phone, the support representative will require that you run certain CICS diagnostic reports and send them to him. This will usually include an Auxiliary Trace of the transaction or series of transactions involved, a coredump if one is available and a listing of your User Option Table. He may also need the CICS Terminal Control Table.

When you call, please have the following information available:

1) Your Operating System.
2) The release of CICS where CICS-JUGGLER runs.
3) The release and PTF level number of CICS-JUGGLER that you have (This is shown on the User Configuration display).
4) Whether you use BTAM or VTAM.
5) Any message numbers involved in the problem, whether produced by CICS, CICS-JUGGLER or the Operating System.

Unicom Systems is committed to effective product support, and we will make every effort to resolve your problem as quickly as possible.

APPENDIX E - THE PRODUCT LINE

This appendix briefly describes all of the products currently available from Unicom Systems, Inc. These products are all available on the master installation tape and may be installed and evaluated if desired.

CICS-WINDOWS is our top-of-the-line CICS-based session manager, providing full transaction session management within CICS. In addition, interactive windowing allows up to nine windows on one screen, each capable of operating transactions as if it were in a full screen.

CICS-WINDOWS allows the terminal operator to define their terminal as two, three...up to nine "logical", or "virtual" terminals. This means that a single terminal can operate more than one task at a time. A PF or PA key is used to "toggle" from one virtual terminal to another. When the operators toggles out of a transaction, the transaction screen and all pertinent information to restart the task is saved in off-line storage. When the operator returns, the screen is restored intact and the transaction may be continued as if it were not interrupted.

In addition, it is capable of "interactive windowing". This means that the terminal operator can divide the terminal screen into multiple parts, or windows, displaying a portion of each application task in each window. Operation of each task can continue while in window mode. Windows can be sized for best fit of the application screens, or the screen display can be "panned" left, right, up and down to view additional information. This feature makes it easy for the operator to work in one application while viewing another, thereby eliminating the need for external hard-copy reports or other media.

Two additional features of CICS-WINDOWS are outbound datastream compression, which can provide savings of up to 40 percent for all outbound terminal traffic, whether the terminal is using CICS-WINDOWS or not, and the "session view" feature, which allows an operator at one terminal to view the current display of another terminal in the network.

CICS-WINDOWS offers significant savings in operator time and productivity, due to the elimination of time-consuming transactions which are required to move from one application to another. Most users experience savings on the average of 30 minutes per day, per operator. This translates to thousands of dollars in time savings per day in many installations, where hundreds or even thousands of terminal operators are present.

Three separately licensed options are available with CICS-WINDOWS. These are:

1). Menu Generation - Create and maintain on-line menus for controlling application programs.
2). Message Broadcasting - Send messages to terminals or users, queue and save messages, display messages in popup windows.
3). On-line Help Creation - Build help windows attached to application screens at transaction, screen, or field level.

CICS-JUGGLER is our CICS session manager providing full transaction session management within CICS. It contains all of the major features of CICS-WINDOWS except for interactive windowing and outbound data compression.

For users who want session management in CICS but do not care about interactive windowing, this product can be a good fit. It is priced below CICS-WINDOWS, yet still offers support for all CICS environments and provides everything you need to take advantage of transaction session management.

AUTOMON-VTAM is a software tool for the systems programmer in a CICS installation. It enhances some of the basic file-handling features of CICS. It provides the ability to define files to CICS dynamically. That is, a programmer can add an on-line file to the system without bringing the system down, which would normally be required. This means that the operation of all terminal operators can continue undisturbed, whereas it would normally be interrupted for 30 minutes or so each time a new file is defined. In addition, it allows on-line files to be opened and closed from another region, thereby alleviating scheduling bottlenecks in operations. It also provides significant savings in storage utilization within the CICS system.

AUTOMON-VTAMalso provides the ability to dynamically access any file in the CICS environment for on-line display or maintenance purposes. This means that a programmer can view or change data records without writing a program, which can result in hours or days of saved time. In addition it offers a powerful file search facility, allowing the formulation of complex boolean queries to extract selected data from a file.

AUTOMON-VTAMprovides significant advantage over standard CICS file control services by offering the following features:

- Resource Definition On-line (RDO) for the FCT
- Batch interface to open and close CICS files from a batch region.
- Dynamic allocation and deallocation (both DOS/VSE and MVS).
- Automatic (first-access) file opens for all releases of CICS.
- Automatic file closes (after specified period of inactivity).
- Automatic file closes (at specified time of day).
- Automatic disable/enable of transactions and programs associated with a file.
- Grouping of file names.
- Display, update, add or delete records of any file.
- Powerful file queries to locate and optionally extract, modify or reformat data records.

This product addresses the area of on-line documentation. It handles the two most-needed functions in this area, which are:

a). Full on-line text maintenance. Complete manuals can be created and maintained on-line. Entire manuals or selected subjects may be printed on the system printer or displayed on the terminal in "list" form (as it appears when printed). Automatic page numbering, section titles, running headers and print extraction are supported. In addition, text from other systems can be imported into the HELP-WINDOWS text file.

b). On-line retrieval of help text from application transactions. Selected sections of a manual, or stand-alone documents may be "keyed" to any application transaction in the on-line environment. A user-defined PF or PA key may be used to retrieve the associated text while operating the transaction. The associated text can be displayed either in full-screen mode or in a "window" on the screen, the position and size of which is defined by the text writer. No modifications to user programs are required to provide on-line help access. Help text can be created for any application - user-written transactions, vendor packages, fourth-generation languages, conversational programs, regardless of the method used by the application for building screens.

HELP-WINDOWS offers a "word-wrap" feature which allows text of any length to fully display in a window of any size. In addition, you can create help text with a user transaction by invoking the transaction screen, then using the cursor to "point" to the fields where help is to be made available. Help text can be associated at the transaction level, the screen level or the field level.

VTAM-WINDOWS is a multiple sessions manager operating at the VTAM level. It has full support for interactive windowing, just like CICS-JUGGLER. With VTAM-WINDOWS, you can toggle and window different VTAM applications, such as CICS, TSO, IMS, etc. A strategic feature of VTAM-WINDOWS is the use of an asynchronous VSAM file for saving screen buffers, which results in a significant reduction in storage overhead over most other VTAM session managers. Because of this feature, VTAM-WINDOWS can run very effectively in DOS/VSE, VSE/SP, VSE/ESA, MVS/SP, MVS/XA and MVS/ESA environments.

In addition to basic VTAM session management and interactive windowing, VTAM-WINDOWS offers the following features and benefits.

- Gate-way processing, provides a single entry-point to the system, with pass-through connection to VTAM applications.
- CICS-JUGGLER interface, automatic activation of CICS-JUGGLER using gate-way.
- Automatic connection to an application in any or all sessions.
- Automatic sign-on to applications.
- Command stream processing, allows a series of transactions to be automatically sent to a VTAM application.
- Cut and paste, provides ability to copy data from one application screen into another.
- Session viewing, lets an operator at one terminal view the current display of any session of another terminal using VTAM-WINDOWS.
- On-line profile changes, allows the operator to alter the choices of PF/PA key usage, window configuration and number of sessions that were set in the customization table.

VTAM-EXPRESS is a terminal datastream compression product which runs at the VTAM level. Its function is to eliminate, as much as possible, superfluous characters in datastreams that are transmitted to and from the host CPU and the terminals. Since the number of characters in a datastream is the major factor in determining the response time for an on-line transaction, the elimination of extra characters can significantly reduce that time. Most users experience a 55 to 80% reduction in terminal traffic with VTAM-EXPRESS, which results in dramatic savings both in time and cost of the computer installation usage.

With VTAM-EXPRESS you can have datastream compression for any VTAM application. CICS, IMS, TSO, any application system which communicates with terminals using VTAM can benefit. It does not require its own region or partition to operate.

VTAM-EXPRESS performs all three major types of datastream compression, that is:

- Elimination of repetitive duplicate characters.
- Outbound image saving, sending only the data fields which changed on this display.
- Inbound mirroring, removing MDT fields so that they do not re-transmit, unless they are actually modified.

Customization options are available in VTAM-EXPRESS to exclude selected applications or terminals from compression, or control the various types of compression being performed. Statistics can be viewed from an on-line display and/or logged to a disk file for batch reporting.

An additional feature of VTAM-EXPRESS is an enhanced buffer trace, which will capture and print the entire buffer contents of VTAM send and receive requests for a designated terminal. IBM's standard VTAM buffer trace will only show you the first 256 bytes of data.

GLOSSARY OF TERMS

There are various terms used in this manual and often used by people when talking about the features of the product which, while not totally unique to CICS-JUGGLER, are defined here for better understanding.