Meridian 1

Customer Controlled Routing

Release 3C

User Guide

Publication number: P0884611
Document status: Standard
Document release: 1.0
Date: September 1998

Publication history

September 1998	Standard 1.0 of this guide is released for general distribution.

Contents

Writing scripts 151

Sample scripts 175

Associating CDNs and scripts 203

CCR messages 217

Key words 229

List of terms 231

Index 241

Figures

Tables

About this guide

About the Customer Controlled Routing system

Welcome to Customer Controlled Routing (CCR).

CCR is a flexible call-routing application that allows you to define how your Automatic Call Distribution (ACD) calls should be handled and routed through a Meridian 1 system.

About this user guide

This user guide is written for ACD supervisors, managers, and system administrators who use CCR to write scripts to handle and route their incoming calls.

Before using CCR, you should go through training, or the self-paced workbook and computer-based instructional diskette available with the system. These will help you understand the concepts of working with CCR.

This user guide

- introduces first-time users to the principles and concepts on which CCR is based
- provides step-by-step procedures for creating and editing user-defined scripts
- helps users navigate through the system

Organization

Take a moment to flip through this user guide to familiarize yourself with its contents and organization. This user guide is organized into the following chapters:

Introducing CCR This chapter presents the overall concepts of CCR and presents some typical applications.

Getting started This chapter describes the CCR user interface and explains how to navigate throughout the system. This chapter also explains how to log in and out of CCR.

Maintaining user profiles This chapter describes how to create and maintain user profiles, which define the level of access end users have on CCR.

Planning your scripts This chapter presents a visual overview of the script writing process and information you should consider when planning your scripts.

Building the Variable Table This chapter explains how to create and maintain global variables, which can then be used system wide for all scripts.

Script commands This chapter describes the commands used in writing scripts.

Script language intrinsics This chapter defines how intrinsics should be used in writing scripts.

Expressions This chapter explains how to use logical, comparison, and mathematical expressions in writing scripts.

Writing scripts This chapter explains how to create, edit, view, install, validate, remove, and delete scripts.

Sample scripts This chapter presents sample scripts and tips for writing effective scripts.

Associating CDNs and scripts This chapter explains how to link scripts with Control Directory Numbers (CDNs) so that they are ready to be used system wide.

CCR messages This chapter presents a list of all messages generated by CCR.

Key words This is a list of reserved words in CCR.

List of terms Refer to this section to review the terminology of CCR.

Introducing CCR

About CCR

Customer Controlled Routing (CCR) is a flexible call-routing application that allows you to define how Automatic Call Distribution (ACD) calls are handled individually, and routed through a Meridian 1 system.

CCR helps you manage your ACD resources to best handle your call load. By using scripts (also referred to as call scripts), you can define individualized treatments for your calls. By associating scripts with Control DNs, you can provide specialized treatment for different call types.

You can combine user-friendly script commands to create various call routing and control schemes and treatments that meet your specific business needs. Call routing commands specify destinations and determine how calls are handled on their way to those destinations. Call control commands ensure that a call is directed to a place where it can either be presented or terminated. Call treatment commands determine how calls are treated while waiting in an ACD queue.

Call statistics on CCR activity are available in ACD MAX, Meridian MAX, and the ACD-C packages.

This chapter covers the following information:

- important CCR concepts (Control DNs, scripts, associations, variables, and profiles)
- typical applications

- the flexibility CCR offers you in call routing and call treatment
- major advantages of using CCR (variables, multiple queuing, dynamic priorities, special call routing and treatment on an individual call basis)

Important concepts

CCR may use terms that are unfamiliar to you. Refer to the “List of terms” chapter for a complete list of terms and their definitions.

Here are some terms that you should understand.

Script This is a collection of statements defining call routing, call control, and call treatment.

CDN A Control DN is a special DN, configured in your Meridian 1 system, to which no agents are assigned. You need to create a script to control calls in the CDN otherwise the calls are put into the default mode. A script is associated with a CDN; thus, all calls entering a CDN are handled by the same script.

Association The Association Table tells the system which script controls calls entering a CDN. A single script may control a number of CDNs.

Variable This is a user-defined name representing a value or set of values. In the Variable Table, you define variables, such as after_hours, and apply real values to them, like 17:00 to 06:00.

Intrinsics Intrinsics are real-time information provided by the system about time, date, calls, and queues.

Profile This defines the access levels for each user.

Wildcard A wildcard character (@) represents 1 through 32 digits of any value and is used in Calling Line Identification (CLID) digit-string comparisons. For example, @234 represents any CLID that ends in 234 (such as 14165551234, 01144519876234, and so on).

Placeholder A placeholder character (?) represents one digit of any value and is used in CLID digit-string comparisons. For example, ?234 represents any 4-digit CLID that ends in 234 (such as 1234, 2234, 3234, and so on).

Typical applications

CCR’s advanced call-handling features make it ideal for companies that need to provide specific call treatment for individual calls. Scripts can be as simple or as sophisticated as your application requires and are not limited to a specific number of steps. Some examples are provided below.

Credit card company example

A credit card company has one customer-service answering group for its 800 number credit cardholders and wants to provide a different level of customer service for each customer group.

The company could designate three different call treatments distinguished by Directory Number Identification Service (DNIS). The company could designate specific DNIS numbers for a variety of services, based on whether the calls were from

- platinum cardholders
- gold cardholders
- regular cardholders

See the “Sample scripts” chapter for the script for this example.

Utility company example

A utility company has one primary customer-service answering group. If a power outage occurs, the company wants to screen its calls as follows:

- Based on the CLID, calls from hospitals have top priority and hear a unique recorded announcement while waiting for an agent.
- Based on the NPANXX (a combination of Numbering Plan Area code and Local Exchange Code), calls from the affected area have lower priority. All callers hear a recorded announcement asking them to hang up if they are calling to report an outage in their area. (If the customer stays on the line, the call is handled at the lower priority.)

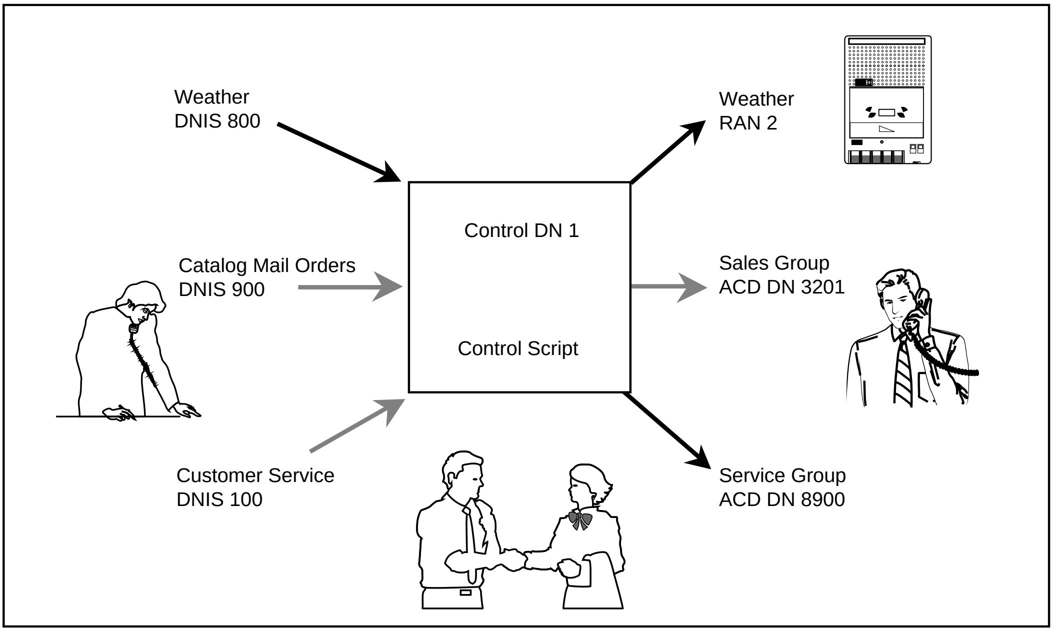
See the “Sample scripts” chapter for the script for this example.

Generic example

In Figure 1 below, several calls are routed through the same CDN and associated script and are sent to different ACD DNs for special treatment based on call information.

In this example, calls come in from three different ACD queues. The script routes the DNIS 800 weather call to a recorded announcement giving the latest weather conditions; it routes the DNIS 900 catalog call to the sales group, and the DNIS 100 customer service calls to the service group.

Figure 1
CCR call-routing example



Flexibility

CCR provides you with the flexibility you need to control your call-routing decision process. You can handle all your CCR script changes from your desktop. Call handling is defined by means of scripts, which can be changed easily to affect calls immediately.

You control three aspects of call handling with CCR:

Call routing Using a script, you can send the call to one ACD queue or multiple ACD queues, redirect the call, or reject it.

Call control You can control the call by sending it to an ACD queue, by redirecting the call, or by terminating it with a disconnect or busy.

Call treatment You can treat the call with GIVE IVR (Interactive Voice Response), GIVE RAN (recorded announcement), GIVE MUSIC, GIVE RINGBACK, GIVE SILENCE, QUEUE TO, or ROUTE TO.

Decisions for call routing and treatment depend on the available intrinsics (real-time information) and call information such as Directory Number Identification Service (DNIS) and Calling Line Identification (CLID).

Call routing

CCR screens incoming calls and determines how each call is going to be routed. CCR then controls the call until it is answered or abandoned. Even though the call is in an ACD queue, the script determines how it is routed and treated.

Using network information (DNIS, NPA, NXX, or the CLID), you can control call routing. Using intrinsics (information based on time of day, day of the week or year, and queue activity), you can further determine how the call is handled. For example, you can have a specified DNIS call sent to a special queue. In addition, you can specify that these calls arriving on Saturday would hear your weekend RAN, but weekday calls would be sent directly to the queue.

A number of methods can be used to route your calls; you can use them individually or combine them.

Queue a call A call can be placed simultaneously in up to eight separate agent groups (queues) at the same priority level or at different priority levels.

Route a call A call can be routed to any number, for example, a local DN or night number. When a call is routed, it is no longer controlled by a script or CDN.

Remove a call from a queue If a call has been in a queue too long, you can remove it so that you can redirect it, assign it to another queue, or treat it as you please.

Disconnect You can disconnect a call after Night RAN is given or if the call is one that you do not want handled.

Call control

Call scripts must provide control to a call. Otherwise, the call could never terminate. This means that the call must either be presented or terminated by a script command as follows:

Queue a call A call can be placed simultaneously in up to eight separate agent groups (queues) at the same priority level or at different priority levels.

Route a call A call can be routed to any number, for example, a local DN or night number. When a call is routed, it is no longer controlled by a script or CDN.

Busy You can give a call a busy tone to indicate that the call will not receive further handling. The call is not actually terminated until the caller ends the call.

Disconnect You can disconnect a call after Night RAN is given or if the call is one that you do not want handled.

Call treatment

Even though calls are queued to an ACD queue, the call receives the treatment defined for the CDN, not the ACD DN. Calls waiting in the same agent queue can receive different call treatments, such as music and recorded announcements. This means that you can define a large number of optional treatments for calls waiting to be answered.

Complete or Interruptible RAN You can define announcements and force callers to listen to a complete announcement before an agent receives the call, or you can determine that the call will be presented as soon as an agent is available.

Unlimited RAN You can get CCR to play unlimited messages of different types while a call is waiting (system engineering dependent).

Ringback A call receives ringback if no previous treatment is defined.

Silence You can determine that the caller hears nothing.

Busy You can initiate a busy tone when queue thresholds have been exceeded, except for trunks requiring answer supervision to return busy, such as CO, WATs, and FEX.

Music You can determine that the caller hears music.

Call Priority You can have CCR place different calls in the queue at different priority levels, or place the same call in different queues at different priority levels.

Major advantages

You can quickly and easily modify call-routing parameters to meet changing business needs. CCR offers you a tool to

- improve customer service
- increase revenue
- increase agent productivity

Calls receive dynamic priority assignment

CCR provides the ability to queue a call at priority levels 1 through 4. Within an ACD group, calls with priority 1 will always be answered first. This feature allows differentiation in the treatment of calls (for example, certain DNIS numbers could have a higher priority than others).

CCR also lets you change the priority of a call once it has been queued. This ability can be useful in combination with a time threshold.

Example

If a call arrives and is queued at priority 2, it will be answered behind priority 1 calls, behind existing priority 2 calls already queued, but *ahead of* any new priority (2 to 4) calls.

If the call waits longer than a specified amount of time, say 120 seconds, the priority of that call could be raised to priority 1. This means that the call will now be answered *behind* any priority 1 calls that are already queued but *ahead of* any new calls to priority 1 to 8 and *ahead of* any existing priority 2 to 4 calls.

For more information about queuing under Meridian 1 via X11 software, see *Advanced Feature Description* (NTP 553-2671-101).

Variables allow easy script changes

You can set up a system-wide Variable Table so that variables defined in the table can be used within scripts. Names can be assigned to represent certain values and these names can be used in scripts. When you change the value of a variable, scripts using that variable automatically use the updated value. This simplifies the management of a call center and saves you time when you are creating and modifying scripts.

Example

If you wanted to provide special treatment for certain CLID numbers, you could define the variable “vip” so that it represents several CLID numbers.

Variable Name	vip
Value	4085551234 4152312567 6192345678
Statement	`QUEUE TO 2500 WITH PRIORITY 1 IF CLID = vip`

In this example, if a call was received from one of the vip numbers, the call would be queued to ACD DN 2500 with the highest priority. The advantage of using variables is that if a value changes (for example, if the 415 area code changes to 410), that change could be made in the Variable Table in CCR. The change would automatically be reflected in all scripts using that variable without incurring system down time.

Calls can be queued at multiple ACD DNs simultaneously

CCR allows a call to be waiting for an available agent in up to eight different queues simultaneously.

The calls could be queued at each of the eight ACD DNs at the same priority or at different priorities. This lets you provide better customer service because the call has a greater chance of being answered quickly. System productivity is also improved because the system is constantly searching for an available agent in as many as eight groups, thus greatly improving the use of resources.

CCR enables specific call routing and treatment by CLID

CCR provides the capability of handling CLID calls individually. This means, for example, handling calls from hospitals first for a utility company, handling high volume purchasers first for a telemarketing firm, or handling VIP accounts first for a travel agency.

Once you set up a variable containing CLID numbers to be treated with priority, you can use the variable within any script. When a call arrives with one of the designated CLID numbers, CCR gives the call special treatment, such as queuing with the highest priority, playing special recorded announcements, or simultaneously queuing the call to several different ACD groups.

Hold in queue for Interactive Voice Response (IVR)

Using computer-controlled voice playback to prompt for telephone touch-tone input, Interactive Voice Response units provide an automated method of providing and accepting information from a caller. IVR capability can also be provided while a call is in an ACD queue. While receiving IVR treatment, the Hold in Queue for IVR feature enables the call to maintain its place in any ACD queue where it may reside.

The Hold in Queue for IVR feature contains an interruptible/non-interruptible facility, which controls whether the IVR session will be interruptible if an agent becomes available to take the call.

Getting started

Before you begin

Because CCR is a computer-based application, you should make sure you understand the concepts in this section before proceeding in this user guide. If you have used computers before, skim through this section to be sure you can relate the terminology used here to the concepts you already know.

The following information is covered in this chapter:

Using your keyboard This section covers cursor movement keys and special function keys.

Types of screens This section describes menus, pop-up menus, dialog boxes, display screens, and the editor screen.

Logging in and logging out of CCR This section covers basic login and logout procedures.

Using your keyboard

CCR supports a DEC VT220 terminal, a DEC VT320 terminal, a DEC VT420 terminal, and a standard IBM PC (or 100% compatible) running Reflection 4+ software.

Keys may vary from one keyboard to another. Figure 2 through Figure 5 illustrate differences in the keyboards. The function key options displayed at the bottom of CCR screens indicate differences in function keys with a slash (/). VT220/VT320/VT420 function keys precede the slash, while PC function keys follow the slash. For example, given the option PF3/F3=EDIT FIELD: PF3 is the EDIT FIELD key if you are using a VT220, VT320, or VT420, while F3 is the EDIT FIELD key if you are using a PC running Reflection 4+.

Figure 2
DEC VT220 keyboard

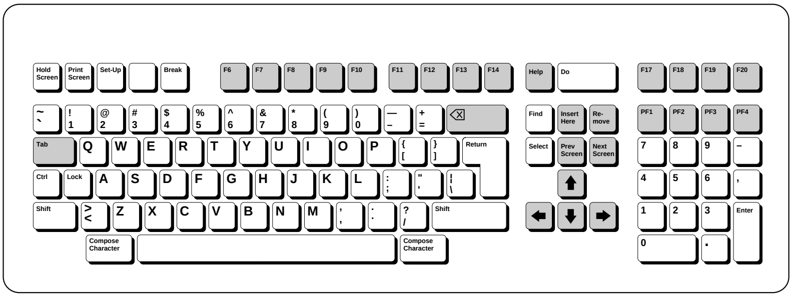

Hold Screen Print Screen Set-Up Break F6 F7 F8 F9 F10 F11 F12 F13 F14 Help Do F17 F18 F19 F20
~ ` ! 1 @ 2 # 3 $ 4 % 5 ^ 6 & 7 * 8 (9) 0 — – + = Find Insert Here Re-move PF1 PF2 PF3 PF4
Tab Q W E R T Y U I O P { [}] Return Select Prev Screen Next Screen 7 8 9 –
Ctrl Lock A S D F G H J K L : ; " ' | \ 4 5 6 ,
Shift > < Z X C V B N M , , . . ? / Shift 1 2 3 Enter
Compose Character Compose Character 0 .

Figure 3
DEC VT420 keyboard

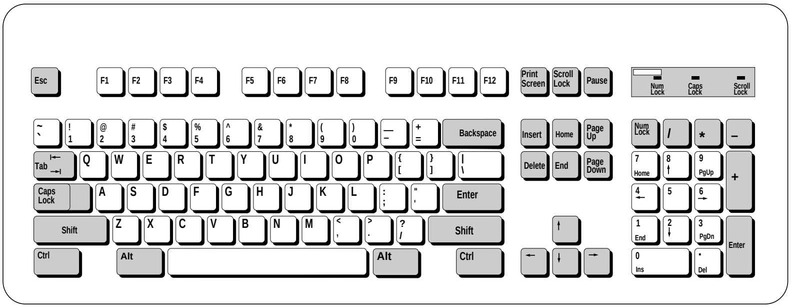

Esc F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 Print Screen Scroll Lock Pause Num Lock Caps Lock Scroll Lock
~ ` ! 1 @ 2 # 3 $ 4 % 5 ^ 6 & 7 * 8 (9) 0 — – + = Backspace Insert Home Page Up Num Lock / * –
Tab Q W E R T Y U I O P { [}] | \ Delete End Page Down 7 Home 8 9 PgUp +
Caps Lock A S D F G H J K L : ; " ' Enter 4 5 6
Shift Z X C V B N M < , > . ? / Shift 1 End 2 3 PgDn Enter
Ctrl Alt Alt Ctrl 0 Ins . Del

Figure 4
IBM XT (10 function) keyboard

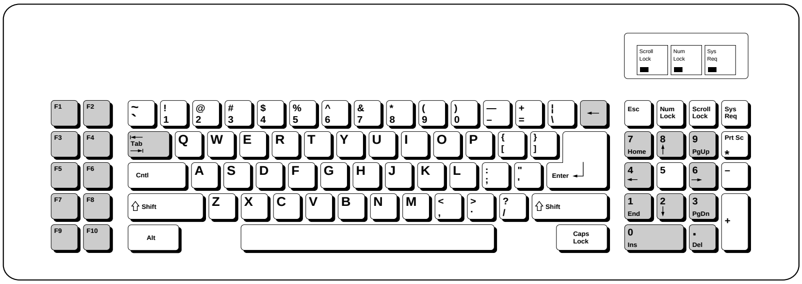
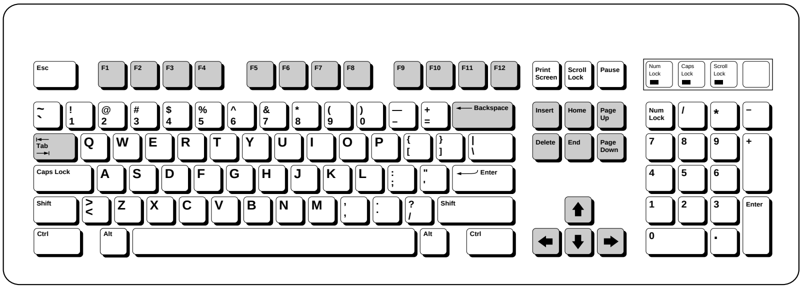


Figure 5
IBM AT Enhanced (12 function) keyboard



You can use your cursor movement keys, as well as the special function keys, to navigate through CCR. Here's how they work.

Table 1
Cursor movement keys

Key	Function
RETURN or ENTER	Accepts the information you have typed or selected. You must press this after most selections. However, if you press this after some selections, you are routed where you do not expect to go.
←	Moves the cursor one space to the left. On some screens, as noted in the function key bar, it toggles to display a Value field.
→	Moves the cursor one space to the right. On some screens, as noted in the function key bar, it toggles to display a Comment field.
↑	Moves the cursor up one line.
↓	Moves the cursor down one line.
TAB	Moves the cursor to the beginning of the next word in the script.
PREV SCREEN	Scrolls the script text up one screen, to the previous screen.
NEXT SCREEN	Scrolls the script text down one screen, to the next screen.
REMOVE/ DELETE	Deletes the character at the position of the cursor.
BACKSPACE	Deletes the character before the cursor position.
INSERT	Toggles the editor between insertion and overwrite modes. The current mode is displayed on the right side of the keylist while in the main editing window. In insert mode, all input moves the existing text to the right. In overwrite mode, the text beneath the cursor is replaced by what is being typed.

Table 2
Special keys

Key	**Function**
F keys and PF keys	Function (F) keys and programmable function (PF) keys perform specific functions, for example, exit or activate. The function keys that apply to the current screen are shown in the function key bar at the bottom of the screen.
HELP/F10	You can access online help from any place in CCR simply by pressing the HELP key.
PF1/F1	Displays the editor commands menu on the editor screen.
PF2/F2	Displays the editor options menu on the editor screen.
PF3/F3	Edit field. Starts and ends editing.
F6	Saves the current editing session in the edit pane or activates the work in session.
F7	Exits from a screen or logs out from the Main Menu.
F8	Prints a screen or moves the cursor to the beginning of the current line in scripting (BOL).
F9	Moves the cursor to the end of the current line in scripting (EOL). Also used to delete a value when editing a list in the Variable Table.

Types of screens

Before you can begin using CCR, you should understand how the user interface works. CCR displays various types of screens, specifically

- menus
- pop-up menus
- dialog boxes containing prompts
- display screens
- editor screen (only used for script writing)

Redrawing screens

To redraw a screen at any point within the CCR interface, press <CTRL> **r**.

Menus

Menus present a list of options from which you make a selection. After you make your selection, the system could display a screen or another menu.

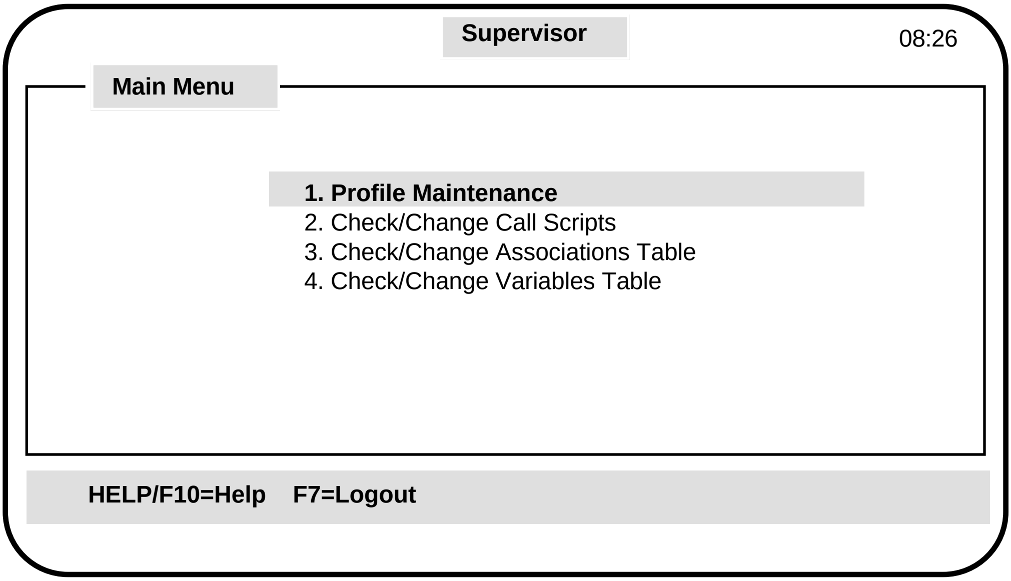
Selecting an option

You can select an option from a menu in one of the two following ways:

- by typing the number corresponding to the option you want to select and pressing Return
- by using the arrow keys to move the cursor bar up or down to highlight the option you want to select and pressing Return

Throughout this guide, the expression “select the option” refers to these two methods. Instructions are not repeated for making a selection. This guide also uses the term “Return” to represent the key labeled Return or Enter on your keyboard.

Figure 6
Sample menu



Supervisor

08:26

Main Menu

1. Profile Maintenance
2. Check/Change Call Scripts
3. Check/Change Associations Table
4. Check/Change Variables Table

HELP/F10=Help F7=Logout

Pop-up menus

Pop-up menus appear with a predefined list of choices for entering in a data field. You work with pop-up menus the same way that you work with menus.

Selecting an option

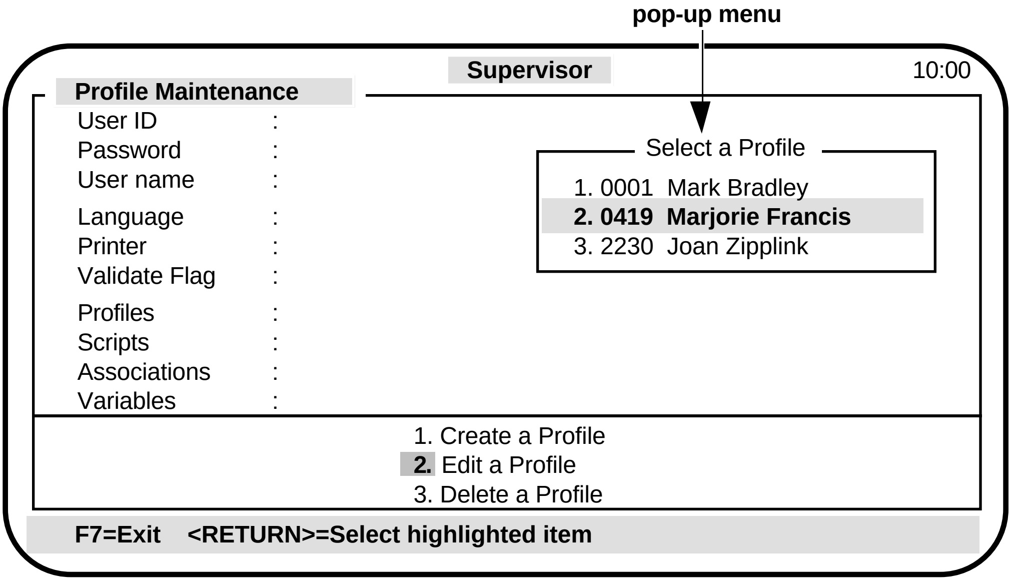
You can select an option from a pop-up menu in one of the two following ways:

- by typing the option number corresponding to the option you want to select and pressing Return
- by using the arrow keys to move the cursor bar up or down to highlight the option you want to select and pressing Return

Shortcut

You can access the pop-up menu with PF3/F3 and then type the option number of your choice (and press Return) instead of using the arrow keys.

Figure 7
Sample pop-up menu



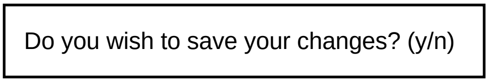
```
                              Supervisor                          10:00
 Profile Maintenance
  User ID          :
  Password         :                  Select a Profile
  User name        :
                                       1. 0001  Mark Bradley
  Language         :                   2. 0419  Marjorie Francis
  Printer          :                   3. 2230  Joan Zipplink
  Validate Flag    :

  Profiles         :
  Scripts          :
  Associations     :
  Variables        :
                            1. Create a Profile
                            2. Edit a Profile
                            3. Delete a Profile
  F7=Exit   <RETURN>=Select highlighted item
```

Note: Selections to which the user does not have access appear fainter.

Dialog boxes

Dialog boxes appear with prompts for your response. Responses can be in either uppercase or lowercase.

Figure 8
Sample dialog box



Do you wish to save your changes? (y/n)

In the above example, you are prompted to type **y** or **n**.

> ***Note:*** You don’t have to press Return after this dialog box. Typing “y” or “n” takes you out of this mode without pressing Return.

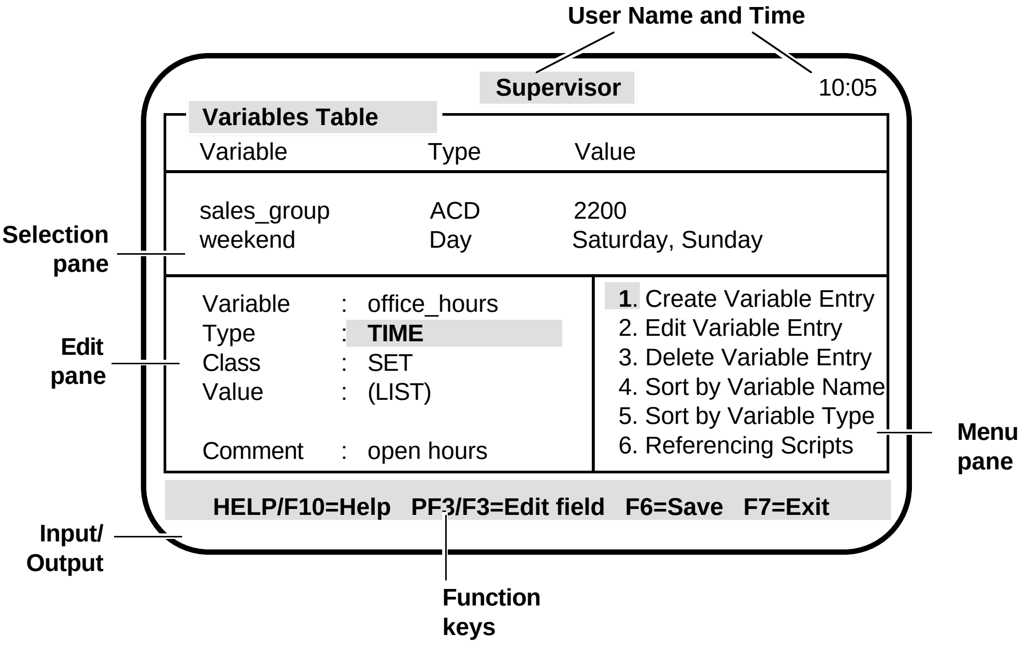
Display screens

To use the system, you must be able to move around between different areas of the screen. There are three areas of display windows:

- menu pane
- selection pane
- edit pane

If you want to exit the current pane, press F7 (EXIT).

Figure 9
Parts of a display screen



Supervisor 10:05

Variables Table

Variable	Type	Value
sales_group	ACD	2200
weekend	Day	Saturday, Sunday

Variable	:	office_hours
Type	:	**TIME**
Class	:	SET
Value	:	(LIST)
Comment	:	open hours

Table 3
Parts of a display screen

Part	Description
User Name and Time	Your User Name and the system time appear at the top of the screen.
Display windows	Up to three display windows appear for any one screen: a menu pane, edit pane, and selection pane.
Function keys	This is the highlighted bar beneath the display window showing the descriptions for special function keys that are allowed for an individual screen.
Input/output area	This area is at the bottom of the screen, but it is not seen until you start typing. If you decide to type in your choice (rather than select it), your input is displayed on the input/output line. This is also the area where error messages are displayed.

Working with menu panes

Menu panes contain options for performing functions in specific areas of the system. Working with the menu pane of a display screen is identical to selecting options from a menu. Specific function keys appear at the bottom of each screen to help you understand what action you can take.

The two ways to select an option are as follows:

- by typing the number corresponding to the option you want to select and pressing Return
- by using the arrow keys to move the cursor bar up or down to highlight the option you want to select and pressing Return

Working with edit panes

Use edit panes to enter and edit your data.

Once you have selected the edit option from the menu pane, you are automatically placed in the edit pane of a display screen.

> ***Note:*** If you want to change menu options once you are in the edit pane, press F7 to exit, do not save, and try again.

Select the field you want to edit by using the arrow keys to move the highlight bar. You can either type in your data or access a pop-up menu with predefined values (if applicable to the field you are editing). To access the pop-up menu, press PF3/F3. When you have finished selecting from the pop-up menu, press PF3/F3 again to return to the edit pane.

You can use the keyboard to edit the selected field if a pop-up menu does not appear. Simply type the information required and press Return. CCR displays the keyboard entries in the input/output area at the bottom of the screen.

CCR checks all data input as soon as you press Return. If the data is in an acceptable format, CCR inserts it into the field. If the information is not acceptable, CCR displays an error message in the input/output area at the bottom of the screen.

Saving your work from an edit pane

When you finish entering the required information in the edit pane of the screen, save your entries by pressing F6. This immediately saves the new or changed item and returns you to the menu pane.

CAUTION

When you are in an edit screen, *you must* press F6 if you want to save your changes before you exit. In some cases, if you press F7 before F6, you return to the menu pane without saving your changes.

Activating your work

After you have saved your entries, you must decide if you want to make them active. To make them active, press F6. If you *do not* activate your entries, they will be deleted.

Working with selection panes

After you have chosen to edit an entry from the menu pane of a display window, the selection pane becomes active. Use the up and down arrow keys to move the highlight bar to the entry of your choice for editing. Then press Return. If you want to exit the selection pane without making a selection, press F7.

Editor screen

This screen is unique to the script writing function and it is the screen you will use most frequently. The function keys at the bottom of your screen show three types of help, as Table 4 shows.

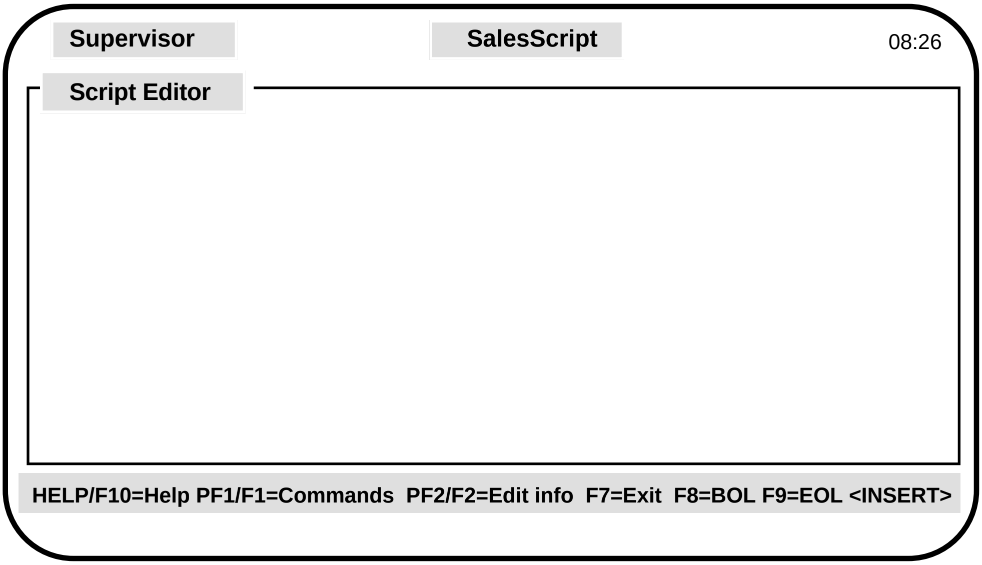
Table 4
Help function keys

Type of Help	Function Key
General help	HELP/F10 = Help
Help for commands	PF1/F1 = Commands
Help for options	PF2/F2 = Edit information

For an explanation of the other function keys, see Table 1 and Table 2.

This is a full screen editor. Most of the cursor movement keys listed in Table 1 apply to this screen.

Figure 10
Editor screen


Supervisor
SalesScript
08:26
Script Editor
HELP/F10=Help PF1/F1=Commands PF2/F2=Edit info F7=Exit F8=BOL F9=EOL <INSERT>

Logging in to CCR

CCR works with both IBM PCs running Reflection 4+ terminal emulation software and DEC VT220 terminals.

Using an IBM PC or IBM-compatible PC running Reflection 4+

If you are using a PC with Reflection 4+ terminal emulation software, there is no need for the PC to be dedicated to CCR; it can be used to run other software. To do this you must know how to start and exit Reflection 4+. The following procedure assumes that you know some basic DOS commands.

Starting Reflection 4+

Refer to the *Customer Controlled Routing Installation and Upgrade Guide* (NTP 553-3203-210) for instructions on downloading Reflection keyboard mapping for the first time. These instructions apply to non-MAX users and need only be followed once.

1 Set your current directory to the directory in which Reflection 4+ is installed.

2 Type **R4 Meridian.cfg** and press Return. The CCR login screen appears.

Using a DEC VT220/VT320/VT420 terminal

Once the VT220 terminal is connected to the Application Module, the “Console Login” prompt appears.

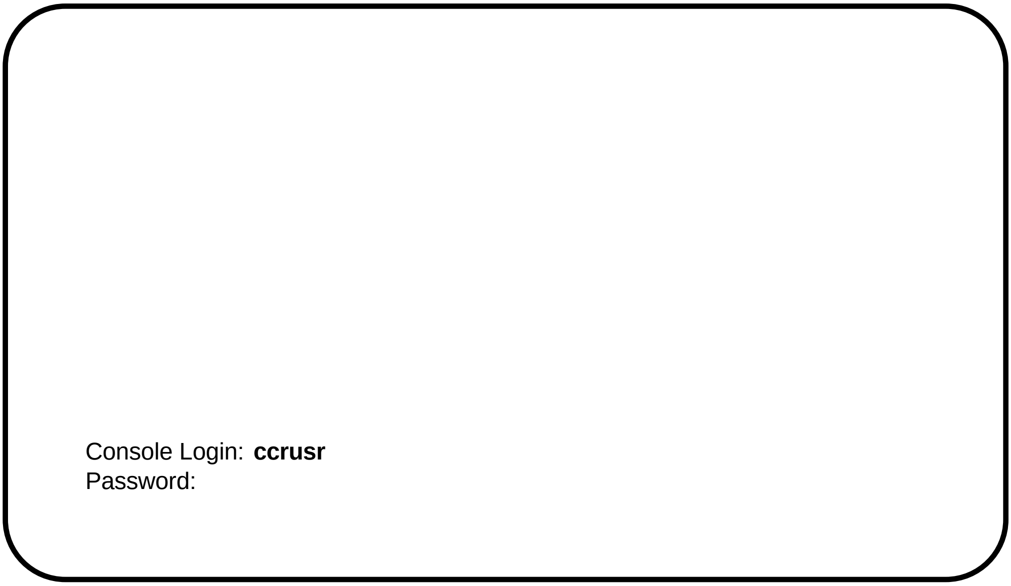
For instructions on setting up the terminal, refer to the *Customer Controlled Routing Installation and Upgrade Guide* (NTP 553-3203-210).

Completing the login procedure

Follow the next steps to complete the login procedure to CCR:

1 Type **ccrusr** and press Return.

Figure 11
CCR login screen

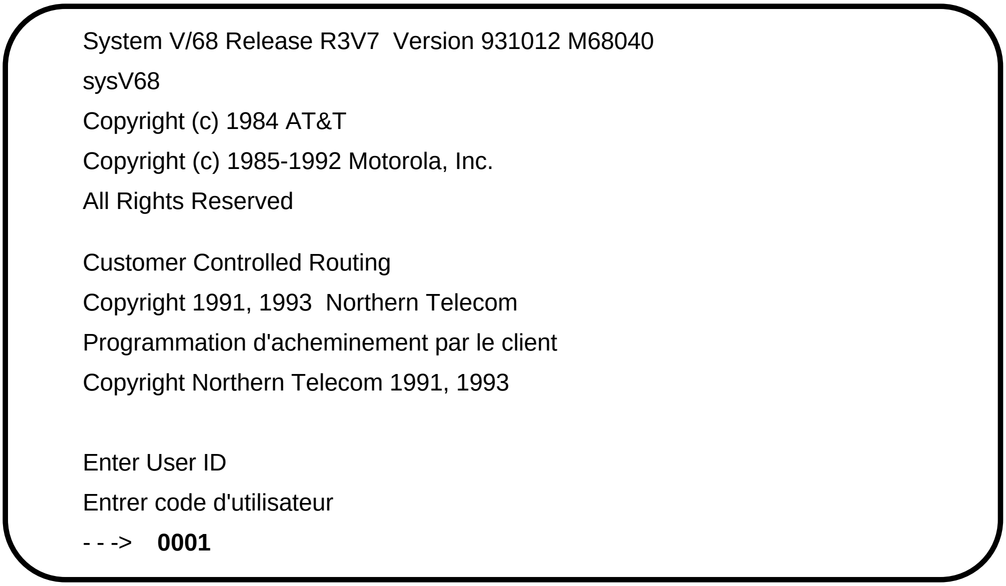


```
Console Login: ccrusr
Password:
```

The password prompt appears.

2 Type the login password, **ccrusr**, and press Return**.**

Figure 12
System Release and User ID screen



```
System V/68 Release R3V7  Version 931012 M68040
sysV68
Copyright (c) 1984 AT&T
Copyright (c) 1985-1992 Motorola, Inc.
All Rights Reserved

Customer Controlled Routing
Copyright 1991, 1993  Northern Telecom
Programmation d'acheminement par le client
Copyright Northern Telecom 1991, 1993


Enter User ID
Entrer code d'utilisateur
- - ->   0001
```

The User ID prompt appears (see Figure 12).

3 Type your User ID, (0001 is used as an example) and press Return**.**

4 Type your password, for example **admin**, and press Return**.**
Notice that your password is not displayed on the screen, because this should be confidential.

Note: Your password is case-sensitive; be sure to type it exactly as it was given to you. This means that uppercase letters must be typed in uppercase and lowercase letters must be typed in lowercase.

Logging out of CCR

Using an IBM PC or IBM-compatible PC running Reflection 4+

If you are using a PC with Reflection 4+ software, complete the following steps to log out of CCR and return to the DOS prompt:

1 Exit to the Main Menu.

2 Press F7 to log out.

3 You are no longer in CCR. Press <CONTROL> <ALT> **x**.

The DOS prompt appears.

Using a DEC VT220/VT320/VT420 terminal

Complete the following steps to log out of CCR:

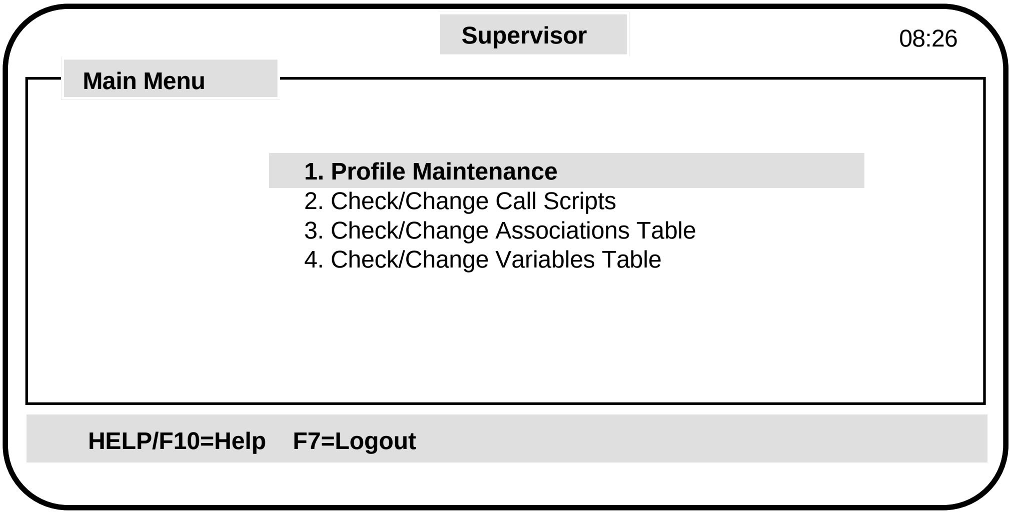
1 Exit to the Main Menu.

2 Press F7 to log out.

Main Menu

The Main Menu appears when you successfully log in to CCR. If you are having trouble, see your system administrator.

Figure 13
Main Menu



Supervisor 08:26

Main Menu

1. Profile Maintenance
2. Check/Change Call Scripts
3. Check/Change Associations Table
4. Check/Change Variables Table

HELP/F10=Help F7=Logout

The Main Menu presents four options for you to select.

Profile Maintenance This option lets you define the operating privileges for each user. If you have global access, you can set up new users and assign their privileges. If you have view access, you can customize your own environment, as well as print and display screens.

Check/Change Call Scripts This option lets you create, edit, or view existing scripts, depending upon the access level defined in your profile. Scripts can also be validated, installed or removed using this option.

Check/Change Associations Table This option lets you create and display the CDNs associated with each script, depending upon the access level defined in your profile.

Check/Change Variables Table This option lets you create and display the names or terms representing a value or set of values, depending upon the access level defined in your profile.

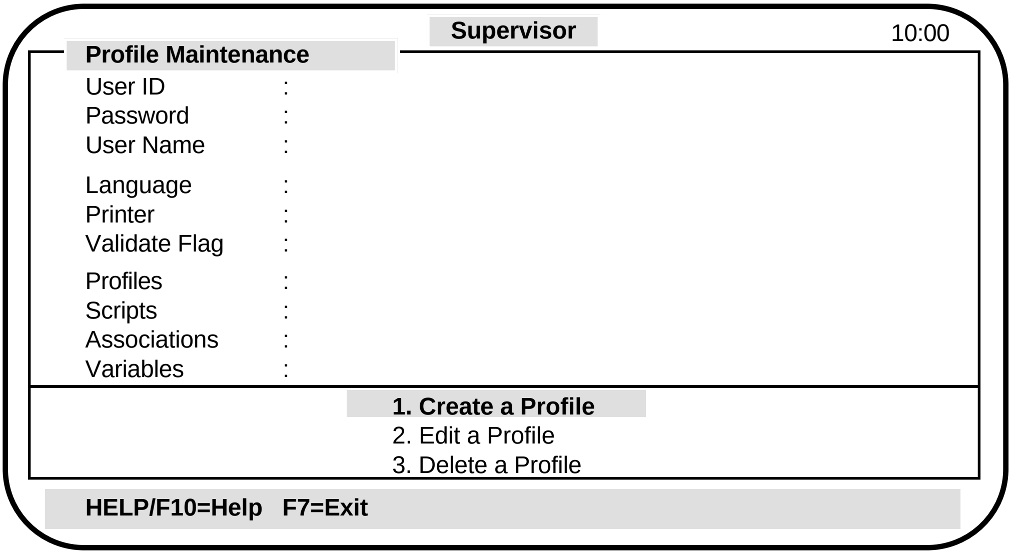
Maintaining user profiles

The Profile Maintenance screen lets you establish User IDs and define the access level for each user (User ID). This screen is typically used only during system setup or when adding or deleting a User ID.

Accessing the Profile Maintenance screen

To access the Profile Maintenance screen, select option 1 from the Main Menu.

Figure 14
Profile Maintenance screen



```
                     Supervisor                 10:00
Profile Maintenance
User ID          :
Password         :
User Name        :

Language         :
Printer          :
Validate Flag    :

Profiles         :
Scripts          :
Associations     :
Variables        :

                    1. Create a Profile
                    2. Edit a Profile
                    3. Delete a Profile

HELP/F10=Help   F7=Exit
```

User profile terms

There are three levels of access users can have for each feature: global, view, and none.

Global Users with global access can create, change, delete, and print any profiles, scripts, variables, and associations. Only one user can have global access at a time. The first global user logged in receives the global access.

View Users with view access can customize their own profiles, but no one else’s. They can only display and print scripts, associations, and variables.

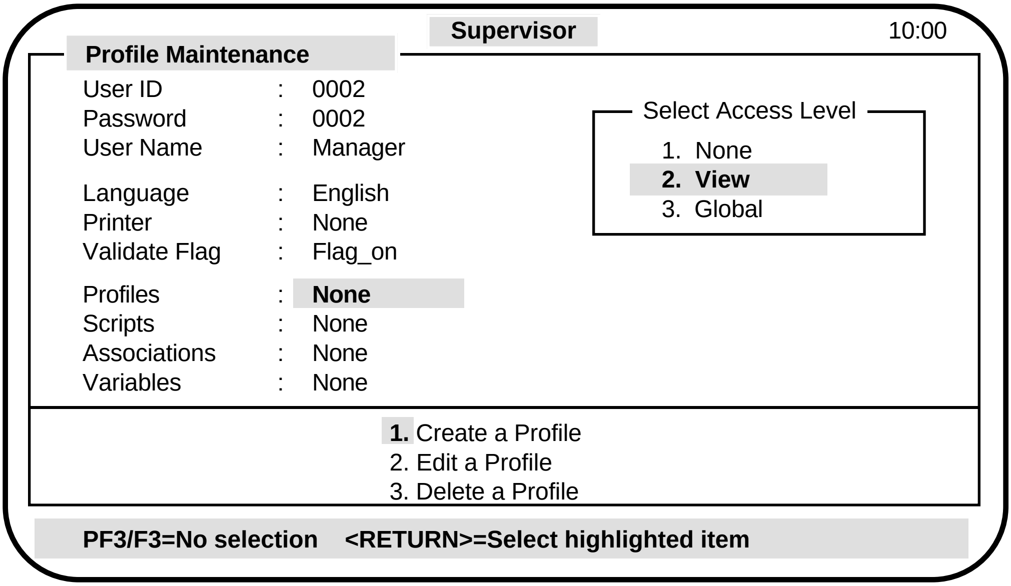
None Users with an access of none have no access to view scripts, profiles, associations, or variables.

CAUTION

Do not set up a user with no access for any features or they will not be able to do anything on CCR.

Global, view, and none access for profiles, scripts, associations, and variables appear in a pop-up menu when you select the field with the highlight bar and press PF3/F3.

Figure 15
Select Access Level pop-up menu



```
                                   Supervisor                         10:00
  Profile Maintenance
  User ID          :   0002
  Password         :   0002                  Select Access Level
  User Name        :   Manager                 1.  None
                                               2.  View
  Language         :   English                 3.  Global
  Printer          :   None
  Validate Flag    :   Flag_on

  Profiles         :   None
  Scripts          :   None
  Associations     :   None
  Variables        :   None

                          1. Create a Profile
                          2. Edit a Profile
                          3. Delete a Profile

  PF3/F3=No selection    <RETURN>=Select highlighted item
```

Creating a profile

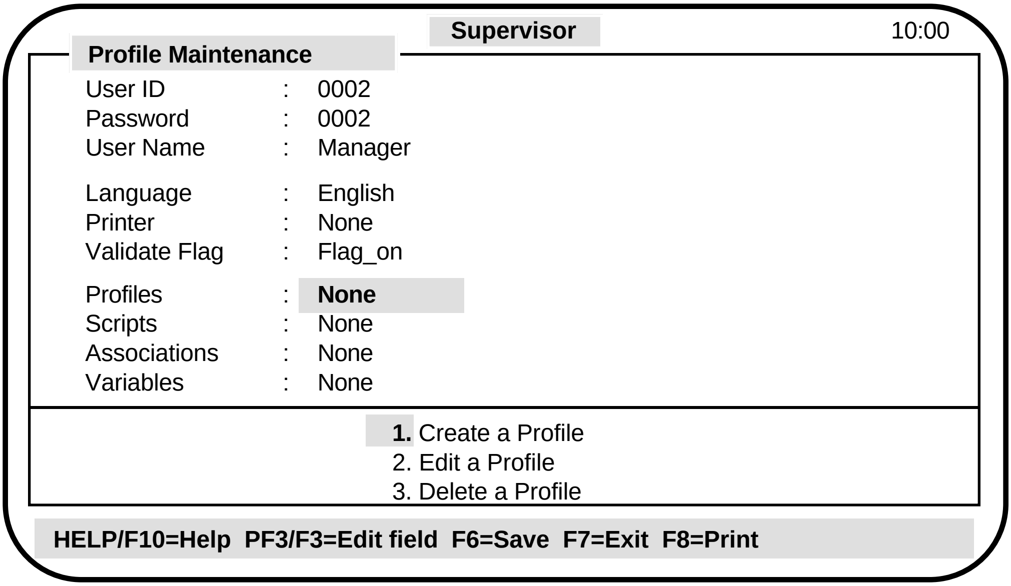
To create a new profile:

1 Select Create a Profile from the Profile Maintenance menu. The screen that appears does not contain any user information.

2 Complete the screen by typing in the data required for each field.

Note: Pop-up screens (accessed by PF3/F3) will help you select the access privileges for this new user. Simply follow the prompts at the bottom of the screen and use the function keys as indicated.

Figure 16
Create a Profile screen



```
                                  Supervisor                     10:00
  Profile Maintenance
  User ID            :   0002
  Password           :   0002
  User Name          :   Manager

  Language           :   English
  Printer            :   None
  Validate Flag      :   Flag_on

  Profiles           :   None
  Scripts            :   None
  Associations       :   None
  Variables          :   None

                              1. Create a Profile
                              2. Edit a Profile
                              3. Delete a Profile

 HELP/F10=Help  PF3/F3=Edit field  F6=Save  F7=Exit  F8=Print
```

3 Once you have created a new profile, press F6 to save it. CCR saves the new profile, and the Profile Maintenance screen now appears on the screen. The profile is enabled as soon as the user logs into CCR.

Note: If you forget this step, the system will prompt you to save your work. Type **y** and press Return.

A user profile has many characteristics that determine the type of access. These are listed below:

User ID This is the ID number for a person using CCR. The User ID must be a four-digit number between 0002 and 9999. User ID 0001 should be reserved for the system administrator.

Password This is associated with the User ID. The password is optional and must be between four and eight characters, numbers or letters. Remember that you must type the password exactly, because it is case sensitive. A password of “Mary” must have an uppercase letter M and lowercase letters for the rest. Valid characters for a password are 09, a-z, and A-Z. The maximum password length is eight characters.

User Name This is the name that appears at the top of the Profile Maintenance screen, for example, Supervisor. It is also associated with the User ID. The User Name can be no longer than 32 characters, including spaces.

Language You can select the language type from a pop-up menu. Any language listed in that pop-up menu is valid. Screens, help text, and most messages appear in the language specified. Script commands are in English only.

Printer You can select a printer from a pop-up menu. You can also select any printer listed in the pop-up menu as the default printer.

Validate flag This allows access to warning messages. A pop-up menu offers Flag_on or Flag_off. If you select Flag_on, CCR displays script warning messages. If you do not want to see script warning messages, select Flag_off (not advisable for new users). CCR always displays script errors.

Profiles Global access allows users to create and change profiles. View access allows users to change their own profiles only. With no access, users cannot access any profile screen.

Scripts Global access allows users to create and change scripts. View access allows users to view and print scripts only, but not alter them. With no access, users cannot access any script screen.

Associations Global access allows users to create and change associations. View access allows users to view and print association screens. With no access, users cannot access any association screen.

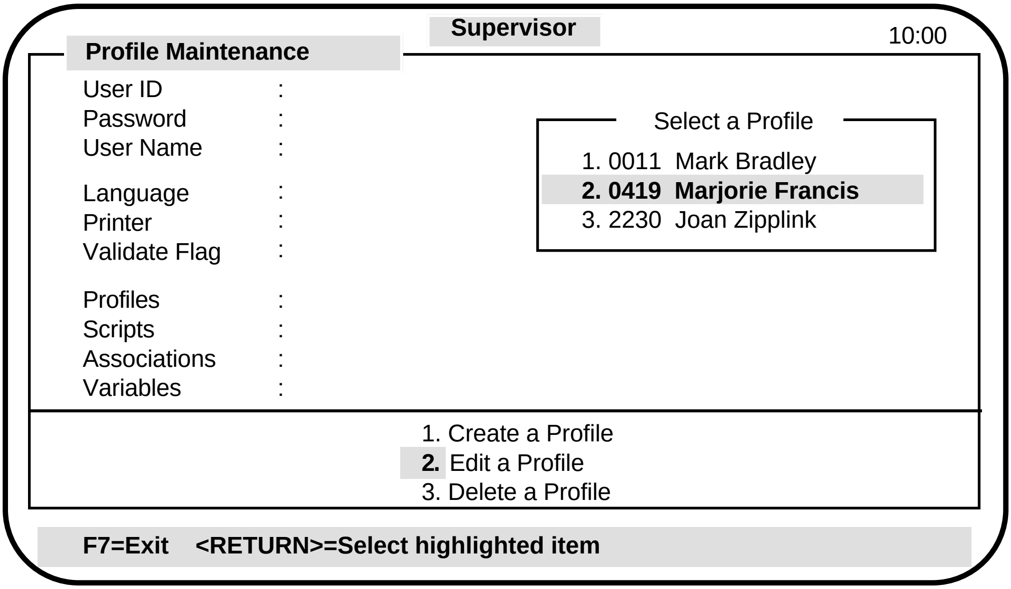
Variables This determines the access users have to the Variable Table entries. A user with global access can create and change entries. A user with view access can only view and print screens. A user with no access cannot access any variable screen.

Editing a profile

To edit a profile, perform the following steps:

1 Select the Edit a Profile option from the Profile Maintenance menu.

Figure 17
Edit a Profile screen



```
                                  Supervisor                        10:00
   Profile Maintenance
   User ID          :
   Password         :                       Select a Profile
   User Name        :
                                        1. 0011  Mark Bradley
   Language         :                   2. 0419  Marjorie Francis
   Printer          :                   3. 2230  Joan Zipplink
   Validate Flag    :

   Profiles         :
   Scripts          :
   Associations     :
   Variables        :

                              1. Create a Profile
                              2. Edit a Profile
                              3. Delete a Profile

   F7=Exit    <RETURN>=Select highlighted item
```

A pop-up menu appears, displaying the existing profiles. The profiles are listed by User Name, with the first entry highlighted. You can edit any profile shown in the pop-up menu. All edit changes take effect when that user logs in to CCR.

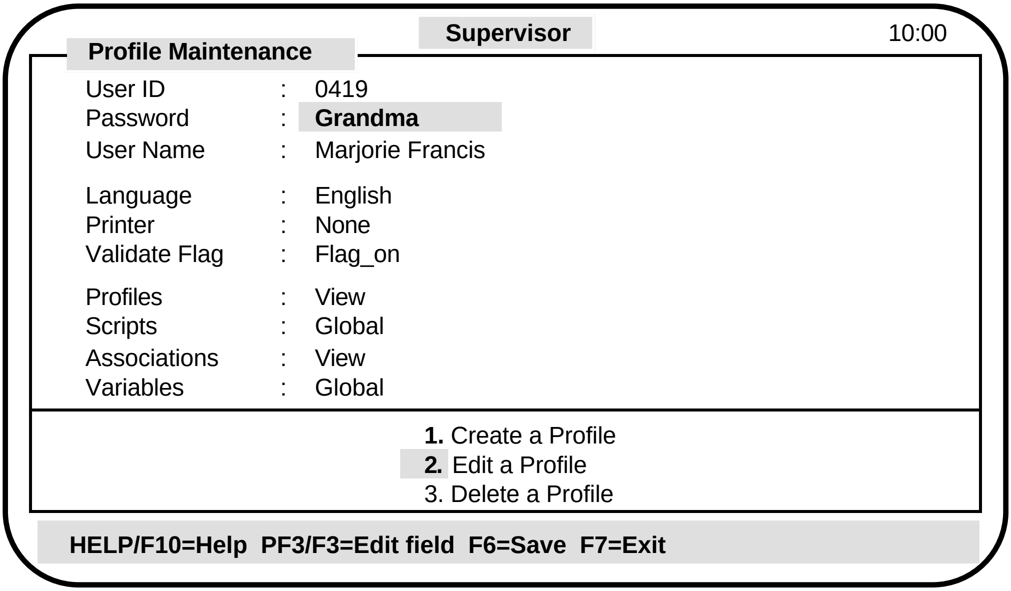
2 Select the profile you want to edit from the pop-up menu. The selected profile appears on the screen. You are now in the edit section, with the existing values for the profile appearing and the User ID highlighted.

3 Select the field you want to change.

Note: Keep records of any changes you make for a given profile. Changes made in Profile Maintenance affect user access to CCR.

4 Type the new information and press Return. As you type, the information appears in the input/output area at the bottom of the screen.

Figure 18
Edit a Profile window



Supervisor 10:00

Profile Maintenance

```
User ID          :  0419
Password         :  Grandma
User Name        :  Marjorie Francis

Language         :  English
Printer          :  None
Validate Flag    :  Flag_on

Profiles         :  View
Scripts          :  Global
Associations     :  View
Variables        :  Global
```

1. Create a Profile
2. Edit a Profile
3. Delete a Profile

HELP/F10=Help PF3/F3=Edit field F6=Save F7=Exit

Once entered, the new information is highlighted.

In many cases, if you make a typing mistake (for example, if you attempt to type too many characters in a field), the system immediately displays a message in the Input/Output area at the bottom of the screen.

The highlight bar remains on your selected item and the original information remains.

5 To revise a field you have already entered, use the up or down arrow key to move to the field, retype the information, and press Return.

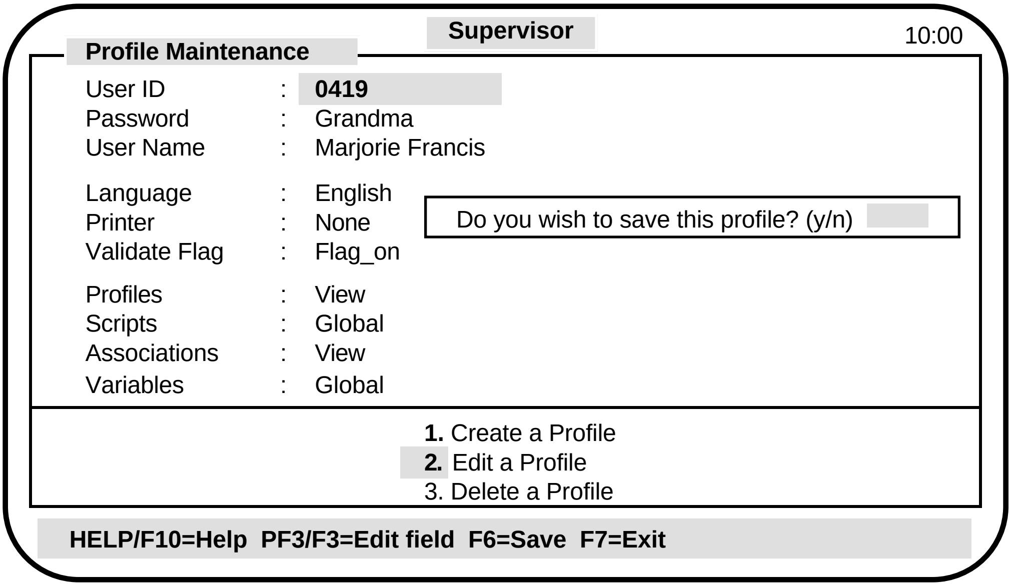
Note: Press HELP/F10 at any time if you need additional information about options and commands.

Saving a profile

To save a profile, perform the following steps:

1 Once you have edited a profile, press F6 to save it.

Figure 19
Save the window



Supervisor 10:00

Profile Maintenance

User ID : **0419**
Password : Grandma
User Name : Marjorie Francis

Language : English
Printer : None
Validate Flag : Flag_on

Profiles : View
Scripts : Global
Associations : View
Variables : Global

Do you wish to save this profile? (y/n)

1. Create a Profile
2. Edit a Profile
3. Delete a Profile

HELP/F10=Help PF3/F3=Edit field F6=Save F7=Exit

Note: If you forget this step, the system will prompt you to save your work. Type **y**.

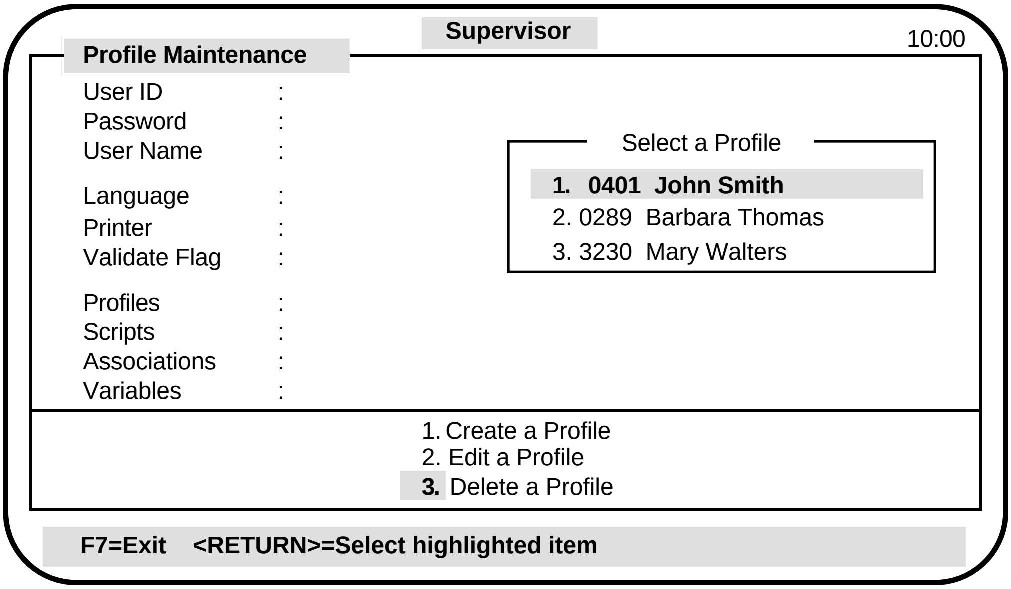
CCR saves the edited profile, and the Profile Maintenance edit pane now appears on the screen. The prompt “Edited profile will be effective on next login” appears. The profile is enabled as soon as this user logs in to CCR.

Deleting a profile

To delete a profile, perform the following steps:

1 Select Delete a Profile from the Profile Maintenance menu.

Figure 20
Delete a Profile window



```
                                 Supervisor
                                                                   10:00
  Profile Maintenance
  User ID         :
  Password        :
  User Name       :                        Select a Profile
                                      1.  0401  John Smith
  Language        :                   2. 0289  Barbara Thomas
  Printer         :                   3. 3230  Mary Walters
  Validate Flag   :

  Profiles        :
  Scripts         :
  Associations    :
  Variables       :

                            1. Create a Profile
                            2. Edit a Profile
                            3. Delete a Profile

  F7=Exit   <RETURN>=Select highlighted item
```

A pop-up menu appears, displaying the existing profiles.

The profiles are listed by User Name with the first entry highlighted. You can delete any profile shown in the pop-up window. Deleted profiles take effect immediately. If the user is currently logged in, the deleted profile is effective after logoff.

2 Select the profile you want to delete from the pop-up menu.

A prompt appears asking you to verify that you want to delete this profile.

3 Type **y** to delete it, or type **n** to cancel the deletion; then press Return.

Planning your scripts

Overview

This chapter contains tips to help you plan how to best write and organize your scripts.

Information covered in this chapter includes the following:

- things to consider before you begin
- visual overview of the scripting process
- forms to help you organize your associations, variables, and scripts

Before you begin

Before you can write a script, or create profiles, associations and variables, you will need to know the following:

- available ACD DNs
- available CDNs
- available DNs
- available RAN routes
- available Music routes
- number of agents per queue
- call center working hours
- holidays

Some of the information (for example, ACD DNs, CDNs) can be printed using overlays in the Meridian 1 system. If you do not know how to do this, ask your system administrator to get this information for you. File this information in a CCR folder or notebook so you will always have it near your CCR terminal when you need to create or edit a script.

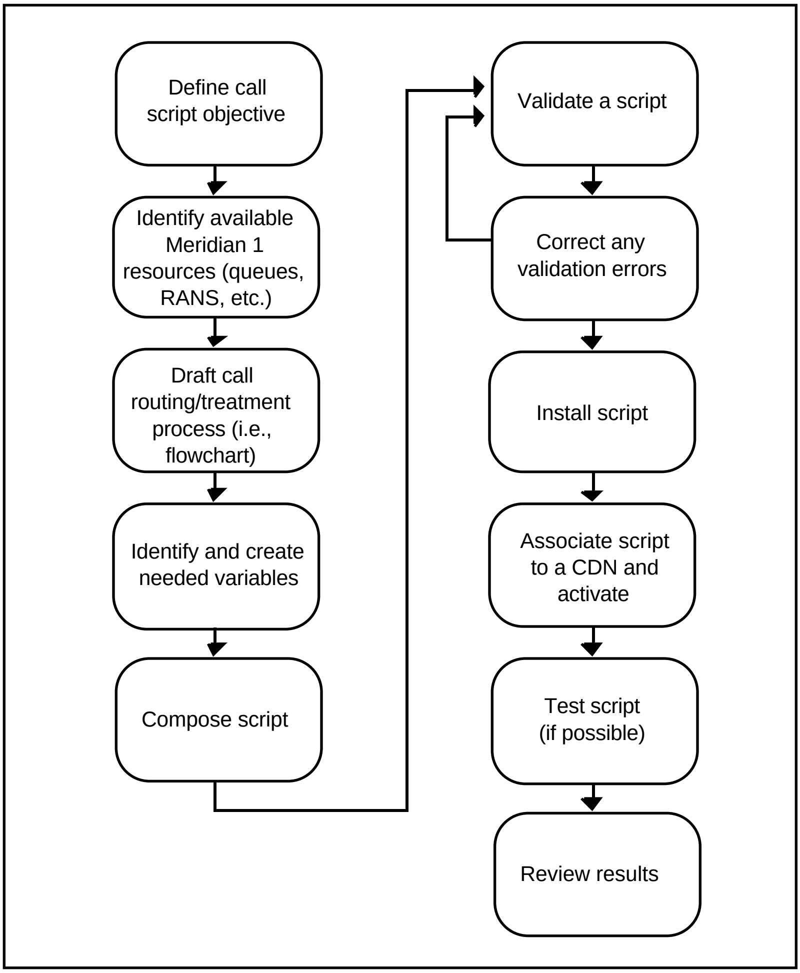
You should also print copies of all profiles, scripts, the Variable Table, and Association Table, and file them in a CCR notebook or folder for reference. There are many reasons for doing this:

- You can quickly look up information regarding your CCR system without having to log on to the system.
- Someone else can easily look through this notebook or folder and see what information already exists. This is particularly helpful if you are not available to answer questions (for example, if you are on vacation).
- If a script or profile is inadvertently deleted from the system, or if there is some system problem, you will have a hard copy to use as a reference in case you need to re-enter information into the system.

Script writing process

The following flowchart shows an overview of the script writing process.

Figure 21
Script writing process



Define call script objective
Identify available Meridian 1 resources (queues, RANS, etc.)
Draft call routing/treatment process (i.e., flowchart)
Identify and create needed variables
Compose script
Validate a script
Correct any validation errors
Install script
Associate script to a CDN and activate
Test script (if possible)
Review results

Sample forms

Photo enlarge these sample forms for best results.

Table 5
Association Table Worksheet

Associations Table Worksheet			
CDN	SCRIPT	ON/OFF	COMMENTS

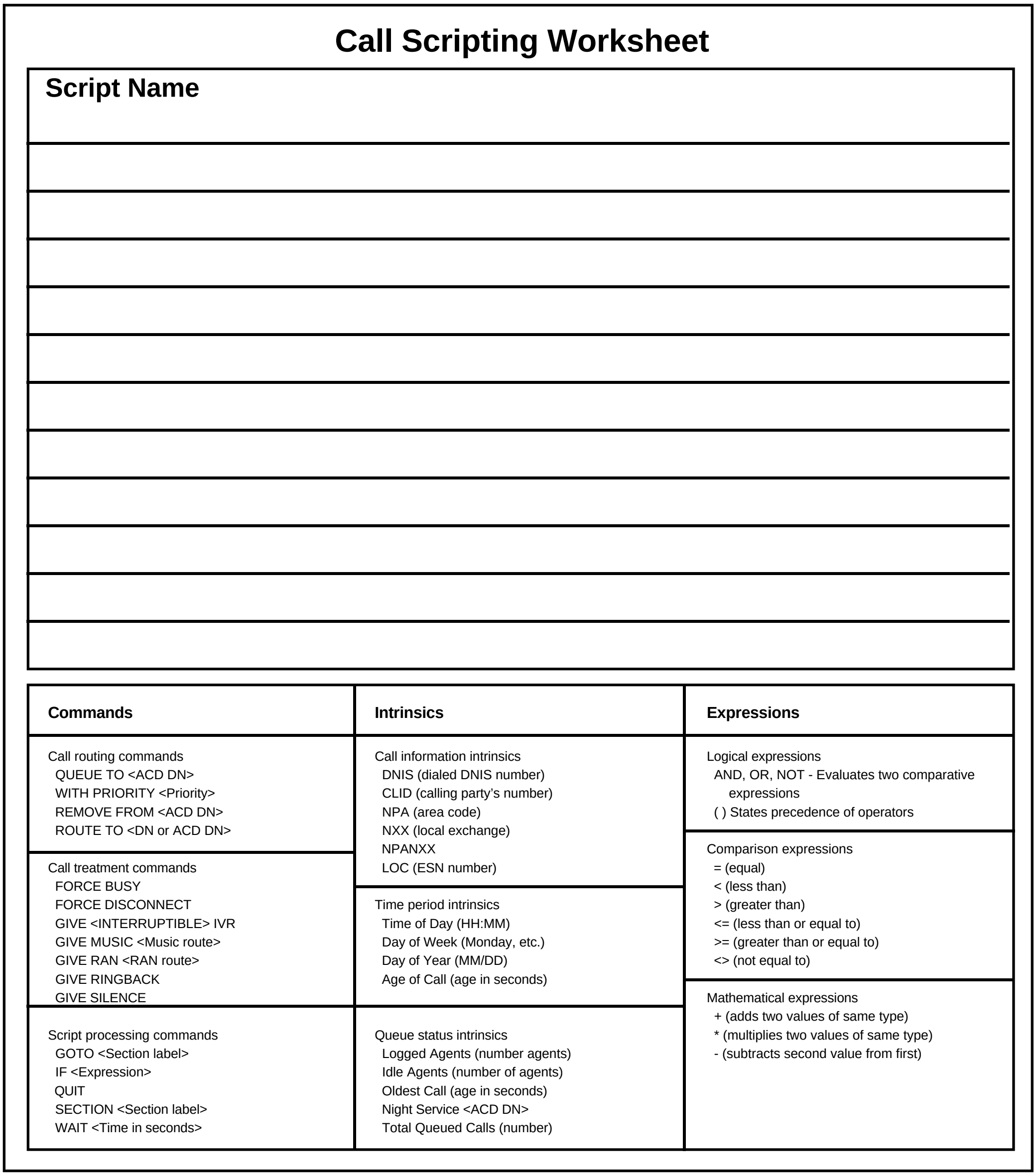
Figure 22
Variable Table Worksheet

Variable Table

Variable	Type	Class	Value	Comments

Variable Types	Definition	Allowed Value	Class
ACD	ACD DNs	2-7 digits (ACD-DN defined on Meridian 1)	Item
Priority	Priority level at which a call may be queued	1,2,3 or 4 (1 indicates highest priority)	Item
DN	Internal DN (other than ACD DNs), & external numbers	1-32 digits	Item
Music	Music routes (route #)	0-511	Item
RAN	RAN routes (route #)	0-511	Item
seconds	Time in seconds	0-32767	Item
time	Time in hours and minutes; based on 24-hour clock	0:00-23:59	Item or set
day	Day of week (with Monday being the first day of week)	Monday-Sunday	Item or set
date	Day of the year	MM/DD (Month/Day)	Item or set
integer	Any number	0-32767	Item
DNIS	DNIS number	1-31 digits	Item or set
CLID	Calling party’s number	1-32 digits	Item or set
NPA	Calling party’s area code	3 digits (first digit is 2-9)	Item or set
NXX	Calling party’s local prefix	3 digits (first digit is 2-9, second digit is 0-9)	Item or set
NPANXX	Calling party’s area code and local prefix	6 digits	Item or set
LOC	Calling party’s location code in a private numbering plan	3 digits	Item or set
Treatment	Treatment specified for IVR	1-7 digits	Item
WildCLID	Calling party’s number containing wild card and placeholder characters	1-32 digits	Item or set

Figure 23
Call Scripting Worksheet



Call Scripting Worksheet

Script Name

Commands	Intrinsics	Expressions
Call routing commands QUEUE TO <ACD DN> WITH PRIORITY <Priority> REMOVE FROM <ACD DN> ROUTE TO <DN or ACD DN>	Call information intrinsics DNIS (dialed DNIS number) CLID (calling party's number) NPA (area code) NXX (local exchange) NPANXX LOC (ESN number)	Logical expressions AND, OR, NOT - Evaluates two comparative expressions ( ) States precedence of operators
Call treatment commands FORCE BUSY FORCE DISCONNECT GIVE <INTERRUPTIBLE> IVR GIVE MUSIC <Music route> GIVE RAN <RAN route> GIVE RINGBACK GIVE SILENCE	Time period intrinsics Time of Day (HH:MM) Day of Week (Monday, etc.) Day of Year (MM/DD) Age of Call (age in seconds)	Comparison expressions = (equal) < (less than) > (greater than) <= (less than or equal to) >= (greater than or equal to) <> (not equal to)
Script processing commands GOTO <Section label> IF <Expression> QUIT SECTION <Section label> WAIT <Time in seconds>	Queue status intrinsics Logged Agents (number agents) Idle Agents (number of agents) Oldest Call (age in seconds) Night Service <ACD DN> Total Queued Calls (number)	Mathematical expressions + (adds two values of same type) * (multiplies two values of same type) - (subtracts second value from first)

Building the Variable Table

Overview

Variables allow you to assign names, numbers, time, queue priorities, or ACD queues to easily understood terms.

A variable is a name used in a script to represent an assigned value or values. All variables are stored in a single Variable Table. All scripts refer to that table for the values of the variables. This makes it easier for someone else to use the variables you create.

There is no limit on the number of variables that can be put into the Variable Table, other than that imposed by limitations on the availability of system memory.

Changing a variable in the Variable Table automatically updates the value (or values) in all scripts using that variable.

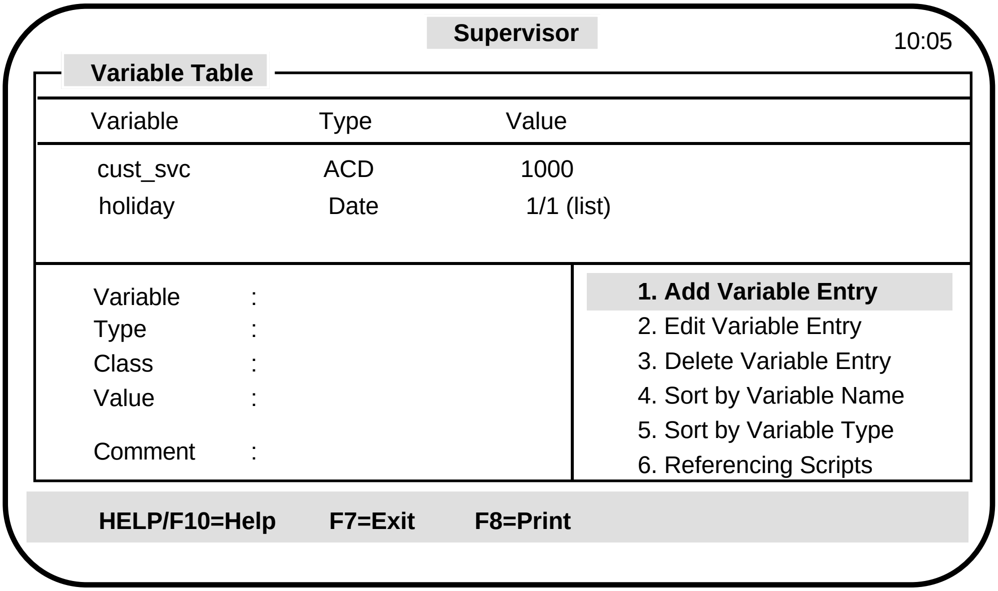
For example, if you have defined your customer service ACD DN as 2500 and you represent this in the Variable Table as service_dn, any script can then use the name service_dn in place of the number 2500. For example, QUEUE TO 2500 could be changed to QUEUE TO service_dn. Simply changing the ACD DN in the Variable Table changes it in all the scripts that reference the service_dn.

> ***Note:*** When you change the value of a variable, only new calls are affected.

Accessing the Variable Table screen

To access the Variable Table screen, select option 4 from the Main Menu. Refer to the "Getting started" chapter to understand how to navigate the three areas of the Variable Table screen.

Figure 24
Variable Table screen



Supervisor 10:05

Variable Table

Variable	Type	Value
cust_svc	ACD	1000
holiday	Date	1/1 (list)

Variable :
Type :
Class :
Value :
Comment :

1. Add Variable Entry
2. Edit Variable Entry
3. Delete Variable Entry
4. Sort by Variable Name
5. Sort by Variable Type
6. Referencing Scripts

HELP/F10=Help F7=Exit F8=Print

Variable Table terms

This section employs the following tables to describe the Variable Table:

Variable Table terms Table 6 describes fields in a Variable Table record. Terms include variable, type, class, value, and comment. The definition column defines the term, describes its function, its valid values, and its use.

Variable Table data types Table 7 describes data types, their formats, and any rules that apply to their use in the Variable Table.

Summary of valid Variable Table items and sets Table 8 gives valid value ranges (with examples) for data types classified as items or sets.

Table 6
Variable Table terms

Terms	Definition
Variable	The name you use to refer to a user-defined value. This is what you use within a script. For example, a variable name may be “main_acd”. Spaces are not used; use an underscore (_) to space between words. Maximum length of a variable is 32 characters. The variable names you see when adding or editing the Variable Table are very specific. You must be sure to use the variables exactly as they are shown.
Type	Refers to the kind of data the name represents. Type determines the value accepted by CCR. For example, for the variable “main_acd,” the type is ACD, indicating an ACD DN. This tells CCR to accept only digit strings 2–7 digits long. CCR also verifies that the specific ACD variable exists in the system. There are 18 types used in scripts. (See Table 7 for type definitions.)
Class	Refers to how the items are categorized by the CCR application. Class is either an item or set. Item A single entry. Set A set can be a list of items, range of items, or a list of ranges. A range, for example, might be Monday through Wednesday. Use periods to specify a range. For example, “Monday..Wednesday” specifies Monday through Wednesday. A list could define non-consecutive information, such as Monday, Wednesday, and Friday. A pop-up menu allows list entries, with each item in the list entered separately, one after the other. The maximum number of items, ranges, or a combination of both is 60.
Value	Links the real system information to the variable name. For example, the variable “main_acd,” being type ACD, might have 2500 entered as the value. Throughout a script the ACD DN 2500 would be used wherever the variable “main_acd” occurs. A value can be a single item or a set, depending on the variable class.
Comment	Text that explains your variable.

Table 7
Variable Table data types

Data type	Definition
ACD	This type is used for variables that represent ACD DNs. It is a digit string 2–7 digits long that is verified with the Meridian 1 system to ensure that it is a defined ACD DN. An ACD variable may only be defined as an item.
CLID	This type is used for variables that represent calling party numbers. It is 1–32 digits long. CLID types can be items or sets (a set can be made up of lists and ranges). They can be internal DNs or numbers presented from the network. CLID may be a standard range, but not a wrapped range. For example, the expression c_number = 1..9999 is valid. However, the expression c_number = 9999..1 is not valid. The wrapped range includes all numbers 9999–99999999999999999999999999999999 (32 digits), as well as 1.
DATE	This type is used for variables that represent specific days of the year. The format is MM/DD. MM refers to the month and DD refers to the day. The first day of the year is 01/01; the last day of the year is 12/31. Dates will not wrap past 12/31 unless specified in a range. A DATE variable may be specified as an item or a set. The expression Day Of Year = 08/01..07/31 is a valid example of a wrapped range because it encompasses the end of one year and the beginning of another. The expression Day Of Year > 12/31 is not valid because 12/31 is the last day of the year, and a range has not been specified to make the date wrap past that day. To make this expression valid, use a range to specify the DATE. For example, Day Of Year = 12/31..01/01 is also a valid example of a wrapped range.

— continued —

Table 7
Variable Table data types, continued

Data type	Definition
DAY	This type is used for variables that represent the days of the week. The format uses the following order: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday. Monday is the first day of the week, and Sunday the last. (Unless a range is specified, the week does not wrap past Sunday). A DAY variable may be specified as a set. For example, the expression Day Of Week = Monday, Wednesday, Friday is true on those three days and false otherwise. A DAY variable may also be specified as a wrapped range. The expression Day Of Week = Saturday..Monday is true on Saturday, Sunday, and Monday. DAY may be an item or a set.
DN	This type is used for variables that represent internal DNs (other than ACD DNs) and external numbers (any number that can be dialed on the Meridian 1 to make an external call). This type consists of a digit string 1–24 digits long. A DN variable may only be defined as an item.
DNIS	This type is used for variables that represent DNIS numbers. It is a digit string 1–31 digits long. DNIS variables may be items or sets. (Meridian 1 X11 Release 24 is required to support more than seven DNIS digits.)
INTEGER	This type is used for variables that represent a number that needs to be expressed. The allowed values are 0–32767. Integers may be operated on by mathematical expressions. Thus, the expression Total Queued Calls main_acd > four * twelve is true when the number of calls queued at the ACD represented by the variable “main_acd” is greater than 48 (4x12)—provided the variable “twelve” is of type INTEGER with the value 12 and the variable “four” is of type INTEGER with the value 4. INTEGER variables may be defined as items only.
— continued —	

Table 7
Variable Table data types, continued

Data type	Definition
LOC	This type is used for variables that represent the calling party number’s Location Code in a private numbering plan (for example, ESN). It is 3 digits long. There are no restrictions on what digits can be used. An LOC variable may be an item or a set.
MUSIC	This type is used for variables that represent Music routes. It is a number from 0 to 511. A MUSIC variable may only be defined as an item.
NPA	This type is used for variables that represent the calling party number’s Numbering Plan Area (area code). The NPA consists of three digits. Effective January 1995, all products must comply with the new NPA naming convention, where the first digit is 2–9, and the second and third digits are 0–9. The NPA is distinguished from the NXX by its position in the 10-digit CLID: the NPA is the first three digits and the NXX is the second three digits. An NPA variable may be an item or a set.
NPANXX	This type is used for variables that represent the calling party number’s Numbering Plan Area plus Exchange. It is six digits long and the first three digits form the NPA, and the second three digits form the NXX. An NPANXX variable may be an item or a set.
NXX	This type is used for variables that represent the calling party number’s Exchange (local prefix). It is three digits long; the first digit is be 2–9, and the second and third digits are 0–9. An NXX variable may be an item or a set.
PRIORITY	This type is used for variables that represent the priority level at which a call may be queued. It will be the digit “1,” "2," "3," or "4," with “1” indicating the highest priority. The digit “1” is the default priority. A PRIORITY variable may only be defined as an item.
RAN	This type is used for variables that represent recorded announcement (RAN) routes. It is a number from 0 to 511. A RAN variable may only be defined as an item.
— continued —	

Table 7
Variable Table data types, continued

Data type	Definition
SECONDS	This type is used for variables whenever the time in seconds needs to be specified. It is a number from 0 to 32767 (9.1 hours). Constants or variables of this type may be operated on by mathematical expressions. A SECONDS variable may be defined only as an item.
TIME	This type is used for variables that represent the time of day in terms of hours and minutes. Based on a 24-hour clock, it uses the format HH:MM. HH refers to the hour and MM refers to the minutes. The 24-hour clock begins at 0:00 (midnight) and ends at 23:59 (11:59 P.M.). For example, 5 P.M. is expressed as 17:00. A TIME variable may be an item or a set. The set may include wrapped ranges. The expression Time Of Day = 22:00..06:00 is a valid example of a wrapped range representing the time period from 10 P.M. (22:00) to 6:00 A.M. (06:00).
TREATMENT	This type is an option for the GIVE IVR command. It determines the Meridian Mail voice service prompt or mailbox greeting the caller will hear. The default is for TREATMENT to be off. Until the link is established for IVR, treatment does not appear as an option. A TREATMENT variable may be defined only as an item. For information on how the default treatment is defined, see “GIVE IVR treatment” in the “Script commands” chapter.
WILDCLID	This type is similar to CLID except that *ranges are not allowed*. WILDCLID types contain wildcard (@) and/or placeholder (?) characters. The expression CLID = 44@ is true for all numbers of 2–32 digits that start with 44. The expression CLID = 44? is true for all 3-digit numbers that start with 44 (such as 440, 441, 442).

— continued —

Table 7
Variable Table data types continued

Data type	Definition
WILDCLID (continued)	*Rules for using wildcard characters:* 1. Only one wildcard character is allowed in any CLID digit string. 2. A wildcard character can be used only at the beginning or end of a string. 3 A wildcard character cannot be used in a range expression but can be used in a list. 4 A wildcard character can only be used with equal (=) or not-equal (<>) comparison operators. 5. A wildcard character can be used only in a CLID intrinsic of the WILDCLID data type. 6. Only variables of the WILDCLID data type will allow the wildcard character to be entered as a value. *Rules for using placeholder characters:* 1. A placeholder character cannot be used in a range expression but can be used in a list. 2. A placeholder character can be used only with equal (=) or not-equal (<>) comparison operators. 3. A placeholder character can be used only in a CLID intrinsic of the WILDCLID data type. 4. Up to 32 placeholders are allowed in a single CLID digit string. 5. Only variables of the WILDCLID data type will allow the placeholder character to be entered as a value. Any combination of 1 wildcard and up to 31 placeholders are allowed in a single CLID digit string.

Table 8
Summary of valid items and sets

Data type	Class Allowed	Valid Value	Example
ACD	Item only	2- to 7-digit number	4100
CLID	Item or Set	1- to 32-digit number	4159408914; 415..515
DAY	Item or Set	Monday–Sunday	Sunday, Monday
DN	Item only	1- to 24-digit number	8914
DNIS	Item or Set	1- to 31-digit number	4587; 1234567, 7891234
Meridian 1 X11 Release 24 is required to support more than 7 DNIS digits.			
INTEGER	Item only	Number 0–32767	5000
LOC	Item or Set	3-digit number	655; 512, 444, 345
MUSIC	Item only	Number 0–511	213
NPA	Item or Set	3-digit number	415; 512, 444, 345
NPANXX	Item or Set	6-digit number	415940; 524133..524199
NXX	Item or Set	3-digit number	940; 512..598
PRIORITY	Item only	1,2,3, 4, 5, 6, 7, or 8	2
RAN	Item only	Number 0–511	9
SECONDS	Item only	Number 0–32767	10
TIME	Item or Set	0:00–23:59	11:30; 01:30..06:30
TREATMENT	Item only	1- to 7-digit number	533
WILDCLID	Item or Set	1- to 32-digit number with wildcards and/or placeholders	44@; 234?; 44@, 45@; 234?, 456? @77; ??0, 9?1 ?456@

Creating variables

CAUTION

Before you begin creating variables, note that to be able to add ACD DNs to the Variable Table, you must be connected to the Meridian 1 system—with the link enabled—because CCR looks for valid ACD DNs.

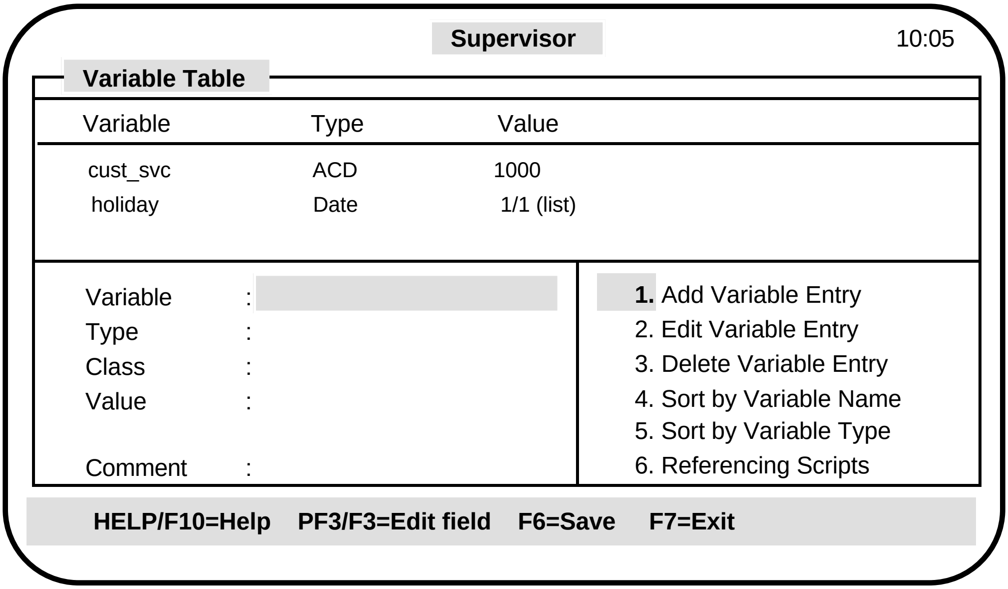
Note: If, before you save this variable, you notice you have made an error in entering the Type of Class, just correct the error by moving the highlight bar to the appropriate field and typing the correct value. If, however, you notice the error after you have saved the variable, you must delete the variable and start again.

To create one or more variables, complete the following steps:

1 Select the Check/Change Variables Table option from the Main Menu. The Variable Table screen appears, showing all existing variables. The Add Variable Entry selection is highlighted in the menu pane.

2 Select the Add Variable Entry option.

Figure 25
Add Variable Entry screen



Supervisor 10:05

Variable Table

Variable	Type	Value
cust_svc	ACD	1000
holiday	Date	1/1 (list)

Variable :
Type :
Class :
Value :

Comment :

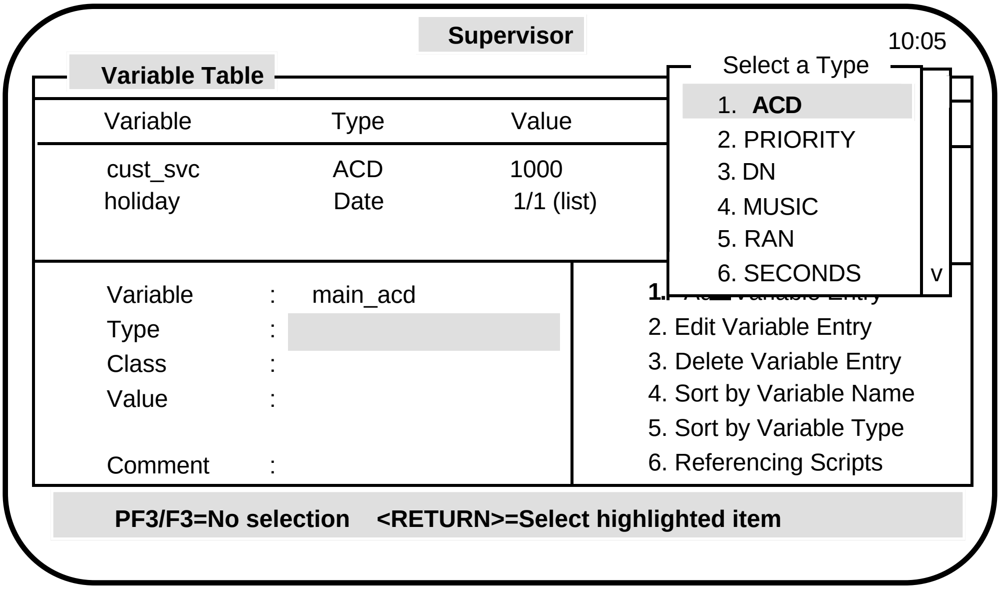
The edit pane appears, with the Variable field highlighted (Figure 25).

3 Type the variable name and press Return.

The variable name can have up to 32 alphanumeric characters. The first character must be a letter. Key words for CCR cannot be used. (See “Key words” at the end of this user guide.)

Pressing Return moves the highlight bar to the next item.

Figure 26
Select a Type pop-up menu



Supervisor 10:05

Variable Table

Variable	Type	Value
cust_svc	ACD	1000
holiday	Date	1/1 (list)

Select a Type

1. **ACD**
2. PRIORITY
3. DN
4. MUSIC
5. RAN
6. SECONDS

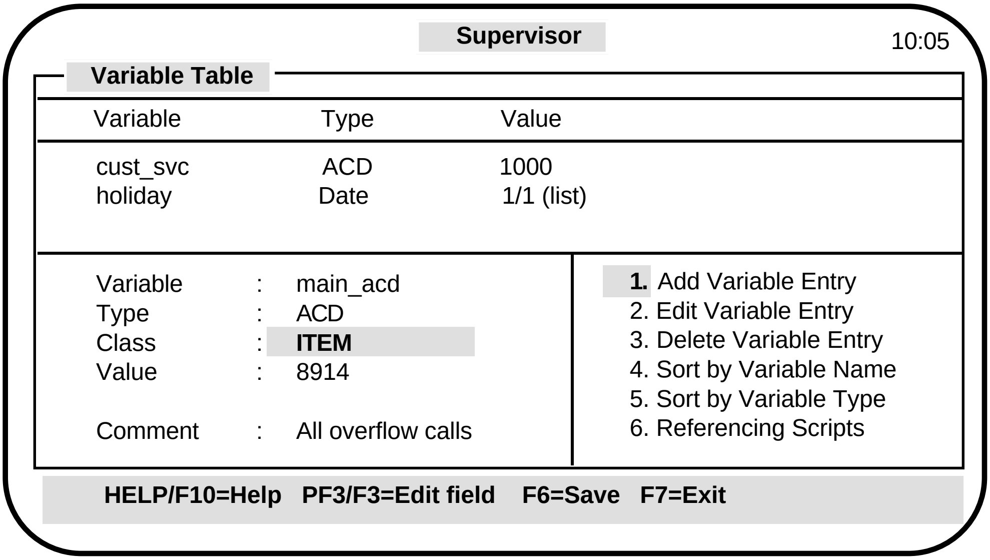
Variable : main_acd
Type :
Class :
Value :
Comment :

1. Add Variable Entry
2. Edit Variable Entry
3. Delete Variable Entry
4. Sort by Variable Name
5. Sort by Variable Type
6. Referencing Scripts

PF3/F3=No selection <RETURN>=Select highlighted item

4 Type a valid data type for the variable and press Return. Or press PF3/F3 to display a list of valid types (illustrated in Figure 26), and then select the type you want.

Figure 27
Select a Class screen



```
                         Supervisor                              10:05
 Variable Table
   Variable              Type             Value

   cust_svc              ACD              1000
   holiday               Date             1/1 (list)

   Variable      :   main_acd                    1. Add Variable Entry
   Type          :   ACD                         2. Edit Variable Entry
   Class         :   ITEM                        3. Delete Variable Entry
   Value         :   8914                        4. Sort by Variable Name
                                                 5. Sort by Variable Type
   Comment       :   All overflow calls          6. Referencing Scripts

   HELP/F10=Help  PF3/F3=Edit field   F6=Save  F7=Exit
```

5 Type your choice for class, either item or set (Figure 27), then press Return. Or press PF3/F3 to see a pop-up menu with your choices. See the chart at the beginning of this chapter for type and class correlation.

If the type has only one class, the class is automatically entered.

If you select Set, see the “Entering a list” procedure for instructions on how to enter a value.

6 Type a value for the variable and press Return. The highlight bar automatically moves to the Comment field.

Note: The chart at the beginning of this chapter lists valid values.

7 Type information that explains your variable and its use in the Comment field, then press Return. This field is optional, but it can be very helpful at a later date.

8 To save the variable you have just created, press F6.

You have just added a variable to the Variable Table. The command screen reappears, with the Add Variable Entry option highlighted. Repeat these steps until your Variable Table is complete.

9 After entering and saving all of your variables in a session, you must press F6 again to activate the additions you have just made to the Variable Table. See “Activating your variables” further on in this chapter.

Variables appear in the Variable Table in the order in which they are entered. To sort them, use either the Sort by Variable Name or Sort by Variable Type option.

Entering a set

A set is a list of values, a range of values, or a combination of items and ranges.

List A list defines non-consecutive values such as Monday, Wednesday, and Friday. A pop-up menu allows you enter up to 60 values in a list. You must enter each value in the list separately.

Range A range defines a series of consecutive values, with the beginning and end values of the range joined by two periods. For example, a range might be Monday through Wednesday. “Monday..Wednesday” specifies Monday through Wednesday. A set can contain more than one range—for example, “Monday..Wednesday” and “Friday..Sunday”.

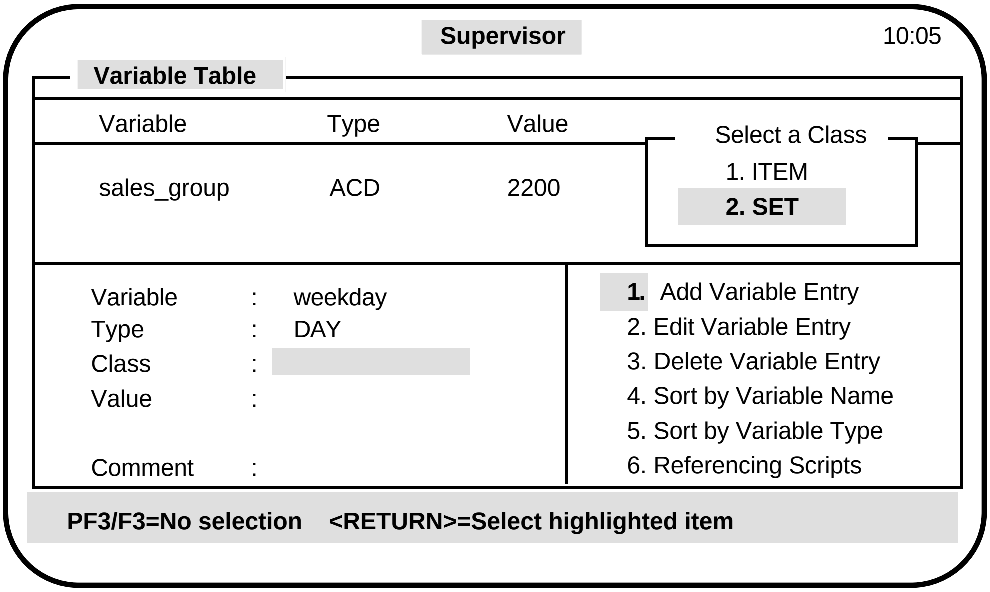
Entering a list

The following procedure shows how to enter a set as a list using the variable weekday as an example.

1. Select the Check/Change Variables Table option from the Main Menu. The Variable Table screen appears, showing all the existing variables.
2. Select the Add Variable Entry option from the menu pane.
3. Type **weekday** in the Variable field and press Return.
4. Press PF3/F3 to display a list of valid types.
5. Select DAY.

   The highlight bar automatically moves to the Class field.

Figure 28
Select a Class pop-up menu



Supervisor 10:05

Variable Table

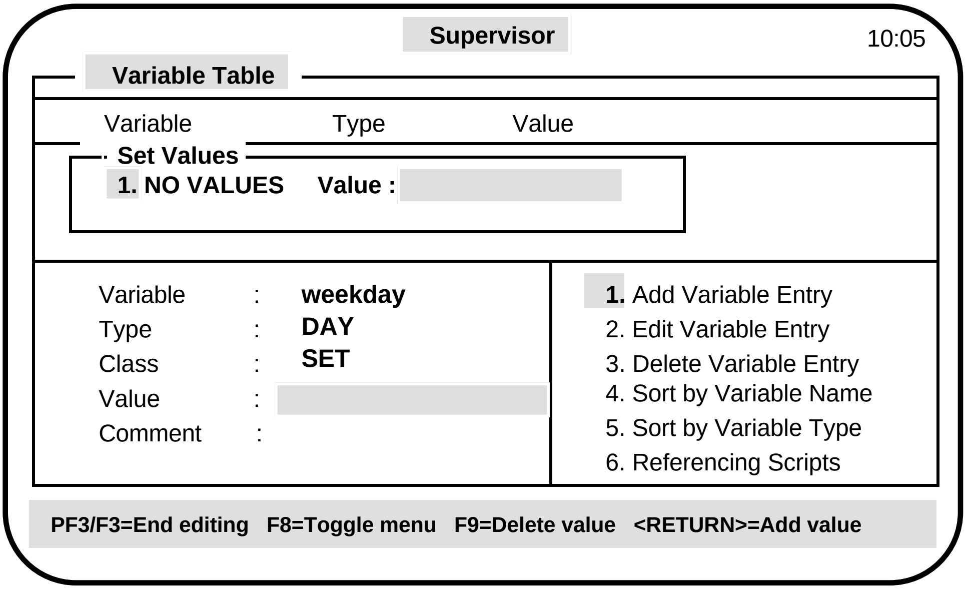
Variable	Type	Value
sales_group	ACD	2200

Variable : weekday
Type : DAY
Class :
Value :

Comment :

6. Press PF3/F3 again to display a list of valid class types.
7. Select SET (Figure 28).

Figure 29
Set Values pop-up menu



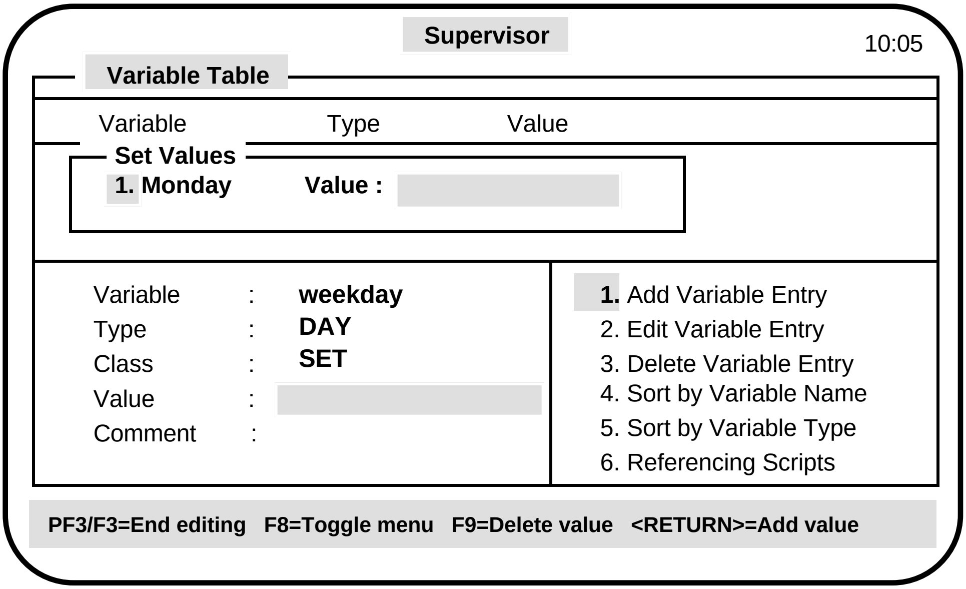
```
                              Supervisor                              10:05
  Variable Table
    Variable            Type          Value
   Set Values
    1. NO VALUES    Value :

    Variable      :   weekday                  1. Add Variable Entry
    Type          :   DAY                      2. Edit Variable Entry
    Class         :   SET                      3. Delete Variable Entry
    Value         :                            4. Sort by Variable Name
    Comment       :                            5. Sort by Variable Type
                                               6. Referencing Scripts

  PF3/F3=End editing   F8=Toggle menu   F9=Delete value   <RETURN>=Add value
```

The highlight bar automatically moves to the Value field.

8 Press PF3/F3 yet again to display a list of existing values, illustrated in Figure 29. If there are no items or ranges yet listed, the words “NO VALUES” appear.

9 Press Return again to begin entering list items (Figure 29).

Figure 30
Set a value



1. Monday   Value :

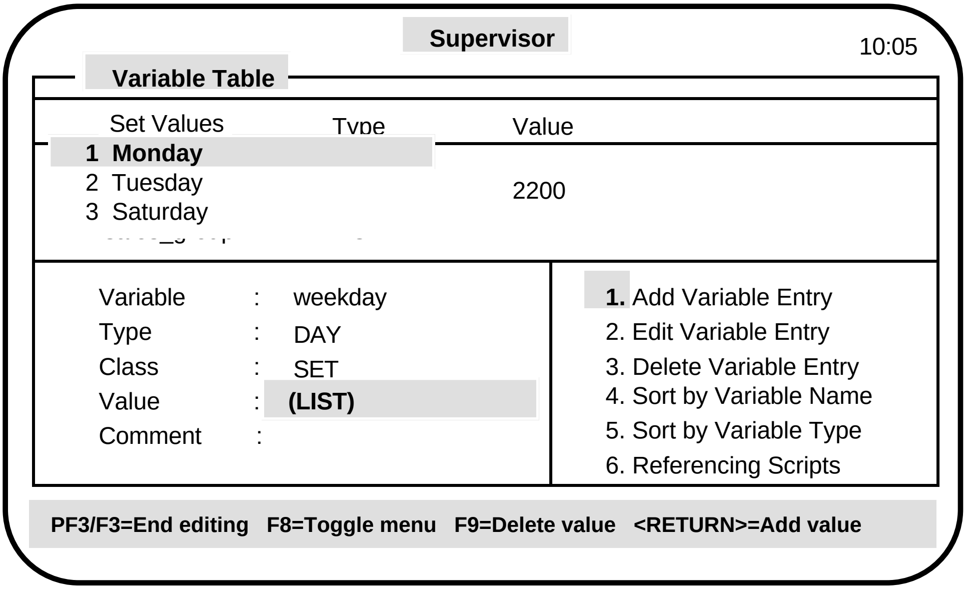
Variable : weekday
Type : DAY
Class : SET
Value :
Comment :

10 Type the first entry in the list (for example, Monday). It can be an item (single value) or a range (see “Entering a range” later in this chapter). Press Return to enter the value.

11 Press Return again to continue entering values (Figure 30).

12 Type in the next list entry. Entries do not have to be consecutive.

Figure 31
List of Values



Supervisor 10:05

Variable Table

Set Values Type Value

1 Monday
2 Tuesday
3 Saturday

2200

Variable	:	weekday
Type	:	DAY
Class	:	SET
Value	:	**(LIST)**
Comment	:	

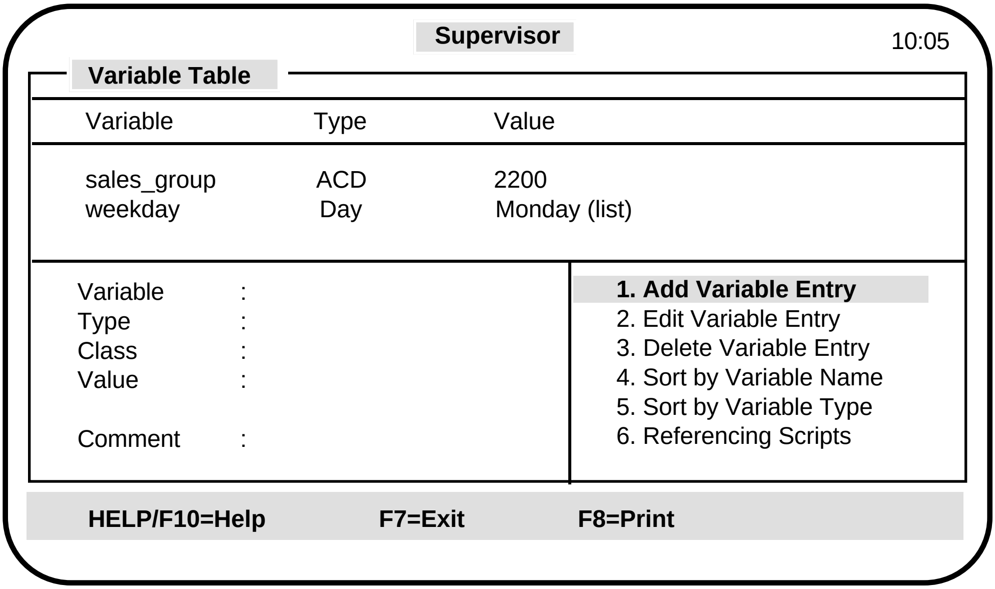
1. Add Variable Entry
2. Edit Variable Entry
3. Delete Variable Entry
4. Sort by Variable Name
5. Sort by Variable Type
6. Referencing Scripts

PF3/F3=End editing F8=Toggle menu F9=Delete value <RETURN>=Add value

13 Repeat steps 6 and 7 for the entire list. The pop-up screen supports up to 60 separate items on the list.

14 Press PF3/F3 when all the list items are entered. The Value field shows that a list of values has been entered (Figure 31).

Figure 32
List of Values window



Variable	Type	Value
sales_group	ACD	2200
weekday	Day	Monday (list)

Variable :
Type :
Class :
Value :

Comment :

If more than one value has been entered for a variable, the word “list” appears in any Value field associated with that variable.

For example, Figure 32 shows the designation “list” following the value associated with the variable “weekday.” If only a single value is entered for the variable “weekday,” the word “list” does not appear.

Entering a range

The following procedure shows an example of how to enter a set as a range using the variable weekday.

1 Select the Check/Change Variables Table option from the Main Menu. The Variable Table screen appears, showing all the existing variables.

2 Select the Add Variable Entry option from the Variable Table screen menu pane.

3 Type **weekday** and press Return.

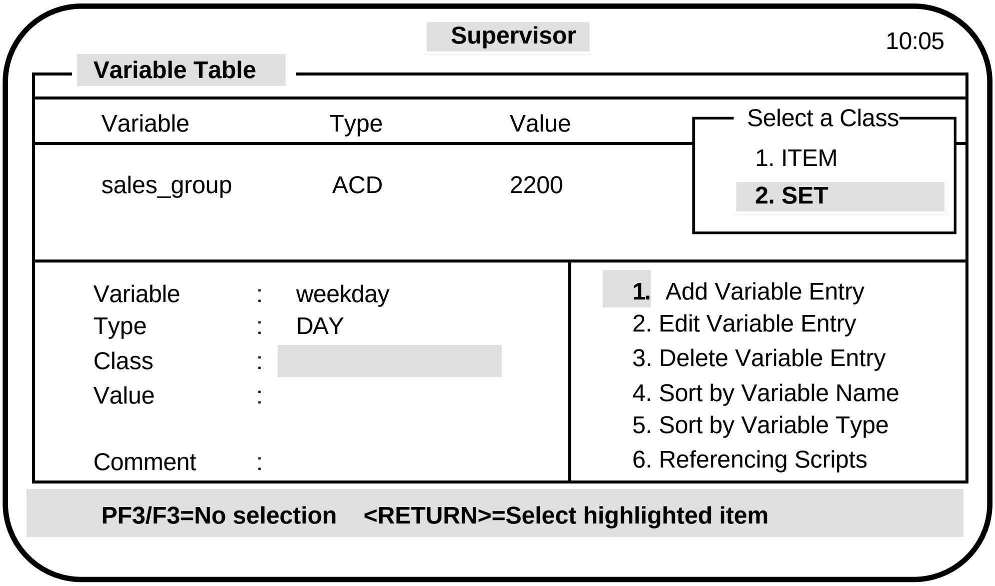
4 Press PF3/F3 to display a list of types.

A list of valid types appears.

5 Select DAY and press Return**.**

6 Press PF3/F3 again to display a list of valid class types.

Figure 33
Select a Class pop-up menu



Supervisor 10:05

Variable Table

Variable	Type	Value
sales_group	ACD	2200

Select a Class
1. ITEM
2. SET

Variable : weekday
Type : DAY
Class :
Value :

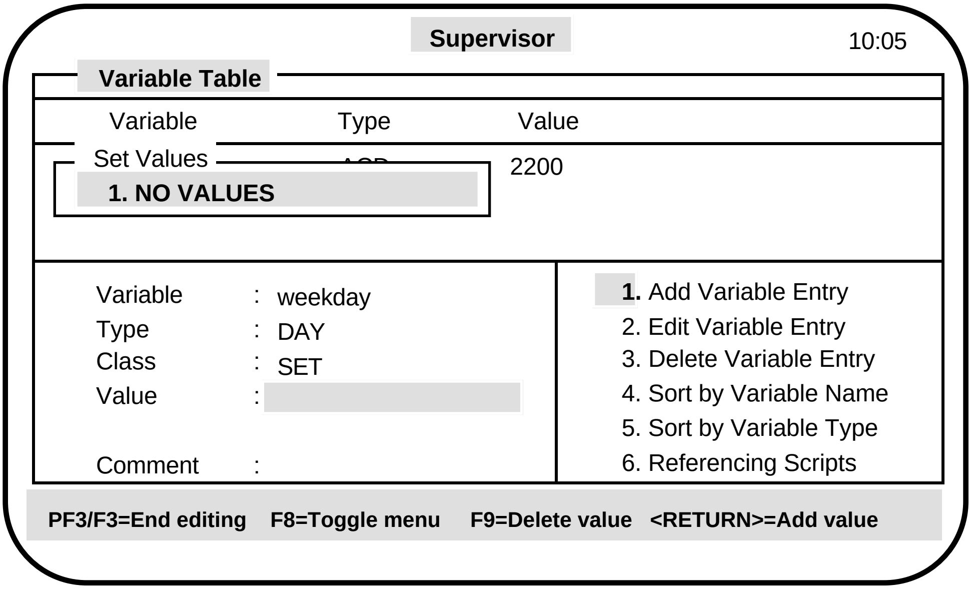
Comment :

1. Add Variable Entry
2. Edit Variable Entry
3. Delete Variable Entry
4. Sort by Variable Name
5. Sort by Variable Type
6. Referencing Scripts

PF3/F3=No selection <RETURN>=Select highlighted item

7 Select SET (Figure 33).

Figure 34
NO VALUES screen



```
                              Supervisor                   10:05
 Variable Table
     Variable              Type             Value
  Set Values               ACD             2200
    1. NO VALUES

    Variable     :  weekday          1. Add Variable Entry
    Type         :  DAY              2. Edit Variable Entry
    Class        :  SET              3. Delete Variable Entry
    Value        :                   4. Sort by Variable Name
                                     5. Sort by Variable Type
    Comment      :                   6. Referencing Scripts

 PF3/F3=End editing   F8=Toggle menu   F9=Delete value   <RETURN>=Add value
```

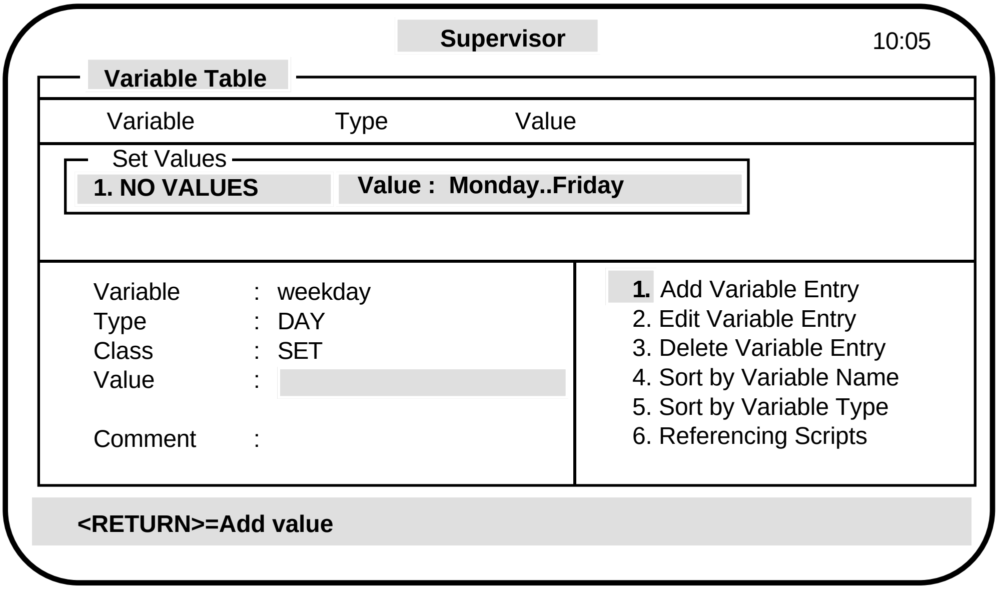
8 Press PF3/F3 again to see the existing values. If there are no items or ranges yet listed, the words “NO VALUES” appear (Figure 34).

9 Press Return to enter your range.

10 Type **Monday..Friday** and press Return. The variable “weekday” now includes Monday, Tuesday, Wednesday, Thursday, and Friday.

Everything specified within the range listed is included in the variable. The pop-up menu supports entries of as many as 32 characters. No more than 15 digits can be entered as part of a range. Specify these values as a list.

Figure 35
Screen with range entered

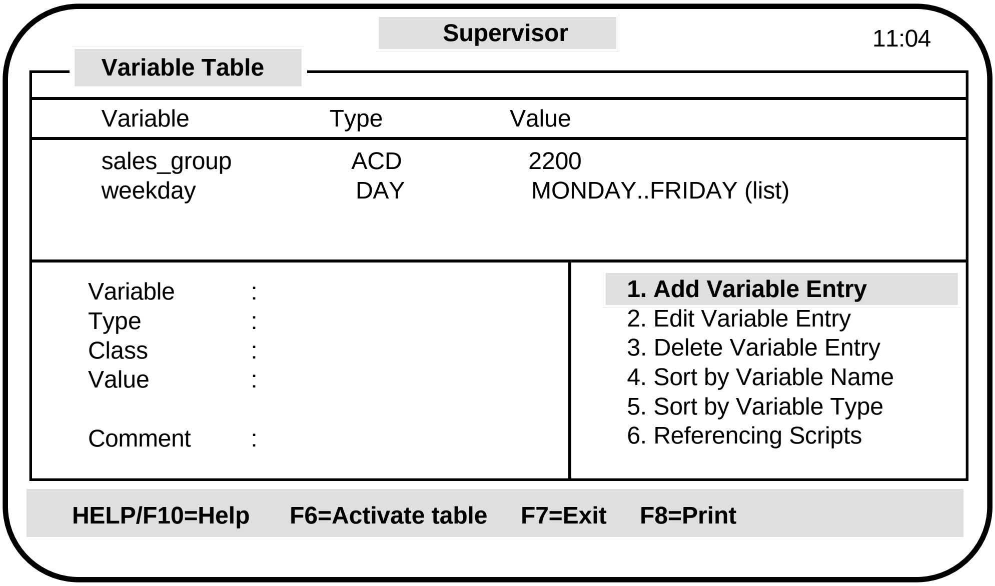


```
                         Supervisor                    10:05
 Variable Table
     Variable            Type             Value
   Set Values
  1. NO VALUES       Value :  Monday..Friday

   Variable     :  weekday               1. Add Variable Entry
   Type         :  DAY                   2. Edit Variable Entry
   Class        :  SET                   3. Delete Variable Entry
   Value        :                        4. Sort by Variable Name
                                         5. Sort by Variable Type
   Comment      :                        6. Referencing Scripts

 <RETURN>=Add value
```

11 To continue entering values, press Return.

12 To end editing, press PF3/F3.

Figure 36
Range of Values window



Supervisor 11:04

Variable Table

Variable	Type	Value
sales_group	ACD	2200
weekday	DAY	MONDAY..FRIDAY (list)

Variable :
Type :
Class :
Value :

Comment :

13 Press F6 to save the variable.

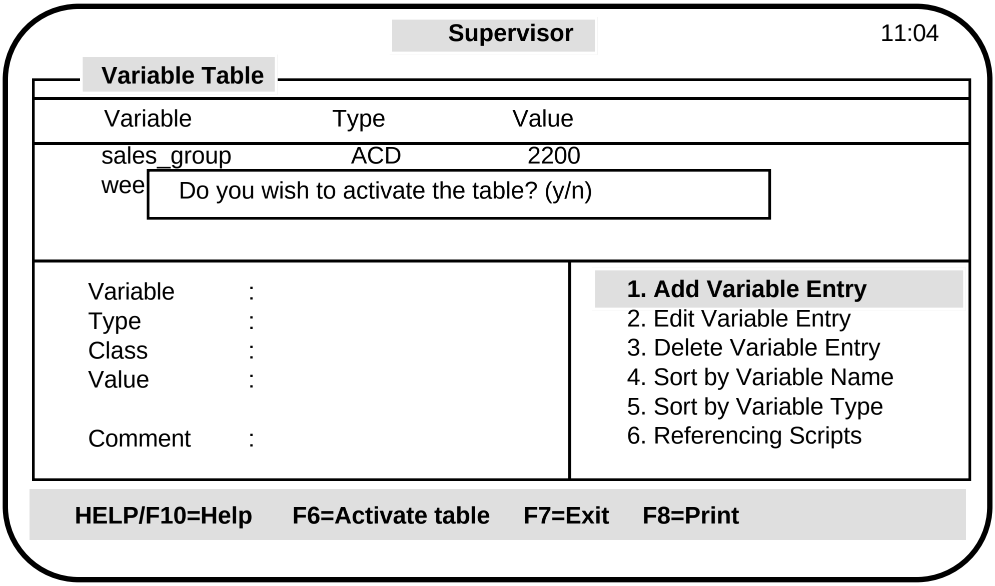
14 Press F6 again to activate the Variable Table if you want the variables to take effect immediately. Figure 36 illustrates the range entered for the example above.

Note: If you do not activate the Variable Table after defining a variable, the variable you have just entered will be erased from memory and will not be included in the Variable Table.

Activating a variable

After you are finished working with variables, you *must* activate them for the variables to become immediately effective in the system. If you do not, your entries will be removed from memory and will no longer appear in the Variable Table.

Figure 37
Activate table window



Supervisor 11:04

Variable Table

Variable	Type	Value
sales_group	ACD	2200
wee		

Do you wish to activate the table? (y/n)

Variable :
Type :
Class :
Value :

Comment :

To activate a variable that you have just created:

1 Press F6 after you have saved the variable. If you forget, and try to exit without activating the table, the system will prompt you (Figure 37).

2 Type **y** and press Return. The Main Menu appears.

Editing an existing variable

The Variable, Value, and Comment fields of a variable can be edited. Editing the variable field makes a copy of the variable under a new name. To change the Type or Class field, delete the variable, and re-enter it with the new type or class specified.

When you edit a variable, and activate it, the changes immediately affect any script where that variable is referenced for *new* calls being handled by the script. Before you start editing, you may want to review script references to the variables you want to change. (See “Referencing scripts” at the end of this chapter.)

> ***Note:*** Changing the variable name is equivalent to creating a new variable. The old variable still exists.

To change values for an existing variable, follow the procedures below.

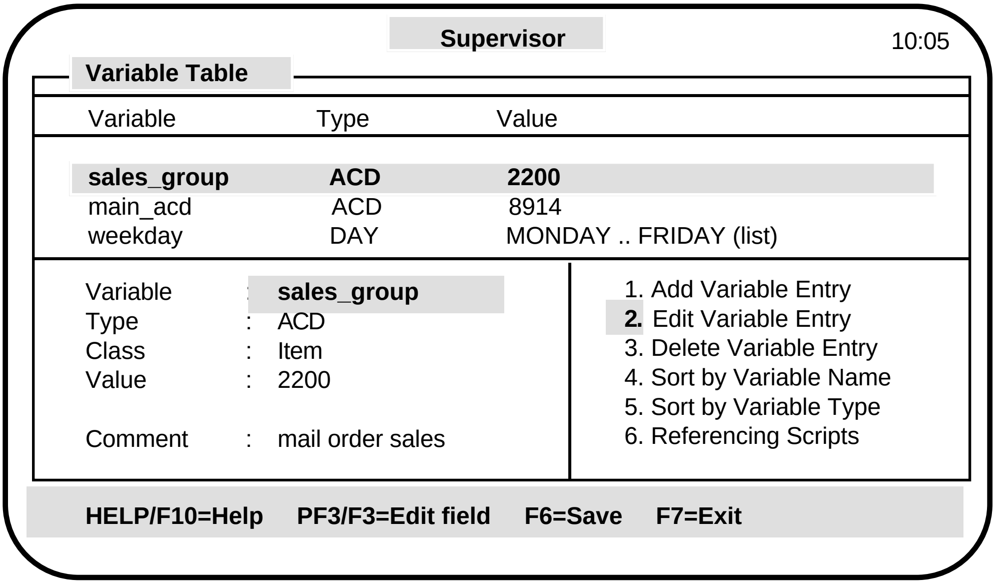
1 Select the Check/Change Variables Table option from the Main Menu. The Variable Table screen appears, showing all the existing variables.

2 Select the Edit Variable Entry option. You are automatically placed in the Variable Table screen, with the first entry highlighted.

3 Scroll through the list (using the up and down arrow keys) until the variable you want to edit is highlighted.

4 Press Return to select the variable.

Figure 38
Edit Variable Entry screen



Variable	Type	Value
sales_group	**ACD**	**2200**
main_acd	ACD	8914
weekday	DAY	MONDAY .. FRIDAY (list)

Variable : **sales_group**
Type : ACD
Class : Item
Value : 2200

Comment : mail order sales

The existing Type, Class, Value and Comment values are displayed on the Edit Variable screen, with the Variable field highlighted (Figure 38).

5 Move the highlight bar to the area you want to change, make the appropriate changes, and press Return.

6 Press F6 to save the variable. The new entry is placed at the top of the Variable Table.

7 To activate the Variable Table and make the variables you have just saved are immediately effective in CCR, press F6 again.

Deleting a variable

No script can reference a variable that has been deleted from the Variable Table. A deletion takes effect as soon as you activate the Variable Table. If you try to remove a variable in an installed or active script, a warning message appears. Before deleting a variable, use the Referencing Scripts option (described later in this chapter) to display a list of all scripts referencing it.

To delete an existing entry:

1 Select Check/Change Variables Table from the Main Menu. The Variable Table screen appears, showing all existing variables.

2 Select the Delete Variable Entry option from the menu pane.

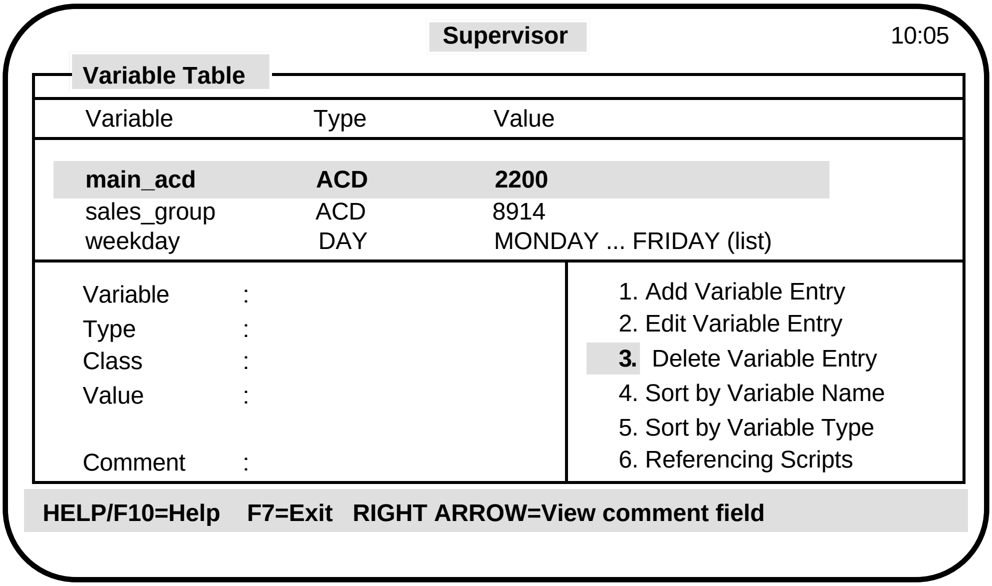
3 Scroll through the list (using the up and down arrow keys) until the variable you want to delete is highlighted.

4 Press Return to select the variable for deletion.

The selected variable is removed from the list. You are now back in the menu pane.

5 Press F6 to activate the table and complete the Delete function.

Figure 39
Delete Variable Entry window



```
                              Supervisor                          10:05
  Variable Table
    Variable            Type           Value

    main_acd            ACD            2200
    sales_group         ACD            8914
    weekday             DAY            MONDAY ... FRIDAY (list)

    Variable       :                          1. Add Variable Entry
    Type           :                          2. Edit Variable Entry
    Class          :                          3. Delete Variable Entry
    Value          :                          4. Sort by Variable Name
                                              5. Sort by Variable Type
    Comment        :                          6. Referencing Scripts

  HELP/F10=Help   F7=Exit   RIGHT ARROW=View comment field
```

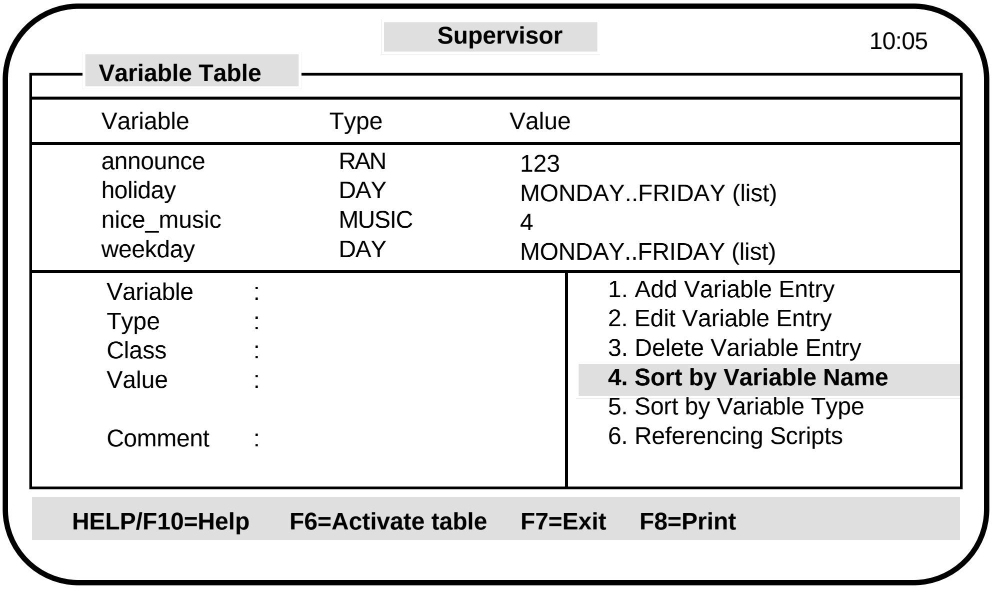
Sorting by variable name and type

Use the Sort by Variable Name and Sort by Variable Type options to reorganize how variables appear in the Variable Table. The following figures show which option to highlight when performing the sort functions.

Sorting by name

This sorts variables in alphabetical order by variable name.

Figure 40
Sort by Variable Name screen

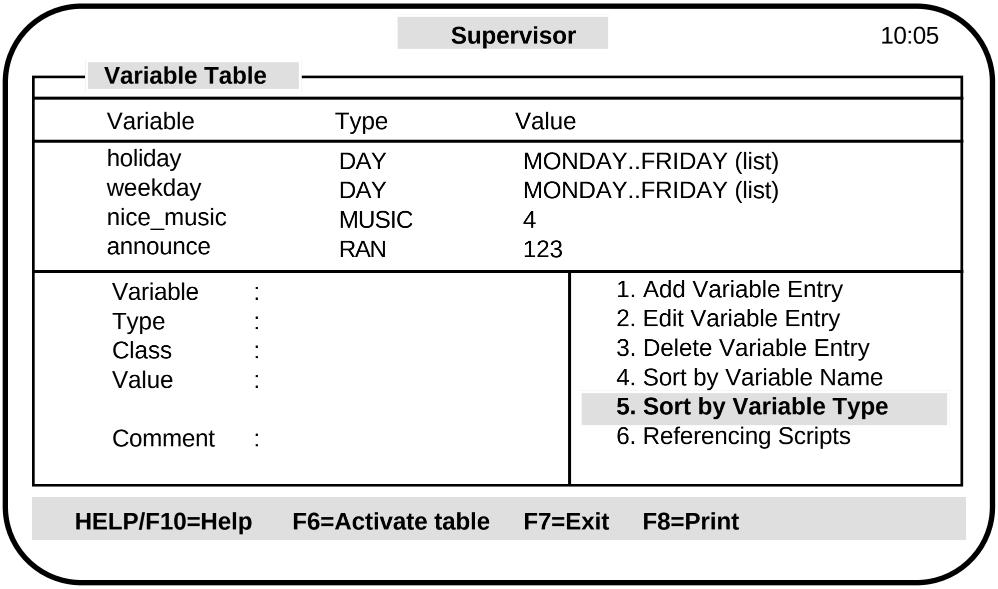


```
                         Supervisor                          10:05
 Variable Table

   Variable            Type          Value
   announce            RAN           123
   holiday             DAY           MONDAY..FRIDAY (list)
   nice_music          MUSIC         4
   weekday             DAY           MONDAY..FRIDAY (list)

   Variable     :                       1. Add Variable Entry
   Type         :                       2. Edit Variable Entry
   Class        :                       3. Delete Variable Entry
   Value        :                       4. Sort by Variable Name
                                        5. Sort by Variable Type
   Comment      :                       6. Referencing Scripts

 HELP/F10=Help   F6=Activate table   F7=Exit   F8=Print
```

Sorting by type

This sorts variables in the order in which the types appear in the pop-up window used to define the variables. (See Figure 26 in the “Creating Variables” section of this chapter).

Figure 41
Sort by Variable Type screen



Variable	Type	Value
holiday	DAY	MONDAY..FRIDAY (list)
weekday	DAY	MONDAY..FRIDAY (list)
nice_music	MUSIC	4
announce	RAN	123

Referencing scripts

Use this option when you want to display a list of scripts using a particular variable. This is helpful before you edit or delete a variable because you will be able to see the scripts that could be affected by your changes.

Before a script can be referenced, it must be created, validated, and installed. For more information, see the “Writing scripts” chapter.

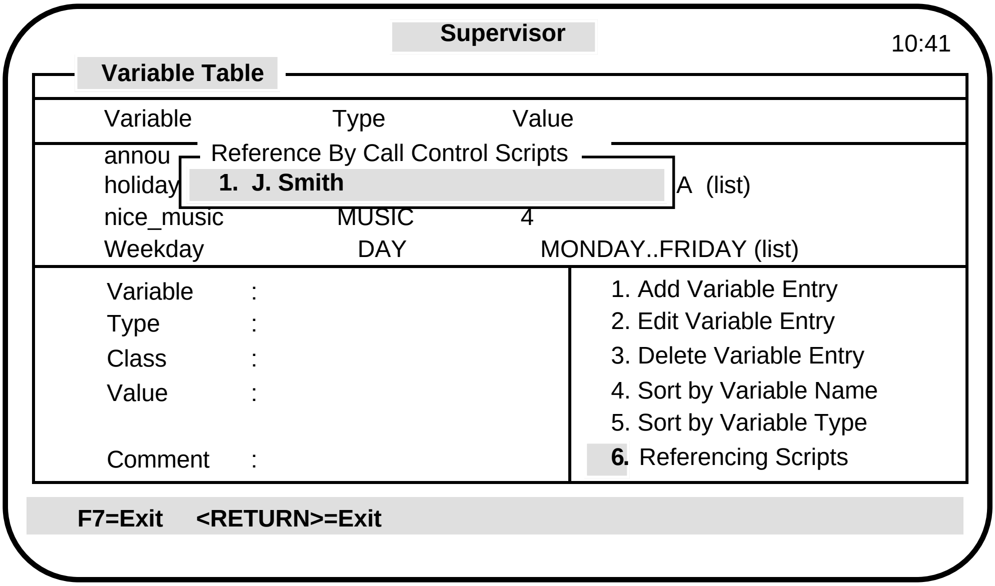
To display a list of scripts referencing a variable, follow these steps:

1 Select Check/Change Variables Table from the Main Menu. The Variable Table screen appears, showing all existing variables.

2 Select the Referencing Scripts option. A list of variables appears.

3 Select the variable name. A list of installed scripts referencing that variable appears.

Figure 42
Referencing Scripts screen



```
                              Supervisor                      10:41
 Variable Table
     Variable            Type           Value
     annou  Reference By Call Control Scripts
     holiday  1.  J. Smith                        A  (list)
     nice_music          MUSIC          4
     Weekday              DAY           MONDAY..FRIDAY (list)
     Variable      :                      1. Add Variable Entry
     Type          :                      2. Edit Variable Entry
     Class         :                      3. Delete Variable Entry
     Value         :                      4. Sort by Variable Name
                                          5. Sort by Variable Type
     Comment       :                      6. Referencing Scripts
   F7=Exit    <RETURN>=Exit
```

Script commands

CAUTION

Unlike calls being processed directly by ACD DNs, calls being handled by CCR have no night treatment defined *unless it is specified by a script.* For this reason, it is always recommended that scripts start with a check for Night Service ACD DN.

This chapter describes how each script command works. Commands are listed in alphabetical order within categories. Note that the “Writing scripts” chapter of this guide includes a “Script writing tips” section.

For information on how the structure of scripts can affect call processing efficiency, refer to the “Tips to assist CCR engineering” section in the “Engineering the CCR system” chapter of the *Meridian Link/Customer Controlled Routing Engineering Guide* (NTP 553-3203-151).

The four basic categories of CCR script language commands are

- call routing commands
- call control commands
- call treatment commands
- script processing commands

Call routing commands may also provide treatment. Table 9 describes call routing commands.

Table 9
Overview of call routing script commands

Command	Description
QUEUE TO	Queues a call to a specified ACD DN. Provides ringback treatment.
REMOVE FROM	Removes a call from an ACD DN queue.
ROUTE TO	Routes a call to any dialed number either on or off the switch.

Table 10 describes call control commands.

Table 10
Overview of call control script commands

Command	Description
QUEUE TO	Queues a call to a specified ACD DN. Provides ringback treatment.
ROUTE TO	Routes a call to any dialed number either on or off the switch.
FORCE BUSY	Gives a busy signal to the caller.
FORCE DISCONNECT	Terminates a call.

Table 11 describes call treatment commands.

Table 11
Overview of call treatment script commands

Command	Description
GIVE IVR	Gives IVR treatment to the call.
GIVE MUSIC	Plays music to the caller.
GIVE RAN	Gives a RAN to the call.
GIVE RINGBACK	Gives the call ringback tone.
GIVE SILENCE	Turns off MUSIC, RINGBACK.
QUEUE TO	Queues a call to a specified ACD DN. Provides ringback treatment.
ROUTE TO	Routes a call to any dialed number either on or off the switch.

Table 12 describes script processing commands.

Table 12
Overview of script processing commands

Command	Description
GOTO	Command that directs the script to a SECTION.
IF	Lets you conditionally execute commands.
QUIT	Ends execution of commands.
SECTION	Labels the target of a GOTO command.
WAIT	Controls the timing of the execution of commands.

Table 13 shows the information required to use each command, and the data type of the object of the command.

Table 13
Information required for script commands

Command	Required Information	Data Type	Options
QUEUE TO	ACD DN	ACD	priority
REMOVE FROM	ACD DN	ACD	
ROUTE TO	DN	ACD DN or CDN	
FORCE BUSY			
FORCE DISCONNECT			
GIVE IVR	ACD DN	ACD DN	priority or treatment
GIVE MUSIC	music route	music	
GIVE RAN	RAN route	RAN	
GIVE RINGBACK			
GIVE SILENCE			
GOTO	label		
IF			
QUIT			
SECTION	label		
WAIT	wait time	seconds	

The following sections describe each command and provide examples of how they are used. Commands are grouped into the three basic categories, and are listed in alphabetical order within each category.

Call routing commands

Information listed in “< >” directly under the command is the information required for that command. The information in “{}” is additional optional information.

Note that QUEUE TO and ROUTE TO, described in this section, are also call control and call treatment commands.

QUEUE TO

<acd dn> {WITH PRIORITY [1, 2, 3, or 4]}

This command enables you to queue calls to the specified ACD DN. You may specify a priority of 1 through 4, with 1 being the highest priority. If you specify no priority, the call defaults to priority 1. If a call is queued twice to the same queue with the same priority, the second command is ignored.

You may queue any given call to no more than eight queues simultaneously by repeating the QUEUE TO command with different ACD DNs. Different priorities can be specified for each QUEUE TO command (see Example 3). If a script attempts to queue a call to more than eight queues simultaneously, CCR executes the first eight QUEUE TOs and ignores the others.

> ***Note:*** When a large number of commands incurring no delay starts a script, the resulting message burst can interfere with system performance. Examples 4 and 5 illustrate ways to limit the number of ACD DNs on which the CCR application is drawing statistics, when using multiple QUEUE TO commands.

CCR executes, in order, all script commands that immediately follow a QUEUE TO—unless the call has been answered or abandoned. Script processing ends when a call has been answered or abandoned.

> ***Note:*** If QUEUE TO (rather than GIVE IVR) is used to queue a call to an IVR queue, the call is considered answered when the IVR agent answers the call, which ends script processing for the call.

If the designated queue is in Night Service, or if the DN specified is not an ACD DN, the QUEUE TO command fails. Check for Night Service to ensure that you are not attempting to queue calls to ACD DNs that are in Night Service.

As the following examples show, the command can be repeated to change the priority of a previously queued call. When an agent becomes free, the call is presented to the agent, is automatically given ringback, and is removed from all other queues. If a call has heard no previous tone, it will automatically hear ringback when a QUEUE TO is executed.

Example 1

```
QUEUE TO 3600 WITH PRIORITY 2

WAIT 10

QUEUE TO 3600 WITH PRIORITY 1
```

This script queues calls to ACD DN 3600 at priority level 2. If an agent is not available to answer the call, the script continues execution. Because no treatment (for example, ringback or music) has been specified for the call, it receives the default ringback treatment until it is presented to an agent in ACD DN 3600—unless the Call Force option is turned on for that ACD DN. After 10 seconds, the call is changed to priority level 1, where it remains until an agent is available or it is abandoned.

Example 2

```
QUEUE TO 3600 WITH PRIORITY 2

WAIT 6          /* Ensures at least a cycle of ringback if
                the call is not answered immediately */

GIVE RAN 12

GIVE MUSIC 24

WAIT 10

QUEUE TO 3600
```

This script queues calls to ACD DN 3600 at priority level 2. If an agent is not available, the call first hears six seconds of ringback (the default treatment following a QUEUE TO command), followed by a RAN message, and then music from music route 24. After 10 seconds of music, the call is changed to the default priority level 1. It continues to receive music until an agent is available.

Example 3

In high traffic situations, this script can be inefficient.

```
QUEUE TO 3600 WITH PRIORITY 3

QUEUE TO 3601 WITH PRIORITY 4

QUEUE TO 3602 WITH PRIORITY 4

WAIT 10

QUEUE TO 3600 WITH PRIORITY 2

QUEUE TO 3603 WITH PRIORITY 1
```

This script queues calls to ACD DN 3600 as priority level 3. It will also queue calls to ACD DN 3601 and to ACD DN 3602, with a priority level 4. If, after 10 seconds, an agent is not available in any of these queues, its priority in 3600 is changed to priority 2 and the call is queued at 3603 with a priority level 1. The call remains in all queues until it is answered or abandoned.

The script illustrated in Example 3 generates a burst of messaging for each incoming call because

- CCR executes the second and third QUEUE TO commands before the Meridian 1 has had a chance to find an Idle Agent for the first QUEUE TO
- CCR requests statistics from the Meridian 1 on all ACD DNs specified in commands

Example 4 shows a way to delay execution of subsequent QUEUE TO commands until the Meridian 1 determines whether they are required.

Example 4

This script prevents CCR from requesting unnecessary statistics from the Meridian 1 by imposing artificial delays between QUEUE TO commands.

```
QUEUE TO 3600 WITH PRIORITY 3

WAIT 2

QUEUE TO 3601 WITH PRIORITY 4

WAIT 2

QUEUE TO 3602 WITH PRIORITY 4

WAIT 10

QUEUE TO 3600 WITH PRIORITY 2

WAIT 2

QUEUE TO 3603 WITH PRIORITY 1
```

This script queues calls to ACD DN 3600 as priority level 3. If an agent is not available at ACD DN 3600, it queues calls to ACD DN 3601 with a priority level 4. If an agent is not available at ACD DN 3601, it queues calls to ACD DN 3602, with a priority level 4. If, after 10 seconds, an agent is not available in any of these queues, its priority in 3600 is changed to priority 2 and the call is queued at 3603 with a priority level 1. The call remains in all queues until it is answered or abandoned.

The above script avoids the burst of messaging for each incoming call by giving the Meridian 1 an opportunity to find an idle agent before the next QUEUE TO is executed. However, using WAIT statements imposes an artificial delay to call processing that may be unacceptable to the operation of the call center. Example 5 shows another way to avoid message bursting.

Example 5

This script prevents CCR from requesting unnecessary statistics from the Meridian 1 by setting conditions that force CCR to wait for the Meridian 1 message evaluating the status of the ACD DN referred to in the previous QUEUE TO statement.

```
QUEUE TO 3600 WITH PRIORITY 3

QUEUE TO 3601 WITH PRIORITY 4 IF (Idle Agents 3600 = 0)

QUEUE TO 3602 WITH PRIORITY 4 IF (Idle Agents 3601 = 0)

WAIT 10

QUEUE TO 3600 WITH PRIORITY 2

QUEUE TO 3603 WITH PRIORITY 1 IF (Idle Agents 3600 = 0)
```

The above script does the same thing that the script in Example 3 does, but avoids the burst of messaging for each incoming call by giving subsequent QUEUE TO only if the previous QUEUE TO was not executed.

CCR-NACD Interworking

With Meridian 1 X11 Release 22.46, the CCR-NACD Interworking feature was introduced. This feature provides greater control for calls that need to be placed in multiple queues throughout a private ISDN enterprise network.

With this enhancement, a call can be logically placed into many queues throughout the network using the Queue To command. Although the call has been placed in the network, the local CCR system still maintains control over it.

Once a remote target is reserved by NACD, a call will be removed from the source CDN queue and presented to the reserved target agent.

To use CCR-NACD Interworking, the NACD Routing DN must be configured with an NACD Night Table defined.

When using CCR-NACD Interworking, remember these points.

- A CDN call, which is controlled by CCR, can only be queued to one NACD Routing DN. The call can be logically queued to up to 20 remote target ACD DNs simultaneously through the NACD Night Table of an NACD Routing DN.
- The Queue To NACD call will be canceled from the NACD Routing DN in any of the following.

  a) caller abandons

  b) call is answered by a local agent

  c) call is reverted to default ACD DN if application or link is down

  d) call is given a Give Busy, Give Overflow, Force Disconnect, or Route To treatment.

REMOVE FROM

<acd dn>

This command lets you remove a call from an ACD DN queue. Use this command when you want to continue handling a call but no longer want it queued to the specified ACD DN. If you have simultaneously queued the call to eight queues, you can use this command to remove the call from one queue, enabling you to queue the call to a another queue.

The ROUTE TO, FORCE BUSY and FORCE DISCONNECT commands automatically remove calls from any queues they are in. The REMOVE FROM command is not needed when using these commands. CCR executes all commands in a script that immediately follow a REMOVE FROM command, unless the call is answered or abandoned.

If the Meridian 1 attempts to remove the call from a queue into which the call was never placed, the REMOVE FROM command fails.

Example 1

```
    REMOVE FROM 4500
```

This command removes a call from ACD DN 4500.

Example 2

```
    REMOVE FROM main_acd
```

This command removes a call from an ACD DN identified by the variable main_acd.

ROUTE TO

<dn or acd dn>

This command routes a call to any dialed number either on or off the switch. Specify the DN just as it would be dialed, including ESN or route access codes if necessary. The ROUTE TO command terminates the handling of the call by the script. The call is removed from any queues that it might be in. All further call treatment is handled by the Meridian 1 system. The calling party gets whatever treatment they would have received had the number been dialed directly.

If you use ROUTE TO to send a call to an ACD DN, CCR gives up control and the call will be controlled by the Meridian 1.

CAUTION

Commands in a script that follow a ROUTE TO command never execute unless reached by a GOTO command and a SECTION label.

The validation of a script with non-executable commands produces a warning message. See Example 2.

Example 1

```
QUEUE TO 3600 WITH PRIORITY 2

GIVE RAN 12

GIVE MUSIC 24

WAIT one_minute

ROUTE TO 5006
```

The script illustrated in Example 1 queues calls to ACD DN 3600 as priority level 2. If an agent is not available, the call may hear a burst of ringback before hearing the RAN from route 12 followed by music from route 24. If the call is not answered after 60 seconds (assuming that “one_minute” is defined as 60 seconds in the Variable Table) the call is routed to extension 5006, which automatically removes it from queue 3600. This completes script handling of the call.

Example 2

```
QUEUE TO 3600 WITH PRIORITY 2

GIVE RAN 12

WAIT one_minute

ROUTE TO 5006

GIVE MUSIC 24
```

The script illustrated in Example 2 is similar to that illustrated in Example 1, however “GIVE MUSIC 24” has been moved to the end of the script.

The GIVE MUSIC command will never execute because it appears after the ROUTE TO command. When the script is validated, a warning message appears, informing you of this situation.

Call control commands

Information listed in “< >” directly under the command is the information required for that command. The information in “{}” is additional optional information.

Note that QUEUE TO and ROUTE TO, described in the “Call routing commands” section of this chapter, are also call routing and call treatment commands.

FORCE BUSY

This command is the recommended way to terminate a call. When executed, it gives the caller a busy signal.

> ***Note:*** Calls on CO, FEX, or WATs trunks cannot be forced busy if the call is in an unanswered state. If a FORCE BUSY is executed for any of these calls, the Meridian 1 routes the call to the CDN’s default ACD DN.

Commands that follow a non-conditional FORCE BUSY command in a script never execute unless they are included in a SECTION reached by a GOTO command.

Validation of scripts with non-executable commands produces warning messages.

FORCE BUSY should not be used after treatment (for example, RAN, MUSIC, RINGBACK) has already been given. FORCE BUSY should be executed at the beginning of a script, not in the middle or at the end.

Example 1

```
FORCE BUSY IF Total Queued Calls 3600 > 100

QUEUE TO 3600 WITH PRIORITY 2

GIVE RAN 12

GIVE MUSIC 24
```

The Meridian 1 counts the number of calls queued to 3600. If there are more than 100 calls already in queue for ACD DN 3600, the incoming call is immediately given busy treatment; it is not queued and it cannot subsequently be answered. If there are not more than 100 calls queued to 3600, the call is queued to ACD DN 3600 as priority level 2. If an agent is not available, the caller may hear a burst of ringback before hearing the RAN followed by music.

Example 2

```
FORCE BUSY IF Total Queued Calls 3600 > Logged Agents
    3600 * 2
```

The Meridian 1 counts the number of calls queued to 3600 and the number of agents logged in to 3600. Calls are forced to busy if the total number of calls queued to 3600 is more than twice the number of agents logged into 3600. This allows the number of calls in queue to fluctuate with the number of agents logged in.

FORCE DISCONNECT

This command allows you to force disconnect a call. The FORCE DISCONNECT command would generally be used after a RAN that informs the caller that the call is terminated (see Example 1).

If FORCE DISCONNECT is executed for a call in an unanswered state, the command answers the call and immediately disconnects it.

If FORCE DISCONNECT is used unconditionally as the first command to treat a call, it will generate a script validation error message. Validation of scripts with non-executable commands produces warning messages (see Example 2).

Example 1

```
SECTION After_hours

   GIVE RAN service_closed

   /* Our office is closed. Please call again.  */

   FORCE DISCONNECT
```

Once the call gets to this point in the script, the call is given RAN and then disconnected.

Example 2

```
SECTION After_hours

    GIVE RAN service_closed
    /* Our office is closed. Please call again.  */

    FORCE DISCONNECT

    GIVE MUSIC 24
```

In the above example, the GIVE MUSIC command is never executed. When the script is validated, a warning message informs you of the problem.

Call treatment commands

Information listed in “< >” directly under the command is the information required for that command. The information in “{}” is additional optional information.

Note that QUEUE TO and ROUTE TO, described in the “Call routing commands” section of this chapter, are also routing and control commands.

GIVE IVR

GIVE [INTERRUPTIBLE] IVR <acd dn> [WITH PRIORITY (1, 2, 3, 4)]
[WITH TREATMENT <treatment value>]

This command requires Release 18 or later X11 software configured with the Hold in Queue for IVR package. If you attempt to use this command with Release 17 or if the X11 optional package is not present, you will see an OPTION NOT PRESENT error message from the compiler during script validation.

The GIVE IVR command temporarily turns control of the call over to Meridian Mail (or other IVR equipment) for Interactive Voice Response treatment. The call is placed in an IVR queue (specified by the ACD value, which must be configured in overlay 23 with IVR=YES) and is presented for IVR treatment when it reaches the top of the queue. The ACD value may be a variable reference (of type ACD) or an ACD DN.

Treatments (such as GIVE MUSIC and GIVE SILENCE) interrupted by the IVR treatment resume when IVR treatment is finished. Commands that follow GIVE IVR are not executed until the IVR treatment is finished.

If the IVR session transfers the call while it is also in one or more ACD queues, the call is presented to the specified destination, where it appears to CCR as an abandoned call. As an abandoned call, it is then released from any other queue and released from CCR’s control.

Use the optional GIVE INTERRUPTIBLE clause to allow IVR treatment to be interrupted by presentation of the call to an agent when the call was assigned to an agent queue before IVR was given (see Example 3, Example 6, and Example 8). If the GIVE INTERRUPTIBLE clause is not used, the IVR treatment cannot be interrupted by presentation to an agent.

Use the optional WITH PRIORITY clause to allow certain calls to receive IVR treatment ahead of others. Priority values are 1, 2, 3, or 4, with 1 being the highest (first served) priority. Priority values can be referenced as constants (the type being determined by the constant's use as a priority value) or as variables of type PRIORITY. If the clause is not used, the default priority of 1 is used (see Examples 1–3, and Example 5).

Use the optional WITH TREATMENT to specify the IVR treatment that Meridian Mail will use to control the call. The treatment value may be specified as a variable reference (of type TREATMENT) or as a constant. Treatment values are 1–7 digits long, with any digit being allowed in any position (see Example 2, and Examples 5–8).

Using WITH TREATMENT eliminates the need to use a dummy ACD queue for night call forwarding to the IVR queue.

If the WITH TREATMENT clause is used for third-party IVR, the clause has no effect.

If WITH TREATMENT clause is not used, the Meridian 1 will provide a default treatment value from the call's CALLED PARTY DN or the treatment defined for the IVR ACD DN. See Example 1, and Examples 3–4.

Treatments for Meridian Mail IVR may be defined in any of the following ways:

- CCR script commands, as described in this section
- the treatment defined for the IVR queue in overlay 23
- the treatment defined for the Called Party DN of the call

The two optional WITH clauses may be present in either order. See Examples 7 and 8.

Like RAN, IVR suspends processing of the script until IVR is complete. However, if the IVR is specified as INTERRUPTIBLE, IVR will be interrupted by presentation of the call to an agent.

Example 1

```
GIVE IVR 8923
```

This command gives the call the IVR treatment defined in Meridian Mail for the Called Party DN of the call.

Example 2

```
GIVE IVR 4567 WITH TREATMENT 9877 IF Oldest Call 760 > 30
```

This command gives IVR treatment defined in Meridian Mail as 9877 to ACD DN 4567 only if the oldest call at ACD DN 760 has waited longer than 30 seconds.

Example 3

```
GIVE INTERRUPTIBLE IVR 8905
```

This command allows the call to be presented to an agent before any IVR treatment is completed, assuming that the call is in an ACD queue.

Example 4

```
GIVE IVR ivr_queue WITH PRIORITY prior1
```

This command gives the call the IVR treatment in Meridian Mail defined in overlay 23 for the variable “ivr_queue” and gives a priority that is defined as the variable “prior1”.

Example 5

```
GIVE IVR menu_q WITH TREATMENT 1234567
```

This command specifies the IVR treatment in Meridian Mail, overriding any Meridian 1 IVR treatment defined for any ACD DN defined to the variable “menu_q.” The priority defaults to priority 1.

Example 6

```
GIVE INTERRUPTIBLE IVR 8888 WITH PRIORITY 2 WITH
    TREATMENT 23456
```

This command specifies both IVR treatment in Meridian Mail and priority, overriding any Meridian 1 IVR treatment defined for the ACD DN 8888.

Example 7

```
GIVE IVR qq WITH TREATMENT treat1 WITH PRIORITY 2
```

This command specifies both IVR treatment in Meridian Mail and priority, overriding any Meridian 1 IVR treatment defined for ACD DNs defined to variable “qq.”

Example 8

```
GIVE INTERRUPTIBLE IVR 9999 WITH PRIORITY 3 WITH
    TREATMENT 3456
```

This command specifies the IVR treatment in Meridian Mail and priority, overriding any Meridian 1 IVR treatment defined for ACD DN 9999.

GIVE MUSIC

<music route>

This command allows you to give music to the caller. If GIVE MUSIC is the first command given for a call, the call immediately receives music.

If the call is in an unanswered state, execution of the GIVE MUSIC command in effect answers the call.

The GIVE MUSIC command may be interrupted by a GIVE RAN or GIVE IVR command following it. Once the RAN or IVR treatment has completed, the music resumes if the call has not been answered or abandoned by then (see Example 3). The GIVE MUSIC command may be terminated by a GIVE RINGBACK or GIVE SILENCE command following it (or by any command that terminates the call). When a command that interrupts or terminates the GIVE MUSIC follows it, that command should be preceded by a WAIT, to prevent the caller from hearing only a brief burst from the music channel before the command following GIVE MUSIC is executed. The following examples illustrate how the commands are processed, and the role of the WAIT command.

This section gives three examples of scripts using the GIVE MUSIC command. Example 3 (which also gives RAN) provides the preferred method of call treatment.

Example 1

```
GIVE MUSIC 24

QUEUE TO 3600
```

Use the above script when:

- you want the caller to hear music as soon as the call is queued
- agents-to-traffic ratio is low, which means that most calls will be queued, rather than answered immediately

Note that if an agent is available, the caller hears a brief burst of music followed by ringback as the call is presented to the agent.

Example 2

```
QUEUE TO 3600

GIVE MUSIC 24
```

Use the above script when agents-to-traffic ratio is high, which means that most calls will be answered immediately, rather than queued.

Note that if an agent is not immediately available, the caller may hear a burst of ringback before hearing music.

Example 3

```
QUEUE TO 3600

WAIT 6          /* Ensures at least a cycle of ringback if
                the call is not answered immediately */

GIVE RAN 10

GIVE MUSIC 24

WAIT 10

GIVE RAN 12
```

If an agent is available, the call is answered immediately. If no agents are available, the caller hears six seconds of ringback before hearing the first RAN, from route 10. Then, if an agent is not available, the caller receives music for 10 seconds followed by a different RAN message, from route 12. At the end of the second RAN, the music is again heard, until the call is either answered by an agent or abandoned.

If a call is receiving music, RAN, ringback, or silence, the call receives ringback when presented to the agent.

GIVE RAN

<ran route>

This command allows you to give a Recorded Announcement (RAN) to a call. The call does not have to be queued to an ACD DN to receive RAN. If RAN is the first command applied to a call, the call is immediately answered with the RAN message. See Example 1.

If the call is in an unanswered state, execution of the GIVE RAN command may or may not answer the call, depending on whether the Meridian 1 has the FC68 package and whether the ASUP prompt for the RAN route (in overlay 16) is set to YES. If ASUP=NO, the RAN may not be heard by callers on CO trunks.

Treatments (such as GIVE MUSIC and GIVE SILENCE) interrupted by the RAN message resume when the message is finished. Commands that follow GIVE RAN are not executed until the RAN message is finished.

Example 1

```
GIVE RAN 12

QUEUE TO 3600
```

In the above script, the caller hears the entire RAN message before being queued to 3600.

Example 2

```
QUEUE TO 3600

WAIT 6          /* Ensures at least a cycle of ringback if
                the call is not answered immediately */

GIVE RAN 12
```

In the above script, only those calls that arrive when all agents are busy receive the RAN message (following six seconds of ringback). A call receiving RAN is presented to an agent as one becomes available, even if the RAN has not completed.

If a call is receiving music, RAN, ringback, or silence, the call receives ringback when presented to the agent.

GIVE RINGBACK

Giving ringback tones generally is not required in scripts because CCR provides ringback when it is normally expected (when a call is initially queued to an ACD DN by the QUEUE TO command, and when the call is presented to an agent).

If you want to give ringback in special situations, use WAIT to control how long the ringback lasts. Giving ringback cancels previous music treatment.

Unlike other treatments, ringback does not continue after a RAN. Thus, if ringback is specified prior to a RAN, the customer receives silence once the RAN is finished.

Example 1

```
GIVE RINGBACK

WAIT 10

GIVE RAN 12

QUEUE TO 3600
```

The call receives 10 seconds of ringback before receiving RAN. After RAN, the call is queued to 3600, and receives silence until presented to an agent.

Example 2

```
QUEUE TO 3600

GIVE MUSIC 15

WAIT 10

GIVE RINGBACK

WAIT 10

GIVE RAN 12
```

If no agents are available, the caller may hear a burst of ringback before the 10 seconds of music, the 10 seconds of ringback, and the RAN. After the RAN, the caller receives silence until presented to an agent. If a call is receiving music, RAN, ringback, or silence, the call receives ringback when presented to the agent.

GIVE SILENCE

This command allows you to turn off either music or ringback that has been previously specified by a GIVE MUSIC or GIVE RINGBACK command. If one of these commands precedes a GIVE RAN, and you want to have silence after RAN, use the GIVE SILENCE command to turn off the music prior to the giving RAN. The following examples illustrate how these commands work together.

Example 1

```
QUEUE TO 3600

GIVE MUSIC 6

WAIT 10

GIVE RAN 12

GIVE SILENCE
```

If you wish to use silence after RAN, this script is not as good as that used in Example 2. If an agent is not available, the caller receives a burst of the default ringback treatment and then 10 seconds of music prior to getting RAN. However, after RAN, the caller receives a brief burst of music before silence. This is because music persists through RAN. The GIVE SILENCE command is not executed until RAN completes, and then it takes a split second before silence can be applied to the call.

Example 2

```
QUEUE TO 3600

WAIT 6          /* Ensures at least a cycle of ringback if
                the call is not answered immediately */

GIVE MUSIC 6

WAIT 10

GIVE SILENCE

GIVE RAN 12
```

If there are no agents available, the caller receives six seconds of the default ringback treatment and 10 seconds of music prior to getting RAN. After RAN, the caller receives silence until presented to an agent.

If a call is receiving music, RAN, ringback, or silence, the call receives ringback when presented to the agent.

Script processing commands

Information listed in “< >” directly under the command is the information required for that command. The information in “{}” is additional optional information.

COMMENT

/* <text> */

Comments allow you to add clarity to a script (see Example 1) or to temporarily turn off a SECTION of the script language while testing a script (see Example 3). The symbol “/*” is used to start a comment, everything else is ignored until a “*/” is found, which then closes the comment. Nested comments (that is, comments within a comment) are not allowed and will not validate if used in a script. Example 2 shows a script that will not be validated.

Example 1

```
/* This script handles calls arriving at the sales
   department. It was created by J. Brown on March 4th
   1991 */

GOTO Closed       IF Time Of Day = after_hours

                  OR Day Of Week = weekend

                  OR Day Of Year = holiday

QUEUE TO sales    IF Total Queued Calls sales < 25

FORCE BUSY        IF Total Queued Calls sales >= 25

SECTION Closed

  GIVE RAN open_hours /* We are open from 8:00 AM to
     5:00 PM, please call during business hours. */

  FORCE DISCONNECT
```

In this script, the first comment indicates why the script is being used and when it was written. The second comment shares a line with a command and indicates what is said by the open_hours RAN.

Example 2

```
/* This is an incorrect example:

QUEUE TO 3600

GIVE RAN agents_busy
```

This script will not execute and will generate an error because the symbol to close the comment (*/) is missing.

Example 3

In the following script, the comment is used to turn off a SECTION of the script language temporarily. This can be advantageous during testing when you may want to test portions of the script SECTION by SECTION.

```
GOTO Closed      IF Time Of Day = after_hours

/* The following section of the script language is turned
    off temporarily, up to the end of the comment.

                 OR Day Of Week = weekend

                 OR Day Of Year = holiday

QUEUE TO sales   IF Total Queued Calls sales < 25

*/

FORCE BUSY       IF Total Queued Calls sales >= 25

SECTION Closed

  GIVE RAN open_hours /* We are open from 8:00 AM to 5:00
      PM, please call then */

  FORCE DISCONNECT
```

GOTO

<section label>

This command allows you to control the flow through the script. GOTO must be used with a valid SECTION label. You can use the GOTO command to direct the flow either forward or backward in the script, and to create a processing loop. More than one GOTO command can point to the same SECTION, potentially causing the SECTION to be executed more than once during the script.

In the following examples of scripts:

- "GOTO Nite_3600" branches forward—bypassing commands—only if the condition specified by the IF command is met
- "GOTO Wait_For_Free_Agents" at the end of the SECTION with the same label creates the processing loop

Both examples show how to use the GOTO command to branch to different SECTIONs and create a loop. However, the script in Example 2 is more complex—but also more efficient—than the script in Example 1.

Example 1

```
GOTO Nite_3600 IF Night Service 3600

QUEUE TO 3600   /* Normal office hours Queue */

GOTO Wait_For_Free_Agents

SECTION Nite_3600

    QUEUE TO 3700      /* Night Service Queue */

    QUIT

SECTION Wait_For_Free_Agents

    GIVE RAN agents_busy

    GIVE MUSIC 8

    WAIT 60

    GOTO Wait_For_Free_Agents
```

This script queues calls received outside normal business hours to 3700. During normal business hours, if all agents are busy, callers will hear a recorded message and music, which are repeated every 60 seconds until an agent is available or the call is terminated. This script unnecessarily repeats the GIVE MUSIC command; once the command is given, the music continues until it is turned off with GIVE RINGBACK or GIVE SILENCE, or until the call is terminated.

Example 2

```
GOTO Nite_3600 IF Night Service 3600

QUEUE TO 3600   /* Normal business hours queue  */

GOTO Wait_For_Free_Agents

SECTION Nite_3600

 QUEUE TO 3700        /* Night Service Queue  */

 QUIT

SECTION Wait_For_Free_Agents

 GIVE MUSIC 8

SECTION Record_Announcement

 GIVE RAN agents_busy

 WAIT 60

 GOTO Record_Announcement
```

This script also queues calls received outside normal business hours to a Night Service (3700). During normal business hours, if no agents are available, the caller hears music. A separate SECTION then gives the recorded message every 60 seconds until an agent is available or the call is terminated. This script does not repeat the GIVE MUSIC command unnecessarily.

IF

<command> IF <intrinsic tests>

The IF command allows you to execute commands only if certain conditions are met.

The IF command impacts the command immediately preceding it no matter how the script is written. Usually the two commands are written on the same line (see Example 1), but they do not have to be (see Example 2).

When writing scripts, be sure to format them so that the way they read is the way they operate. The way the script in Example 2 is written, it appears that the GOTO is unconditional, but that is not the case.

Example 1

```
GOTO Night_Service IF Day Of Week = weekend

QUEUE TO 3600
```

This script segment branches to the SECTION Night_Service *only* if it is the weekend, as defined in the Variable Table. Otherwise, the call is queued to ACD DN 3600. In this example, the IF command affects the GOTO command because it immediately precedes the IF.

Example 2

```
GOTO Night_Service

IF Day Of Week = weekend QUEUE TO 3600
```

This script operates exactly like the previous script although it has been written differently. The IF command always applies to the command preceding it.

QUIT

This command provides flow control in the script by ending further execution of commands. Use QUIT at the end of each SECTION to prevent the script from continuing to execute a SECTION simply because it follows an executable command. See Example 2.

A QUIT command can also be used to indicate the end of a script, but it is not required. See Example 1.

Example 1

```
QUEUE TO 3600

GIVE RAN 12

GIVE MUSIC kroc

QUIT
```

This script queues the call to 3600. If there are no agents available, the caller may hear a burst of ringback before hearing the RAN followed by music. The QUIT command has no impact on this script.

Example 2

```
GOTO Night_Service IF Night Service 3600

QUEUE TO 3600

GIVE MUSIC kroc

QUIT

SECTION Night_Service

   GIVE RAN closed

   FORCE DISCONNECT
```

Calls handled by this script branch to SECTION Night_Service only if queue 3600 is in Night Service, causing the call to receive the specified recorded message before being disconnected. If the queue is not in Night Service, the call receives music from the route defined as “kroc” in the Variable Table. The QUIT command prevents the execution of SECTION Night_Service. Without the QUIT command, the call would receive a burst of music followed by the closed RAN message, before being disconnected.

SECTION

<section label>

This command labels the target of a GOTO command (see Example 1). Every GOTO command must be used with a valid SECTION label. More than one GOTO command may point to the same SECTION (see Example 2).

> ***Note:*** SECTIONs cannot have the same name as variables or CCR key words.

A SECTION command may be used and/or executed without a GOTO pointing to it. However, the script validation process generates a warning message to indicate an untargeted SECTION.

The presence of a SECTION command does not block the flow of the script to the following command.

Example 1

```
GOTO Nite_3600 IF Night Service 3600

QUEUE TO 3600

GIVE MUSIC 8

QUIT

SECTION Nite_3600

    GIVE RAN closed

    FORCE DISCONNECT
```

A call being handled by this script branches to SECTION Nite_3600 only if queue 3600 is in Night Service. It then receives RAN and is force-disconnected. This is an example of using the GOTO to do a forward branch.

If ACD DN 3600 is not in Night Service, the call queues to 3600 and receives music if there are no agents available. The QUIT command prevents the script from continuing on and giving calls the closed RAN message.

Example 2

```
GOTO Nite_3600 IF Night Service 3600

GOTO Nite_3600 IF Time Of Day = out_of_hours

QUEUE TO 3600

GIVE MUSIC 7

QUIT

SECTION Nite_3600

    GIVE RAN closed

    FORCE DISCONNECT
```

This is an example where two GOTO commands point to the same SECTION.

WAIT

<seconds>

Generally, this command specifies the duration of the execution of the prior command in the script. WAIT cannot be the first command applied to a call.

Although the WAIT command can be specified with any number of seconds ranging from 0 to 32767 seconds (9.1 hours), it is executed in two-second intervals. For example, WAIT 1 and WAIT 2 have the same effect on the call. Any variable used with WAIT must be type SECONDS.

> ***Note:*** A WAIT command preceding a QUIT command has no effect on the script.

Example 1

```
QUEUE TO 3600 WITH PRIORITY 2

GIVE RAN 12

GIVE MUSIC 24

WAIT 15

QUEUE TO 3600
```

This script queues calls to ACD DN 3600 as priority level 2. If no agents are available, the call receives RAN followed by music. If the call has not been answered 16 seconds after the RAN finishes (WAIT is executed in two-second intervals), its priority changes to the default level 1. It continues to receive music until an agent is available.

Example 2

```
QUEUE TO 3600 WITH PRIORITY 2

GIVE RAN 12

GIVE MUSIC 24

WAIT one_minute

ROUTE TO 5006
```

If an agent is not available, the call may receive a burst of ringback before hearing RAN followed by music. If not answered or abandoned after a minute of music, the call is routed to extension 5006. It is automatically removed from queue 3600.

Script language intrinsics

Overview

Script language intrinsics are specific pieces of real-time information that can be tested using IF commands. Intrinsics can be based on time, call and queue information.

Time period intrinsics Time period intrinsics are specific times of day, elapsed time for the call, days of the week, and dates during the year. These intrinsics can be used in scripts to test the current time period against some desired time period or range of time periods. The time period intrinsics are

- Age Of Call
- Day Of Week
- Day Of Year
- Time Of Day

Queue status intrinsics The queue status intrinsics deal with ACD DN queues. These intrinsics can be used in scripts to test an ACD DN for a desired condition or set of conditions. An ACD DN must be specified along with the queue status intrinsic. The queue status intrinsics are

- Idle Agents
- Logged Agents
- Night Service
- Oldest Call
- Total Queued Calls

Call information intrinsics The call information intrinsics deal with specific pieces of information about the called or calling parties. These intrinsics can be used in scripts to test the current calling or called party for a desired condition or set of conditions. Availability of these intrinsics depends on the features in your Meridian 1. The call intrinsics are

- CLID (Calling Line Identification)
- DNIS (Directory Number Identification Service)
- LOC (first 3 digits of private network call)
- NPA (Numbering Plan Area code)
- NXX (local exchange)
- NPANXX (combined NPA and NXX used to identify calls for routing and treatment)

This section includes a description of each intrinsic and examples of how each is used.

Time period intrinsics

AGE OF CALL

Type: SECONDS

This intrinsic allows you to route or treat a call based on the age (in seconds) of the call. Time is measured from the point at which a CCR script receives a call for processing. It should not be used with the test for equality (=) as it is extremely unlikely that the test will be true.

Variables of type SECONDS specified in the Variable Table can be used in comparisons with this intrinsic. Lists and ranges cannot be used to express time. The arithmetic operators for multiplication (*), addition (+) and subtraction (-) may be used with this intrinsic. For more information on arithmetic operators, see the “Expressions” chapter.

Examples of statements using the Age Of Call intrinsic follow.

Example 1

```
GIVE RAN 12 IF Age Of Call > 60
```

RAN is given if the call is older than 60 seconds.

Example 2

```
GIVE RAN 12 IF Age Of Call > 60

AND IF Age Of Call < 180
```

A range can be accomplished by using two intrinsics. RAN is given if the call is older than 60 seconds but less than 180 seconds.

Example 3

```
GIVE RAN 12 IF Age Of Call > one_minute
```

RAN is given if the call is older than 60 seconds, provided one_minute is defined in the Variable Table as type SECONDS with the value of 60.

Example 4

```
GIVE RAN 12 IF Age Of Call * 2 > one_minute
```

RAN is given if the call is older than 30 seconds, provided that one_minute is defined in the Variable Table as type SECONDS with the value of 60.

DAY OF WEEK

Type: DAY

This intrinsic allows you to route or treat a call based on a specific day of the week. The days are specified as Monday, Tuesday, Wednesday, Thursday, Friday, Saturday or Sunday, with Monday as the first day and Sunday as the last day of the week.

Variables of type DAY specified in the Variable Table can be used in comparisons with this intrinsic.

Examples of statements using the Day Of Week intrinsic follow.

Example 1

```
GIVE RAN 12 IF Day Of Week = Friday
```

RAN is given on Fridays from 0:00 AM to 11:59 PM.

Example 2

```
GIVE RAN 12 IF Day Of Week > Friday
```

RAN is given on Saturday and Sunday because Sunday is the last day of the week, and is consequently an implied upper bound.

Example 3

```
GIVE RAN 12 IF Day Of Week = Saturday, Sunday
```

RAN is given on Saturday and Sunday.

Example 4

```
GIVE RAN 12 IF Day Of Week <= Friday
```

RAN is given Monday (the implied lower bound) through Friday.

Example 5

```
GIVE RAN 12 IF Day Of Week = Monday..Friday
```

RAN is given Monday through Friday.

Example 6

```
GIVE RAN 12 IF Day Of Week = Friday..Monday
```

RAN is given on Friday, Saturday, Sunday and Monday. This is an example of a wrapped range, in that it passes through Sunday.

Example 7

```
GIVE RAN 12 IF Day Of Week = weekend
```

RAN is given on Saturday and Sunday, provided that weekend is a variable defined as Saturday and Sunday.

DAY OF YEAR

Type: DATE

This intrinsic allows you to route or treat a call based on a specific day (date) of the year. The dates are specified as “MM/DD”, where “MM” is the month and “DD” is the day. Variables of type DATE specified in the Variable Table can be used in comparisons with this intrinsic.

Examples of statements using the Day of Year intrinsic follow.

Example 1

```
GIVE RAN 12 IF Day Of Year = 7/4
```

RAN is given on the 4th of July.

Example 2

```
GIVE RAN 12 IF Day Of Year <> 7/4
```

RAN is given every day except the 4th of July.

Example 3

```
GIVE RAN 12 IF Day Of Year >= 12/21
```

RAN is given from December 21st through December 31st. In this example, the December 31st upper bound is implied.

Example 4

```
GIVE RAN 12 IF Day Of Year = 12/21..1/1
```

RAN is given from December 21st through January 1st of the following year. This is an example of a wrapped range in that it passes through December 31st and continues on to January 1st of the next year.

Example 5

```
GIVE RAN 12 IF Day Of Year = 1/1, 3/29, 5/27, 7/4,
9/2, 11/28, 11/29 OR Day Of Year >= 12/21
```

RAN is given on January 1st, March 29th, May 27th, July 4th, September 2nd, and November 28th and 29th and from December 21st through December 31st (the implied upper bound).

TIME OF DAY

Type: TIME

This intrinsic allows you to route or treat a call based on a time of day. Time is specified by a 24-hour clock from 00:00 (midnight) to 23:59. It is recommended that comparisons made be against a range of time, since tests against a specific time are almost never true. Time ranges are specified by using two periods (..) between the beginning and ending times.

For Time Of Day, the time is tracked in one-minute intervals.

Comparisons to a specific time are permitted but not recommended, since tests for a specific time will almost never be true (see Examples 4 and 5). A comparison against a specific time means for a specific minute (for example, “IF Time Of Day = 08:00” is true for 8:00–8:01).

Comparisons to a range of time where either the upper or lower bound is implied (for example, in Examples 2 and 3) are not recommended. It is preferred to explicitly state the range of time.

Variables of type TIME specified in the Variable Table may be used in comparisons with this intrinsic.

Examples of statements using the Time Of Day intrinsic follow.

Example 1

```
GIVE RAN 12 IF Time Of Day = 08:00..17:00
```

RAN is given from 8:00 AM through 5:59 PM.

Example 2

```
GIVE RAN 12 IF Time Of Day >= 17:00
```

RAN is given from 5:00 PM through 11:59 PM. The statement becomes false at exactly midnight (0:00). In this example, the upper bound of 23:59 is implied. Thus, it is the same as: `IF Time Of Day = 17:00..23:59`. Using an implied upper bound in a time comparison is not recommended.

Example 3

```
GIVE RAN 12 IF Time Of Day < 8:00
```

RAN is given from 0:00 AM through 7:59 AM and is stopped at exactly 8:00 AM. In this example, the lower bound of 0:00 is implied. Thus, it is the same as: `IF Time Of Day = 0:00..7:59.` Using an implied lower bound in a time comparison is not recommended.

Example 4

```
GIVE RAN 12 IF Time Of Day = 8:00
```

RAN is given only for the 60 seconds between 8:00 AM and 8:01 AM. Checking on such a limited time range is generally not recommended.

Example 5

```
GIVE RAN 12 IF Time Of Day = 8:00, 9:30, 17:45
```

RAN is given for the 60 seconds between 8:00 AM and 8:01 AM, for the 60 seconds between 9:30 AM and 9:31 AM, and for the 60 seconds between 5:45 PM and 5:46 PM. Checking on such a limited time range is generally not recommended.

Example 6

```
GIVE RAN 12 IF Time Of Day = 17:00..07:59
```

RAN is given from 5:00 PM through 7:59 AM; the statement is false from 8:00 AM to 4:59 PM. This is a wrapped range in that it passes through midnight.

Example 7

```
GIVE RAN 12 IF Time Of Day < 8:00 OR Time Of Day >=
17:00
```

RAN is given from 5:00 PM through 7:59 AM; RAN is not given from 8:00 AM to 4:59 PM. This statement is equivalent to that given as Example 6.

Example 8

```
GIVE RAN 12 IF Time Of Day >= 8:00 AND Time Of Day
< 17:00
```

RAN is given from 8:00 AM through 4:59 PM; the statement is false from 5:00 PM to 7:59 AM. This statement is equivalent to that given as Example 1.

Queue status intrinsics

IDLE AGENTS

Type: INTEGER

Data type: <ACD DN>

This intrinsic allows you to route or treat a call based on the number of agents logged in at the specified ACD DN, but not on an active call and not currently in Not Ready mode. Putting a position into Make Set Busy (MSB) mode logs it out of the queue. Putting a position into Not Ready (NRD or NRDY) mode changes its status from Idle to Not Idle.

Variables of type INTEGER specified in the Variable Table can be used in comparisons with this intrinsic (see Example 2). Lists and ranges cannot be used for the number of idle agents. The Total Queued Calls and Logged Agents intrinsics can also be used in comparisons with the Idle Agents intrinsic. The arithmetic operators for multiplication (*), addition (+), and subtraction (-) may be used with this intrinsic (see Example 3).

Examples of statements using the Idle Agents intrinsic follow.

Example 1

```
QUEUE TO 3600 IF Idle Agents 3600 < 25
```

The call is queued to 3600 if less than 25 agents are logged in and idle.

Example 2

```
QUEUE TO 3600 IF Idle Agents 3600 < max_agents_3600
```

The call is queued to 3600 if the number of idle agents is less than the value of the variable “max_agents_3600”, defined as type INTEGER.

Example 3

```
QUEUE TO 3600

IF Total Queued Calls 3600 <= Idle Agents 3600 * 2
```

The call is queued to 3600 if the number of calls to 3600 is less than or equal to twice the number of agents logged in to queue 3600. For example, if 15 agents are logged in and idle in queue 3600, and the number of queued calls is less than 30, then the call is queued to 3600.

LOGGED AGENTS

Type: INTEGER

Data type: <ACD DN>

This intrinsic allows you to route or treat a call based on the number of agents logged in to the specified ACD DN.

Variables of type INTEGER specified in the Variable Table can be used in comparisons with this intrinsic. Lists and ranges cannot be used for the number of queued calls. Other intrinsics of type INTEGER may also be used in comparisons with this intrinsic.

The arithmetic operators for multiplication (*), addition (+) and subtraction (-) may be used with this intrinsic.

Examples of statements using the Logged Agents intrinsic follow:

Example 1

```
GIVE RAN 12 IF Logged Agents 3600 < 25
```

RAN is given if the number of agents logged in to queue 3600 is less than 25.

Example 2

```
GIVE RAN 12 IF Logged Agents 3600 < max_agents_3600
```

RAN is given if the number of agents logged in to queue 3600 is less than the value for max_agents_3600 as defined in the Variable Table.

Example 3

```
GIVE RAN 12

IF Total Queued Calls 3600 <= Logged Agents 3600 * 2
```

RAN is given if the number of calls queued to 3600 is less than or equal to twice the number of agents logged in to queue 3600. If 15 agents are logged in, and the number of queued calls is less than 30 the RAN is given.

NIGHT SERVICE

Type: BOOLEAN

Data type: <ACD DN>

This intrinsic allows you to make decisions on whether a command is to be executed based on the Night Service status at the specified ACD DN.

An example of a statement using Night Service follows.

Example 1

```
GIVE RAN 12 IF Night Service 3600
```

RAN is given if ACD DN 3600 is in Night Service. ACD agents are put into Night Service when all agents are logged out or the queue has been put into Night Service using the Night Service key on the supervisor telephone.

> ***Note:*** You must ensure that all scripts provide for Night Service for ACD DNs—or calls will stay in the script and ring with no answer until they are abandoned.

OLDEST CALL

Type: SECONDS

Data type: <ACD DN>

This intrinsic allows you to route or treat a call based on the amount of time, in seconds, that the oldest call at the specified ACD DN has waited.

Do not test this intrinsic for equality (=), as it is unlikely the call is exactly the same specific number of seconds old when the intrinsic is tested.

Variables of type SECONDS specified in the Variable Table can be used in comparisons with this intrinsic. Lists and ranges cannot be used for the age of the oldest call.

The arithmetic operators for multiplication (*), addition (+) and subtraction (-) may be used with this intrinsic.

Examples of statements using the Oldest Call intrinsic follow.

Example 1

```
GIVE RAN 12 IF Oldest Call 3600 > 30
```

RAN is given if the oldest call queued at 3600 has been waiting longer than 30 seconds.

Example 2

```
GIVE RAN 12 IF Oldest Call 3600 > one_minute
```

RAN is given if the oldest call queued at 3600 has been waiting longer than 60 seconds (provided that one_minute has been defined in the Variable Table as type SECONDS, with a value of 60).

Example 3

```
GIVE RAN 12 IF Oldest Call 3600 > Age Of Call
```

This statement might be used if the call coming into ACD DN 3600 has previously been queued to another ACD DN. If the oldest call queued to 3600 has been waiting longer than the call coming into 3600, the incoming call receives a recorded message.

Example 4

```
GIVE RAN 12 IF Oldest Call 3600 * 2 > Age Of Call
```

This statement takes the number of seconds that the oldest call in queue 3600 has been waiting, and multiplies it by two. Then it compares the resulting value with the number of seconds that the incoming call has been waiting. RAN is given only if the age of the incoming call is greater than twice the age of the oldest call For example, if oldest call has been waiting 15 seconds, the incoming call receives RAN only if it has been waiting longer than 30 seconds (that is, 15 * 2 seconds).

TOTAL QUEUED CALLS

Type: INTEGER

Data type:<ACD DN>

This intrinsic allows you to route or treat a call based on the total number of calls (not just CCR calls) queued at the specified ACD DN.

Variables of type INTEGER specified in the Variable Table can be used in comparisons with this intrinsic. Lists and ranges cannot be used for the number of queued calls. Other intrinsics of type INTEGER may also be used in comparisons with this intrinsic.

The arithmetic operators for multiplication (*), addition (+) and subtraction (-) may be used with this intrinsic.

Examples of statements using the Total Queued Calls intrinsic follow:

Example 1

```
QUEUE TO 3600 IF Total Queued Calls 3600 < 25
```

The call is queued to 3600 if the number of calls queued to 3600 is less than 25.

Example 2

```
QUEUE TO 3600

IF Total Queued Calls 3600 < max_queued_calls_3600
```

The call is queued to 3600 if the number of queued calls at 3600 is less than the value for max_queued_calls_3600 as defined in the Variable Table.

Example 3

```
GIVE RAN 12

IF Total Queued Calls 3600 >= Logged Agents 3600 * 2
```

RAN is given if the number of calls queued at 3600 is greater than or equal to twice the number of agents logged in at 3600. If 15 agents are logged in, RAN is given if the number of queued calls is greater than or equal to 30.

Call information intrinsics

CLID (Calling Line Identification)

Type: CLID

Data type: CLID or WILDCLID

This intrinsic allows you to route or treat a call based on the CLID (Calling Line Identification) or the Automatic Number Identification (ANI) digits of the call being handled. The CLID, associated with ISDN services, is the number of the calling party. The ANI, associated with non-ISDN services, is typically the calling number or billing account number. If neither the CLID nor the ANI is available for a call, the result of any CLID test is false.

The CLID can be specified with 1–32 digits. Lengths can vary as follows:

- For Meridian 1 internal calls, the CLID is the extension of the calling telephone set. Example 1 shows a test on an internal DN that is four digits in length.
- For external calls in the North American numbering plan, CLID can be a seven-digit number (if the area does not use the NPA code) or a 10-digit number (if the area does use the NPA code).
- For external calls in international numbering plans, the number of digits in a local or long distance call differs from plan to plan.

Lists and ranges can be used with this intrinsic. If the CLID is defined as a list or range, tests for equality determine whether or not the CLID is a member of that list or range. See Examples 2 and 3.

If a script controls calls that are subject to more than one numbering plan, use the Flexible Digit Strings (FDS) feature to specify CLID or ANI digits. To use the FDS feature, specify the CLID as a WILDCLID constant in the script (see Examples 5 and 6), or define a WILDCLID variable in the Variable Table (see Example 7). For rules in using the WILDCLID data type, refer to Table 7 in the “Building the Variable Table” chapter.

Example 1

```
GIVE RAN 12 IF CLID = 4000
```

RAN is given if the CLID is the internal DN 4000. This can test calls that have transferred from another ACD group or a voice mail queue.

Example 2

```
GIVE RAN 12 IF CLID = 4000..4100
```

RAN is given if the CLID is in the range from 4000 to 4100.

Example 3

```
GIVE RAN 12 IF CLID = 4100, 4800
```

RAN is given if the CLID is 4100 or 4800.

Example 4

```
GIVE RAN 12 IF CLID = vip_customers
```

RAN is given if the CLID is a member of the variable “vip_customers”, defined in the Variable Table as type CLID. The value of the variable may be a single CLID number or a set of CLID numbers.

Example 5

```
GIVE RAN 12 IF CLID = 21@
```

RAN is given if the CLID starts with 21. The wildcard character (@) represents as many as 32 characters of any value.

Example 6

```
GIVE RAN 12 IF CLID = 2?00
```

RAN is given if the CLID is 2000, 2100, 2200 and so on through 2900. The placeholder character (?) represents a single character of any value.

Example 7

```
GIVE RAN 12 IF CLID = belgium
```

RAN is given if the CLID is a member of the variable “belgium”. If the variable is defined in the Variable Table as type WILDCLID, the value of the variable may include the wildcard character (@) or the placeholder character (?).

Note: Be sure to provide for those calls arriving without CLID.

DNIS (Directory Number Identification Service)

Type: DNIS

Data type: DNIS

This intrinsic allows you to route or treat a call based on the DNIS (Dialed Number Identification Service) number for the call being handled. The DNIS comprises the last one through 31 digits dialed by the calling party.

Lists and ranges may be used with this intrinsic.

Examples of statements using the DNIS intrinsic follow.

Example 1

```
GIVE RAN 12 IF DNIS = 2374
```

RAN is given if the DNIS number is 2374.

Example 2

```
GIVE RAN 12 IF DNIS = 2374, 2500..2509, 2789
```

RAN is given if the DNIS number is 2374 or 2789 or included in the range from 2500 through 2509.

Example 3

```
GIVE RAN 12 IF DNIS = product_1
```

RAN is given if the DNIS number is a member of the variable “product_1” as defined in the Variable Table as type DNIS. Its value may be a single DNIS number or a set of DNIS numbers.

LOC

Type: LOC

Data type: LOC

This intrinsic allows you to route or treat a call based on the LOC of the call being handled. The LOC comprises the first three digits for private network calls. For example, if the number is 6-222-343-5343, the LOC is 343.

The format for these private numbers is 222 LOC XXXX.

If LOC is not available for the call, any LOC tests will be false. Items and sets can be used with this intrinsic. The LOC must be specified as a 3-digit number.

An example of a statement using the LOC intrinsic follows.

Example 1

```
GIVE RAN 12 IF LOC = 646
```

RAN is given if the LOC is 646.

Note: Be sure to provide for calls arriving without LOC.

NPA (Numbering Plan Area or Area Code)

Type: NPA

Data type: NPA

This intrinsic allows you to route or treat a call based on the NPA (Numbering Plan Area or Area Code) of the call being handled. For calls originating in North America, the NPA comprises the first three digits of a 10-digit CLID. The format for CLID numbers in North America is (NPA) NXX XXXX.

Effective January 1995, all products must comply with the new NPA naming convention, where the first digit is any number from 2 through 9, and the second and third digits are any number from 0 through 9. Sets, lists, and ranges can be used with this intrinsic, as illustrated in Examples 2–4.

If CLID is not available for the call, any NPA test is false.

Examples of statements using the NPA intrinsic follow.

Example 1

```
GIVE RAN 12 IF NPA = 415
```

RAN is given only if the NPA is 415.

Example 2

```
GIVE RAN 12 IF NPA = 408..415
```

RAN is given if the NPA is in the 408–415 range.

Example 3

```
GIVE RAN 12 IF NPA = 415, 408, 803, 215..219
```

RAN is given if the NPA is 415, 408, 803 or in the 215–219 range.

Example 4

```
GIVE RAN 12 IF NPA = california
```

RAN is given if the NPA is a member of variable “california”, defined in the Variable Table as type NPA. It can be an item or a set of NPA numbers.

Note: Be sure to provide for those calls arriving without NPA.

NXX (Local Exchange Code)

Type: NXX

Data type: NXX

This intrinsic allows you to route or treat a call based on the NXX (Local Exchange Code) of the call being handled. The NXX is the second three digits of the 10-digit CLID for calls originating in North America. The format for CLID numbers in North America is (NPA) NXX XXXX. If the CLID has only seven digits, the first three are NXX.

The NXX must be specified as a 3-digit number, where the first digit is any number from 2 through 9; and the second and third digits are any number from 0 through 9. Sets, lists and ranges can be used with this intrinsic.

If CLID is not available for the call, any NXX test is false.

Examples of statements using the NXX intrinsic follow.

Example 1

```
GIVE RAN 12 IF NXX = 940
```

RAN is given if the NXX is 940.

Example 2

```
GIVE RAN 12 IF NXX = 940..950
```

RAN is given if the NXX is in the range from 940 to 950.

Example 3

```
GIVE RAN 12 IF NXX = 940, 356, 833, 225..229
```

RAN is given if the NXX is 940, 356, 833 or in the range from 225 to 229.

Example 4

```
GIVE RAN 12 IF NXX = district_1
```

RAN is given if the NXX is a member of a variable “district_1”, defined in the Variable Table as type NXX. Its value may be a single NXX number or a set of NXX numbers.

Note: Be sure to provide for calls arriving without NXX.

NPANXX

Type: NPANXX

Data type: NPANXX

This intrinsic allows you to route or treat a call based on the NPANXX of the call being handled. The NPANXX (NPA+NXX) comprises the first three digits plus the second three digits of the 10-digit CLID for calls originating in North America. The format for these CLID numbers is (NPA) NXX-XXXX.

Items or sets can be used with this intrinsic. The NPANXX must be specified as a 6-digit number. The NPA portion follows the rules for NPA. The NXX portion follows the rules for NXX.

If CLID is not available for the call, any NPANXX test is false.

Examples of statements using the NPANXX intrinsic follow.

Example 1

```
GIVE RAN 12 IF NPANXX = 415940
```

RAN is given only if the NPANXX is 415940.

Example 2

```
GIVE RAN 12 IF NPANXX = 415940..415950
```

RAN is given if the NPANXX is in the range 415940–415950.

Example 3

```
GIVE RAN 12 IF NPANXX = 415940, 408356, 516833,
215225..215229
```

RAN is given if the NPANXX is 415940, 408356, 516833, or in the 215225–215229 range.

Example 4

```
GIVE RAN 12 IF NPANXX = area_1
```

RAN is given if the NPANXX is a member of the variable “area_1”, defined in the Variable Table as type NPANXX. Its value can be a single NPANXX number or a set of NPANXX numbers.

Note: Be sure to provide for calls arriving without NPANXX.

Table 14 lists all intrinsics and provides information about each.

Table 14
Intrinsic information

Intrinsic	Intrinsic Type	Information Required	Data Type
Time of Day	TIME		
Day of Week	DAY		
Day of Year	DATE		
Age of Call	SECONDS		
Logged Agents	INTEGER	ACD DN	ACD
Idle Agents	INTEGER	ACD DN	ACD
Total Queued Calls	INTEGER	ACD DN	ACD
Oldest Call	SECONDS	ACD DN	ACD
Night Service	BOOLEAN	ACD DN	ACD
DNIS	DNIS		DNIS
CLID	CLID		CLID
CLID	CLID		WILDCLID
NPA	NPA		NPA
NXX	NXX		NXX
NPANXX	NPANXX		NPANXX
LOC	LOC		LOC

Expressions

There are three kinds of expressions that can be used in IF tests. These expressions are Comparison, Logical and Mathematical. Each of these equation types is described in the following sections.

Order of operations

An operator is the symbol used to express a mathematical expression. An example of an operator is the “+” symbol for addition. When evaluating conditional expressions, the operator with the highest precedence (importance) is evaluated first, then the one with the next highest, and so on down to the operator with the lowest importance.

When operators appear more than once, or when two operators with equal importance appear in the same expression, they are evaluated left to right. Operators in conditional expressions are evaluated in the following order:

parentheses ( )

multiplication *

addition + and subtraction -

comparison operators =, <>, >, <, >=, <=

NOT

AND/OR

left to right

Thus, if an expression contains parentheses, the partial expression within the parentheses is resolved first. Then any multiplications are resolved, then additions and subtractions, and so on. The following examples compare execution of statements both with and without parentheses. In Example 1 the parentheses do affect execution of the statement; in Example 2 the parentheses do not affect execution of the statement.

Example 1

Given the statement

```
GIVE RAN 123 IF DNIS = 5557555 OR
(DNIS >= 5555656 AND DNIS <= 5556756)
```

The following is true:

- The partial expression within the parentheses (if DNIS is a number from 5555656 through 5556756) is resolved first to be true or false.
- After the expression within the parentheses is resolved, the remainder of the expression, which uses only comparison operators, is resolved left to right.
- The GIVE RAN command is executed only if the calling party dialed in on either of the following:
  - — a DNIS number with a value from 5555656 through 5556756
  - — the DNIS number 5557555

Without the parentheses, the statement would read

```
GIVE RAN 123 IF DNIS = 5557555 OR DNIS >= 5555656 AND
DNIS <= 5556756
```

This expression, which uses comparison operators only, is resolved left to right. The GIVE RAN command is executed only if both of the following are true:

- the DNIS number is 5557555 or the DNIS number is greater than or equal to 5555656
- the DNIS number is less than or equal to 5556765

In other words, the GIVE RAN command would never be executed.

Example 2

Given the statement

```
QUEUE TO 2500 IF Logged Agents > 10 AND Oldest Call < 60
```

The call is queued to ACD queue 2500 only if more than 10 agents are logged in at 2500 and no call has been in the queue as long as 60 seconds. Both partial expressions (on either side of the AND operator) must be true for the statement to be true. Specifically, if either partial expression tests false, the QUEUE TO command is not executed.

Parentheses may have no effect on the order in which the expression is resolved, but may be used simply to enhance the readability of an expression.

Given the expressions

```
IF (DNIS = 4000 AND DNIS = 8000)
```

and

```
IF DNIS = 4000 AND DNIS = 8000
```

The following is true:

- The expression IF DNIS = 4000 AND DNIS = 8000 is resolved first to be true or false, whether it is in parentheses or not.

Comparison expressions

The script language comparison allows you to compare values of intrinsics, variables and constants. The data type on each side of the comparison equation must be the same for the comparison to be a legal expression. This enables you to compare two DN type values or two INTEGER type values, but not a DN type to an INTEGER type.

The comparison operators are: <, >, <=, >=, =, and <>. Each operator requires two operands to form a comparison expression (for example, operand1 < operand2). An operand can be an intrinsic, a variable, a constant or a mathematical expression. For example, the expression

```
Total Queued Calls main_acd <= 40
```

uses an intrinsic (of type INTEGER) as the left-hand operand and a constant (of type INTEGER) as a right-hand operand. The expression compares the number of calls queued to any ACD DN defined as main_acd with the constant 40. If the number of queued calls is less than or equal to 40, the expression is true.

Comparison expressions have a higher precedence than logical expressions. Refer to the “Logical expressions” section of this chapter for more information. Comparison expressions are used to evaluate a specific situation. For example, a test of the expression

```
Time Of Day >= 08:00
```

always yields either a yes (true) or no (false) answer.

Values that can be only true or false are referred to as Boolean values. The state of several situations at once (for example, it is later than 08:00 in the morning and at least one agent logged into the queue) can be evaluated by writing several comparison expressions and then joining them into a logical expression consisting of a series of yes/no answers.

The comparison operators <, >, <=, and >= can be used with ITEM type (that is, single value) variables, ITEM type constants, and intrinsics only; sets cannot be used. In this context, comparison operators have the following meanings:

- The < operator allows two values of the same type to be compared so that, if the first value is less than the second value, the expression is true.
- The > operator allows two values of the same type to be compared so that, if the first value is greater than the second value, the expression is true.
- The <= operator allows two values of the same type to be compared so that, if the first value is less than or equal to the second value, the expression is true.
- The >= operator allows two values of the same type to be compared so that, if the first value is greater than or equal to the second value, the expression is true.
- The comparison operators = and <> can be used with sets as well as ITEM types in the following way:
  - The = operator allows two values of the same type to be compared. If both values are single values and if the first value is equal to the second value, the expression is true.

    If the left side value is an item, and the right side value is a set, the expression is true if the item on the left is contained within the set on the right. A set value can only be used on the right side of this expression. See Examples 2 and 3.
  - The <> (not equal to) operator allows two values of the same type to be compared. If both values are single items and if the first value is equal to the second value, the expression is false. If the first value is an item, and the second value is a set, the expression is true if the first value is not a member of the set. A set can be used only on the right side of this expression. See Example 1.

Example 1

```
QUEUE TO 2500 IF CLID <> 4087307001
```

The QUEUE TO command is executed only if the calling party’s number is not (408) 730-7001.

Example 2

```
QUEUE TO 2500 IF Time Of Day = 08:00..17:00
```

The QUEUE TO command is only executed if the current time is between 8:00 AM and 5:00 PM.

Example 3

```
QUEUE TO 2500 IF DNIS = 5556, 4567, 1234, 8799
```

The QUEUE TO command is only executed if the DNIS number of the call is one of 5556, 4567, 1234 or 8799. If the call has no DNIS number, or the DNIS number is not one of the above numbers, the QUEUE TO is not executed.

> ***Note:*** CLID intrinsics using the WILDCARD data type can only be used with the = and <> comparison operators.

Logical expressions

The script language allows you to string together several comparison expressions and allow complex evaluations in one IF command. The logical equations are: AND, OR, NOT, and ( ). Table 15 summarizes the effect of each operator when logically evaluating comparison expressions.

The AND operator evaluates two comparison expressions. If both expressions are true, the expression is true. If either expression is false, the condition for executing the command is not met. See Example 1.

The OR expression evaluates two comparison expressions. If either expression is true—or both expressions are true— the condition for executing the command is met. If both expressions are false, the condition for executing the command is not met. See Example 2.

The NOT expression evaluates a conditional expression negates it. If the result of the expression (without the NOT clause) is true, the entire expression (with the NOT clause) is false. If the expression (without the NOT clause) is false, the entire expression (with the NOT clause) is true. See Example 3.

Example 1

```
QUEUE TO 2500 IF (Time Of Day >= 8:00)
AND (Time Of Day <= 17:00)
```

The QUEUE TO command as seen here is only executed if the current time is between 8:00 AM and 5:00 PM, inclusive.

Example 2

```
ROUTE TO 2500 IF (Day Of Week = Saturday)
OR (Day Of Week = Sunday)
```

The ROUTE TO command as seen here is executed only if the current day is Saturday or Sunday.

Example 3

```
QUEUE TO 2500 IF NOT (Night Service 2500)
```

The call is queued to 2500 only if queue 2500 is not in Night Service.

Table 15
Boolean Table

Operation	Value of Item 1	Value of Item 2	Value of Expression
AND	FALSE	TRUE	FALSE
	FALSE	FALSE	FALSE
	TRUE	TRUE	TRUE
	TRUE	FALSE	FALSE
OR	FALSE	TRUE	TRUE
	FALSE	FALSE	FALSE
	TRUE	TRUE	TRUE
	TRUE	FALSE	TRUE
NOT	TRUE		FALSE
	FALSE		TRUE

For example:

```
(Item 1) and (Item 2)
  True          True     =  True

(Item 1) and (Item 2)
  True          False     =  False
```

Mathematical expressions

The script language mathematical expressions allow you to perform mathematical calculations on variables, constants, and intrinsics of type INTEGER and SECONDS. The mathematical expressions are * (multiplication), + (addition), and - (subtraction).

Mathematical expressions take precedence over comparison or logical expressions. The * operator takes precedence over either the + or the - operators. See Example 5.

The * operator multiplies two values: both of type INTEGER, or one of type INTEGER and the other of type SECONDS. The result of this operation (INTEGER or SECONDS respectively) can be compared to other like values to complete an expression. See Examples 1 and 2.

The + operator adds two values: both of type INTEGER, or both of type SECONDS. The result of this operation can be compared to other like values to complete an expression. See Example 3.

The - operator subtracts two values: both of type INTEGER, or both of type SECONDS. The result of this operation can be compared to other like values to complete an expression. See Example 4.

Note: Do not use subtraction if the expected result is negative.

The ( ) operator allows the precedence for evaluating a mathematical operation to be changed. Mathematical operations are evaluated left to right with the multiplication * operator taking precedence over the + and - operators. With parentheses, a user can force an expression to be evaluated in a desired order. See Example 5.

Example 1

```
FORCE BUSY IF Logged Agents 2500
> Total Queued Calls 2500 * 4
```

The FORCE BUSY command is executed only if ACD DN 2500 has more than four times as many queued calls as it has agents logged in.

Because mathematical operators take precedence over comparison operators, the number of calls queued to ACD DN 2500 is multiplied by four before that value is compared with the number of agents in queue 2500.

Example 2

```
FORCE BUSY IF Oldest Call 2500 > min_wait * Logged
Agents 2500
```

The FORCE BUSY command is executed only when the oldest call in queue 2500 has waited longer than the minimum wait time (a variable of type SECONDS) times the number of agents in the queue (INTEGER).

Because mathematical operators take precedence over comparison operators, the minimum waiting time is multiplied by the number of agents in queue 2500 before that value is compared with the time that the oldest call in queue 2500 has been waiting.

Example 3

```
ROUTE TO 5000 IF Total Queued Calls 2500 < Total
Queued Calls 2501 + Total Queued Calls 2502
```

The ROUTE TO command is executed only if ACD 2500 has fewer queued calls than the combined number of calls queued in ACD DNs 2501 and 2502.

Because mathematical operators take precedence over comparison operators, the number of calls queued to ACD DN 2501 is added to the number of calls queued to ACD DN 2502 before that value is compared with the number of calls queued to ACD DN 2500.

Example 4

```
GIVE RAN 234
IF Oldest Call 2500 - Oldest Call 2501 < 60
```

The GIVE RAN command is executed only if the waiting time of the oldest call in ACD DN 2500 minus the waiting time of the oldest call in ACD DN 2501 is less than 60 seconds.

Example 5

```
REMOVE FROM 2500 IF (2 + Logged Agents 2500) * 3
> Total Queued Calls 2500
```

The REMOVE FROM command is executed only if the number of agents in queue 2500, when added to two and then multiplied by three, is greater than the number of calls queued against ACD DN 2500.

Because the * operator takes precedence over the + operator, without the parentheses the number of agents in queue 2500 would be multiplied by three before being added to two.

Writing scripts

Overview

This chapter covers the mechanics of the user interface for writing scripts. Refer to the “Planning your scripts” and “Sample scripts” chapters for information about what to consider before you begin writing scripts.

This chapter covers information about the following:

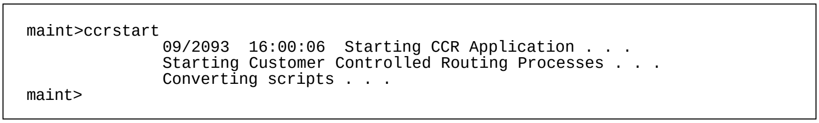
- scripting terms
- working with scripts (creating, validating, installing, removing, deleting, editing, and viewing)

Note: CCR accepts up to 1000 user-defined scripts.

CCR Release 3 conversion feature

The compiler has been changed for CCR Release 3. Because of this, all scripts are automatically recompiled (that is, revalidated) when the “ccrstart” command is executed for the first time after CCR is upgraded to Release 3. Figure 43 illustrates the messages that appear on the screen. The “Converting scripts . . .” message appears only once after an upgrade to Release 3. If this message overwrites the maint prompt, press <cr> to redisplay the prompt.

Figure 43
Script conversion messages



```
maint>ccrstart
              09/2093  16:00:06  Starting CCR Application . . .
              Starting Customer Controlled Routing Processes . . .
              Converting scripts . . .
maint>
```

Each script is checked for its validator version number. If the validator version number is changed, the conversion feature checks the script executable and revalidates the script. Manual intervention is required only if errors are found during the revalidation. All script revalidations and errors are reported to the CCR log files.

Accessing the Check/Change Call Scripts menu

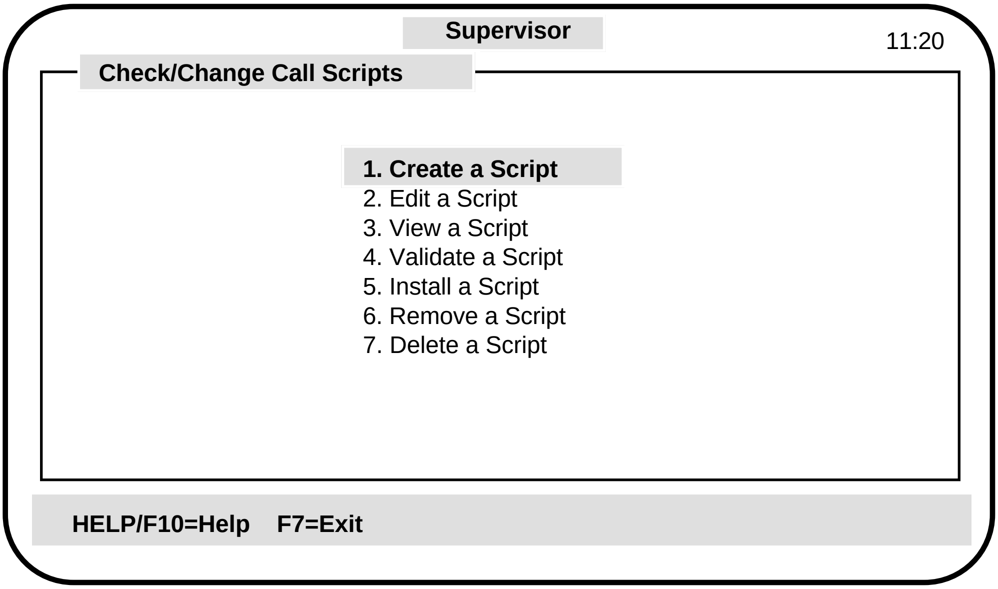
To access the script option, complete the following steps:

1 Select the Check/Change Call Scripts option from the Main Menu. Global access allows you to create, edit, print, validate, remove, delete, and install the scripts. View access allows you only to view and print the scripts.

The Check/Change Call Scripts menu appears.

2 Select any of the options shown in Figure 44.

Figure 44
Check/Change Call Scripts menu



Supervisor

11:20

Check/Change Call Scripts

1. Create a Script
2. Edit a Script
3. View a Script
4. Validate a Script
5. Install a Script
6. Remove a Script
7. Delete a Script

HELP/F10=Help F7=Exit

Script terms

Listed below are some terms that you need to know when you write and use scripts:

Create This allows you to name, edit, print, and import a new script. An existing script can be imported into this new script.

Edit This selection enables you to edit and print any script that is not installed. During an edit session, a script can be printed and another script can be imported.

View This allows you to view and print the text of any script.

Validate This confirms the accuracy of a script once it has been created or edited. If there are any errors in the script, such as syntax errors, the script will not validate, an error message appears at the bottom of your screen, and you return to the script editor. If you have selected Flag_on in your user profile, you will also see error, warning, and information messages alerting you to possible problems. Only scripts that successfully validate without errors can be installed. Flag_off will only alert you to error messages.

Install This prepares a validated script to be associated with a CDN to handle and treat calls.

Remove This removes a script from installed status. You cannot delete installed scripts that are associated with a CDN. Scripts must be removed before they can be changed.

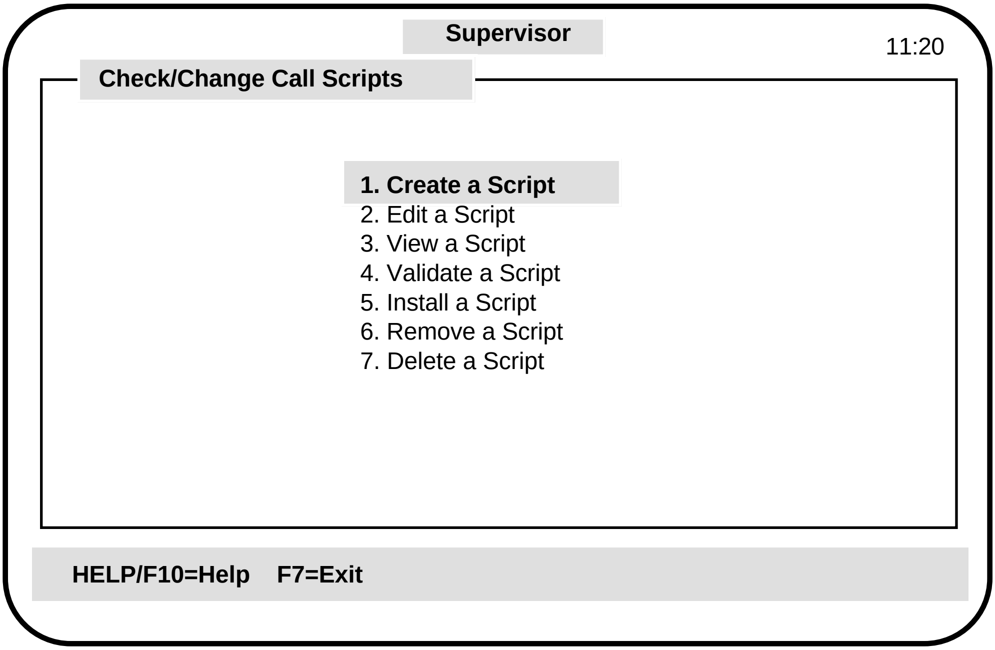
Delete This deletes a script from the system entirely. You can only delete scripts that are not installed.

Creating a script

Complete the following procedure to create a new script:

1 Access the Check/Change Call Scripts menu.

Figure 45
Check/Change Call Scripts menu



Supervisor 11:20

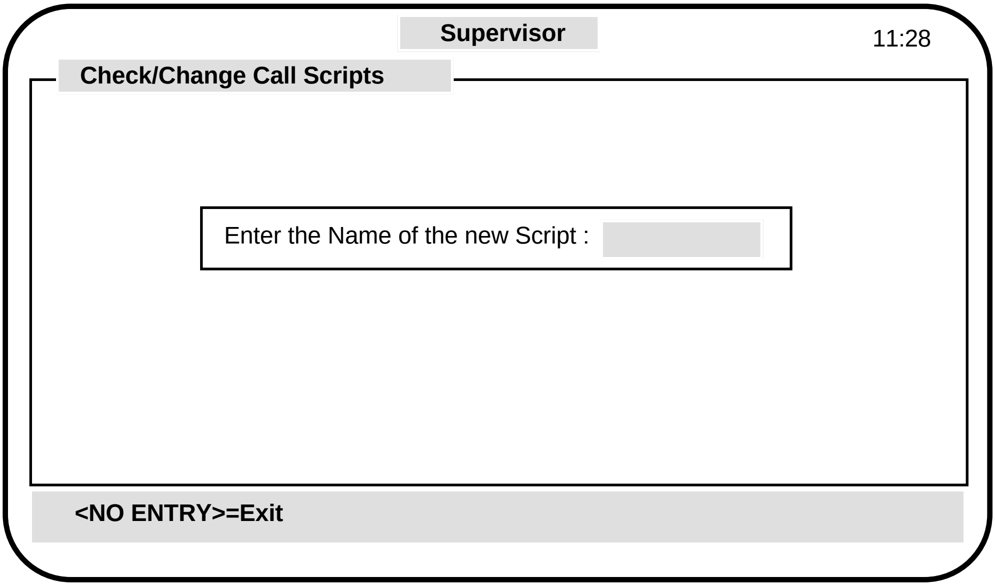
Check/Change Call Scripts

1. Create a Script
2. Edit a Script
3. View a Script
4. Validate a Script
5. Install a Script
6. Remove a Script
7. Delete a Script

HELP/F10=Help F7=Exit

2 Select the Create a Script option from the Check/Change Call Scripts menu.

Figure 46
Create a Script screen



Supervisor

11:28

Check/Change Call Scripts

Enter the Name of the new Script :

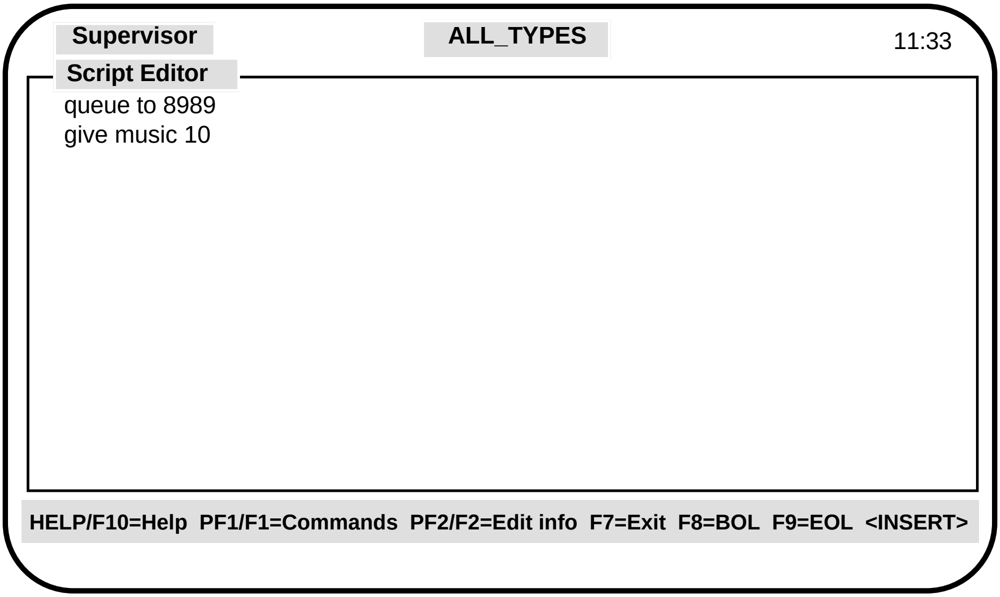
<NO ENTRY>=Exit

The Create a Script screen appears.

3 Type in a unique name for the script. The name can have up to 12 alphanumeric characters (blanks or spaces cannot be used in a script name).

Hint: You can use an underscore (_) to indicate a space.

Figure 47
Script Editor window



Supervisor
ALL_TYPES
11:33
Script Editor

```
queue to 8989
give music 10
```

HELP/F10=Help PF1/F1=Commands PF2/F2=Edit info F7=Exit F8=BOL F9=EOL <INSERT>

The script editor appears, with the script name indicated at the top of the screen, as illustrated in Figure 47.

4 Begin writing your new script. Text is inserted at the cursor position. You can use the cursor movement and function keys while writing your script, as explained in the “Getting started” chapter. You can also access the online help for additional information.

Note: Refer to the “Sample scripts” chapter for specific script writing tips.

5 Press F6 to save your changes before you exit. A dialog box prompts you to save your script when exiting. This is helpful if you want to leave after entering only a partial script.

To revise a script at a later time, use the Editing a Script option.

Note: To use your script, you must validate it. Refer to the “Validating a script” section for more information.

Validating a script

After writing or editing a script, it must be validated to confirm its accuracy, and installed, before you can use it to handle any calls.

If there are any errors—such as syntax errors—in the script, it will not be validated. An error message appears, and you return to the script editor.

Only scripts that are validated without errors can be installed.

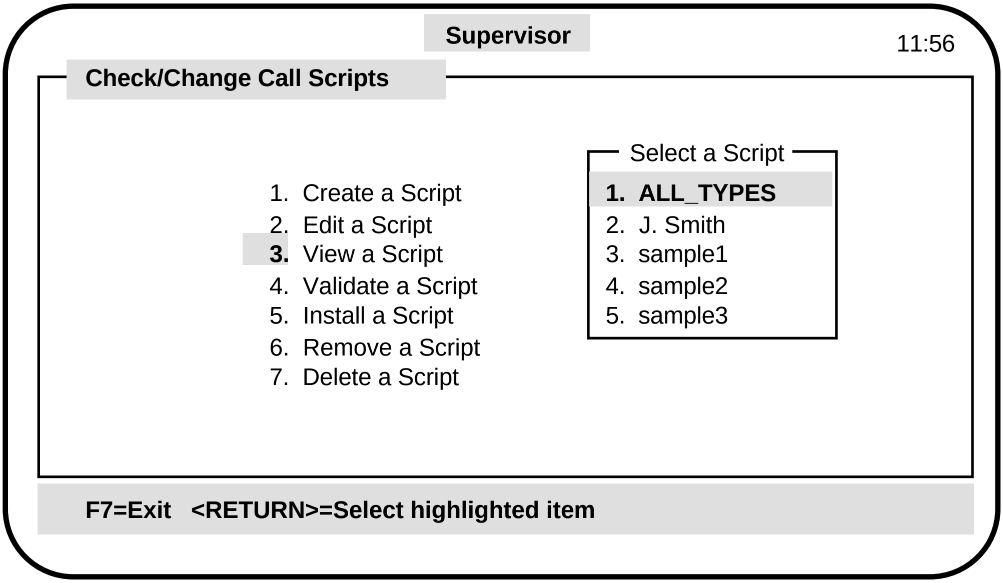
> ***Note:*** If you have set your user profile's validation flag to Flag_off, you will see only error messages at the bottom of your screen. If you have set your user profile's validation flag to Flag_on, you will also see warning and information messages that alert you to possible problems.

Refer to the "CCR messages" chapter in this manual to understand any validation messages CCR displays.

Complete the following procedure to validate a script:

1 Select the Validate a Script option from the Check/Change Call Scripts menu.

Figure 48
Validate a Script window



Supervisor 11:56

Check/Change Call Scripts

1. Create a Script
2. Edit a Script
3. View a Script
4. Validate a Script
5. Install a Script
6. Remove a Script
7. Delete a Script

Select a Script

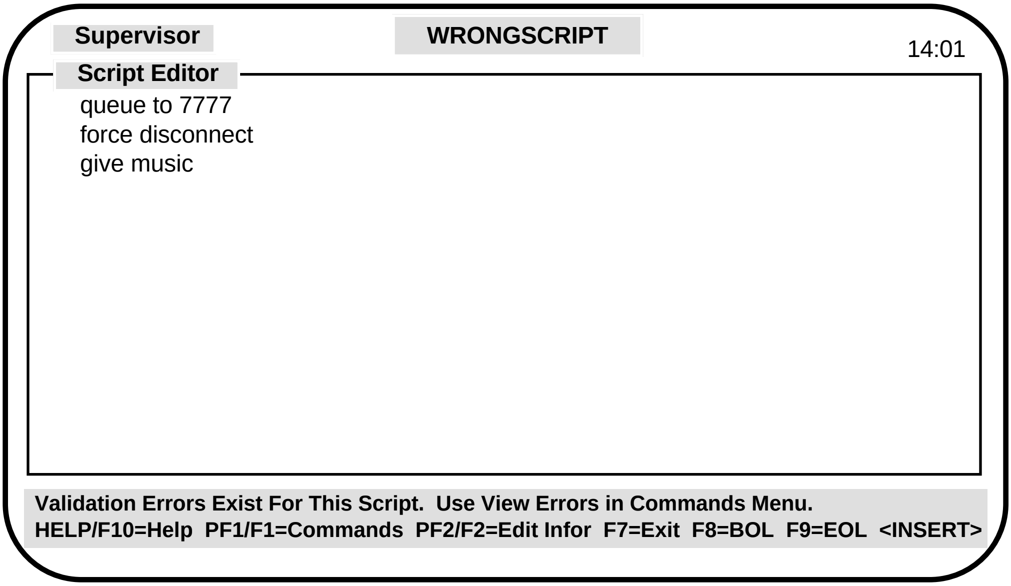
1. ALL_TYPES
2. J. Smith
3. sample1
4. sample2
5. sample3

F7=Exit <RETURN>=Select highlighted item

A pop-up menu appears, with a list non-installed scripts (that is, the scripts ready for validation) is illustrated in Figure 48.

2 Select the script you want to validate.

Figure 49
Validation unsuccessful

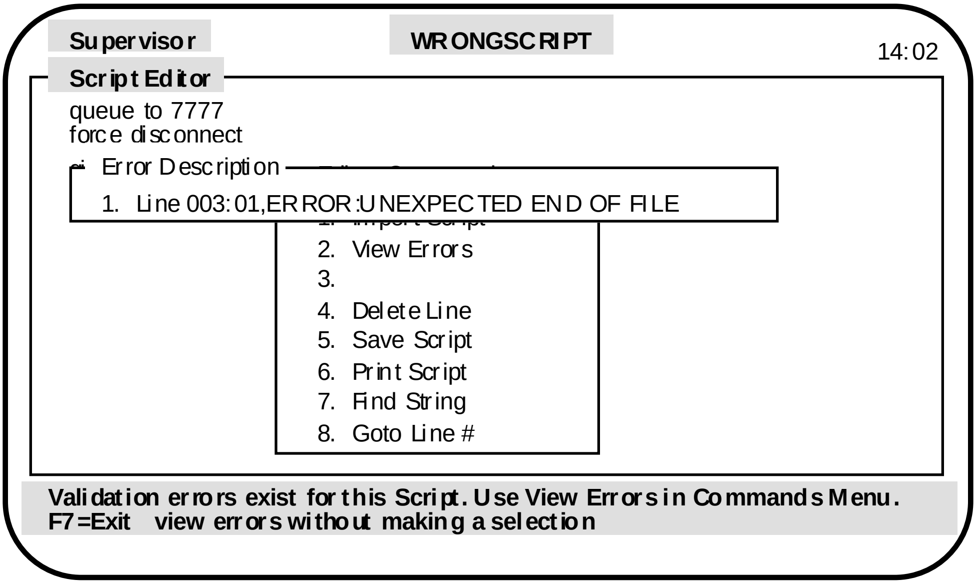


```
queue to 7777
force disconnect
give music
```

```
Validation Errors Exist For This Script.  Use View Errors in Commands Menu.
HELP/F10=Help  PF1/F1=Commands  PF2/F2=Edit Infor  F7=Exit  F8=BOL  F9=EOL  <INSERT>
```

The pop-up menu disappears and a message appears at the bottom of the screen If the validation is successful, the message tells you so. If the validation is not successful, the message tells you to check for errors in the script (Figure 49).

Figure 50
View Errors screen



```
queue to 7777
force disconnect
```

1. Line 003:01,ERROR:UNEXPECTED END OF FILE

Validation errors exist for this Script. Use View Errors in Commands Menu.
F7=Exit view errors without making a selection

If an error is found during validation, you are automatically returned to the script editor.

3 Press PF1/F1 and select View Errors. An Error Description window displays a list of errors (Figure 50).

4 Select an error in the window and press Return. The cursor is placed at the error location in the Script Editor window.

5 Edit the script so as to correct the error.

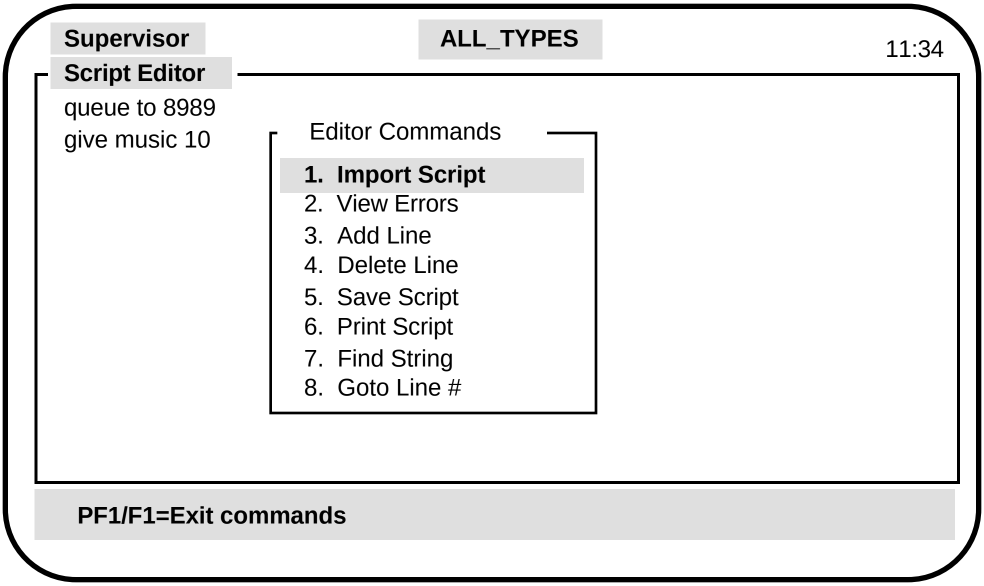
6 When you have corrected the error, revalidate the script.

For an explanation of the editor commands, see the “Editing a script” section.

Editing a script

When you are in the script editor, several editor command options are available (as shown in Figure 51) by pressing PF1/F1.

Figure 51
Script Editor menu



```
Supervisor                 ALL_TYPES
Script Editor                                     11:34
queue to 8989
give music 10       Editor Commands
                    1.  Import Script
                    2.  View Errors
                    3.  Add Line
                    4.  Delete Line
                    5.  Save Script
                    6.  Print Script
                    7.  Find String
                    8.  Goto Line #

PF1/F1=Exit commands
```

Editor commands

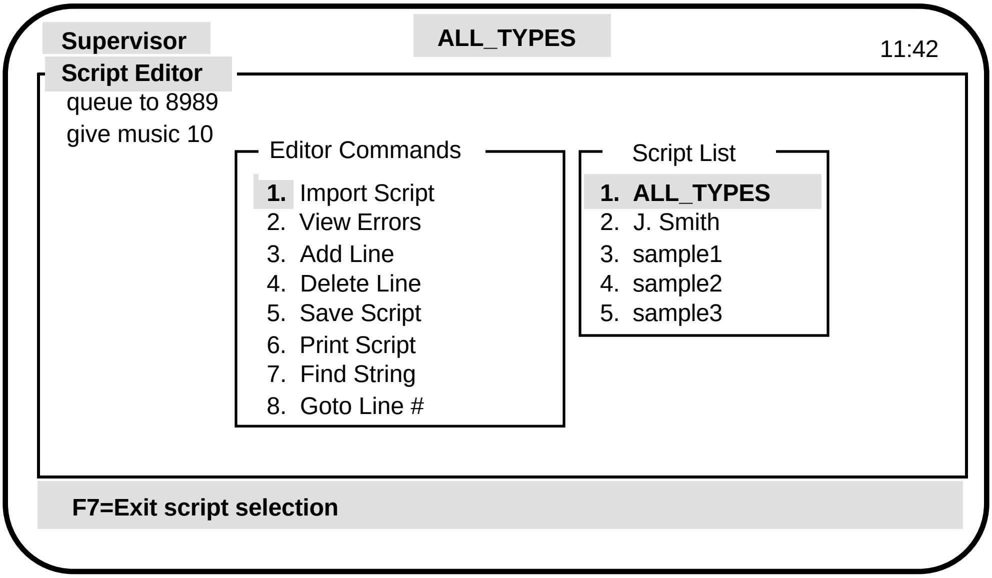
Import Script

With this option, you can import the entire text of another script into a new script. The imported script begins on the line following the line that contains the cursor.

When you select this command, a pop-up menu displays a list of existing scripts. To bring a copy of that script into the script being edited, select the desired script. Only a complete script can be imported; portions cannot be imported.

This is the only way that you can make a copy of a script without changing the original script. Once you have made changes to the new script, you can delete the previous one.

Figure 52
Import Script menu



Supervisor

ALL_TYPES

11:42

Script Editor

queue to 8989
give music 10

Editor Commands

1. Import Script
2. View Errors
3. Add Line
4. Delete Line
5. Save Script
6. Print Script
7. Find String
8. Goto Line #

Script List

1. **ALL_TYPES**
2. J. Smith
3. sample1
4. sample2
5. sample3

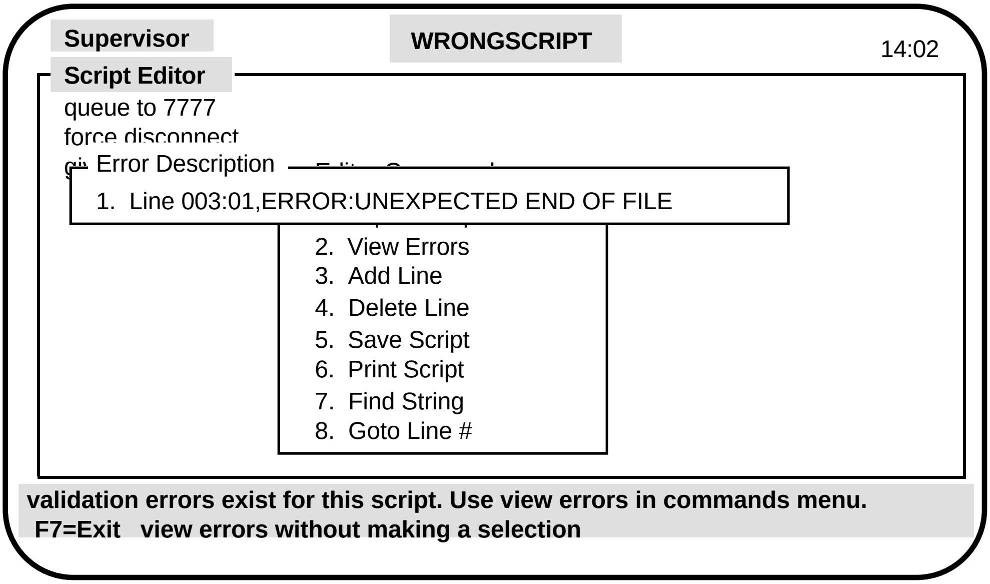
F7=Exit script selection

View Errors

Select the View Errors option to display a screen showing any warning, error, and information messages for a script. If your user profile’s validation flag is set to Flag_off, the screen shows error messages only. If your user profile’s validation flag is set to Flag_on, the screen also shows warnings and information messages.

Selecting an error (highlighting the error and pressing Return) places the cursor at the error in the editor screen. If there are no errors or warnings for the script, the command area is shaded and you cannot open it. See the “CCR messages” chapter for more information.

Figure 53
View Error screen



queue to 7777
force disconnect

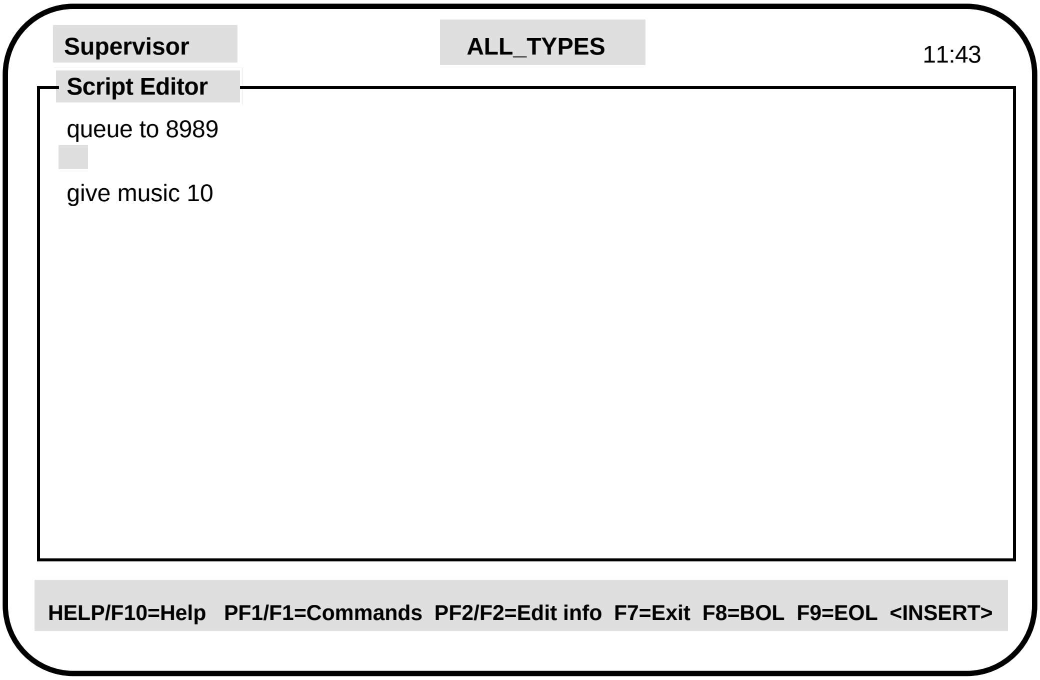
1. Line 003:01,ERROR:UNEXPECTED END OF FILE

validation errors exist for this script. Use view errors in commands menu.
F7=Exit view errors without making a selection

Add Line

This command inserts a blank line below the line of the cursor, as shown in the following figure.

Figure 54
Add Line screen



Supervisor ALL_TYPES 11:43

Script Editor

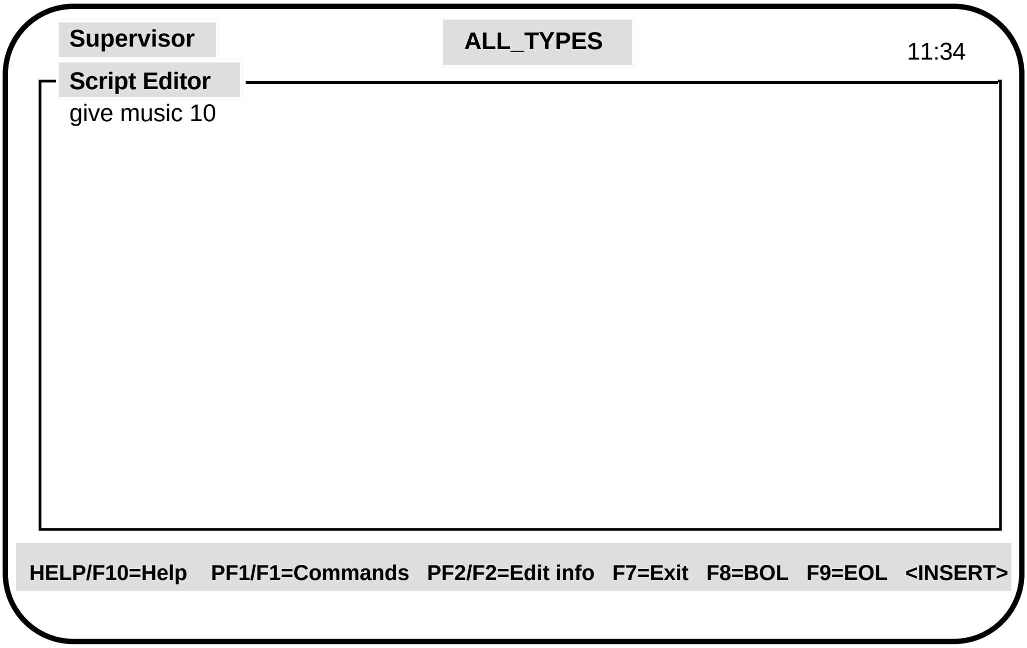
```
queue to 8989

give music 10
```

HELP/F10=Help PF1/F1=Commands PF2/F2=Edit info F7=Exit F8=BOL F9=EOL <INSERT>

Delete Line

This command deletes the line at the cursor location, as shown in the following figure.

Figure 55
Delete Line screen



Supervisor

ALL_TYPES

11:34

Script Editor

```
give music 10
```

HELP/F10=Help PF1/F1=Commands PF2/F2=Edit info F7=Exit F8=BOL F9=EOL <INSERT>

Save Script

This command saves the script and returns you to the editor.

Print Script

This command lets you print the script to the default printer (or any listed printer) as specified in your profile.

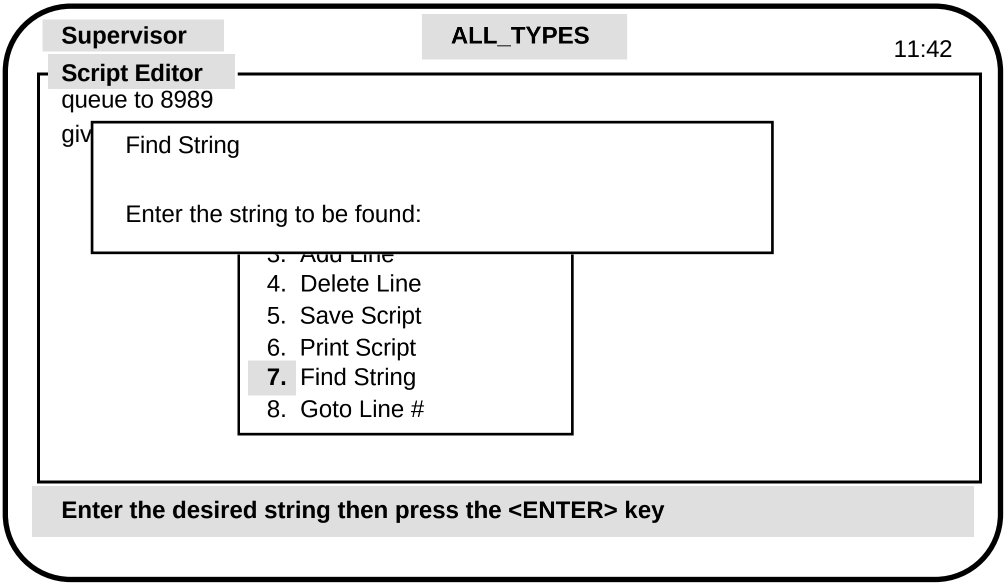
Find String

This command finds a specified text string in the script.

When you select the command, a pop-up menu appears. You have the following options:

- To define a search, type up to 32 characters (case sensitive).
- To move the cursor to the first character of the located string, press Return.
- If the search is not successful, a dialog box appears indicating that a string has not been found. To restore the cursor to its original position in the text, press Return.

Figure 56
Find String pop-up menu



Supervisor

ALL_TYPES

11:42

Script Editor

queue to 8989

giv

Find String

Enter the string to be found:

3. Add Line
4. Delete Line
5. Save Script
6. Print Script
7. Find String
8. Goto Line #

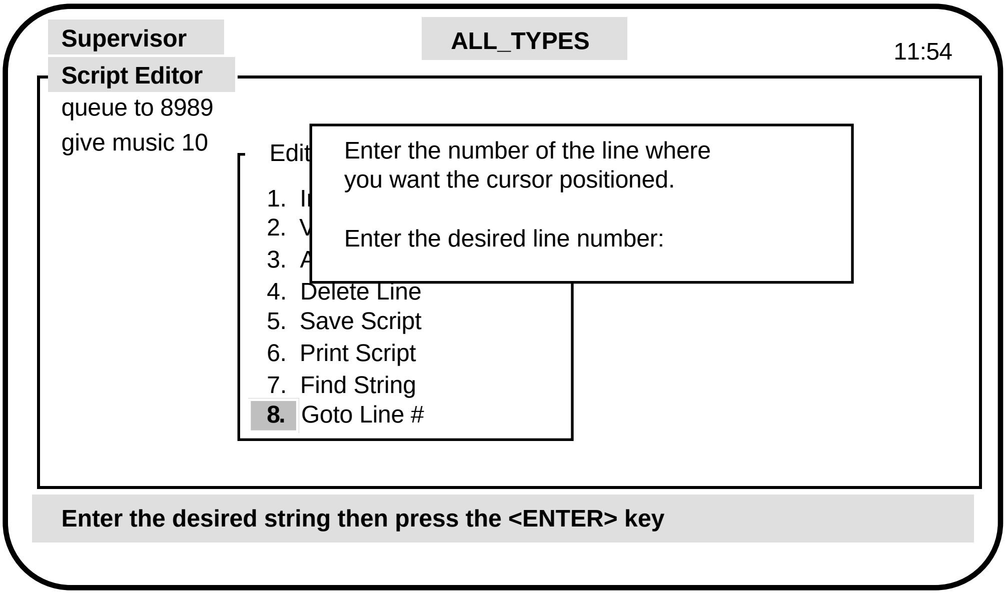
Enter the desired string then press the <ENTER> key

Go to Line

This command moves the cursor to the beginning of the specified line number.

Validation errors and warnings are specified by line numbers, but the script editor does not show line numbers. Use the Go to Line # command to find lines listed in an error file—or use the View Errors command instead.

Figure 57
Go to pop-up menu



queue to 8989
give music 10

Enter the number of the line where
you want the cursor positioned.

Enter the desired line number:

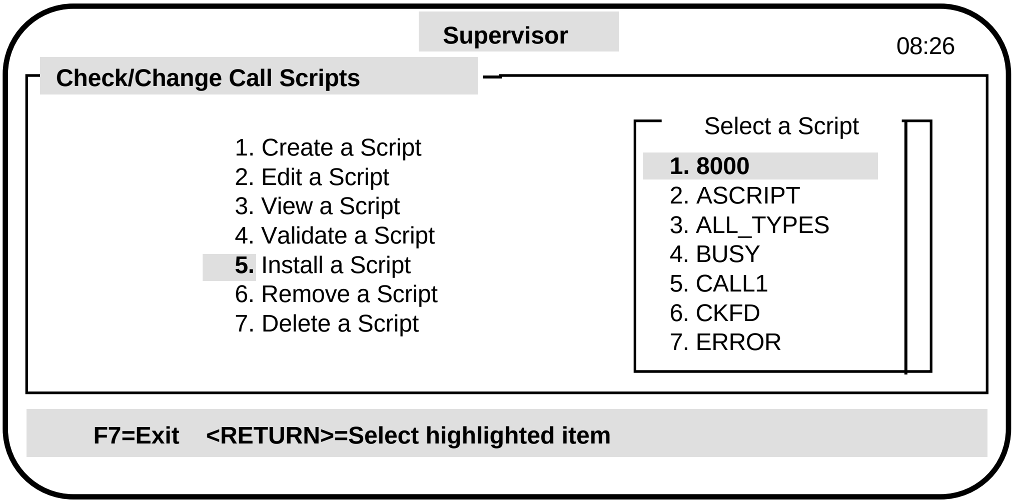
Enter the desired string then press the <ENTER> key

Installing a script

After you have validated a script, you are ready to install it so that it can be associated. For more information, see the “Associating CDNs and scripts” chapter.

1 Select the Install a Script option from the Check/Change Call Scripts menu.

Figure 58
Install a Script window



Supervisor 08:26

Check/Change Call Scripts

1. Create a Script
2. Edit a Script
3. View a Script
4. Validate a Script
5. Install a Script
6. Remove a Script
7. Delete a Script

Select a Script

1. 8000
2. ASCRIPT
3. ALL_TYPES
4. BUSY
5. CALL1
6. CKFD
7. ERROR

F7=Exit <RETURN>=Select highlighted item

A pop-up menu appears (Figure 58), listing the validated scripts.

2 Select the script you want to install by highlighting its name and pressing Return.

If the installation is successful, the pop-up menu disappears and a message at the bottom of the screen tells you that installation is successful.

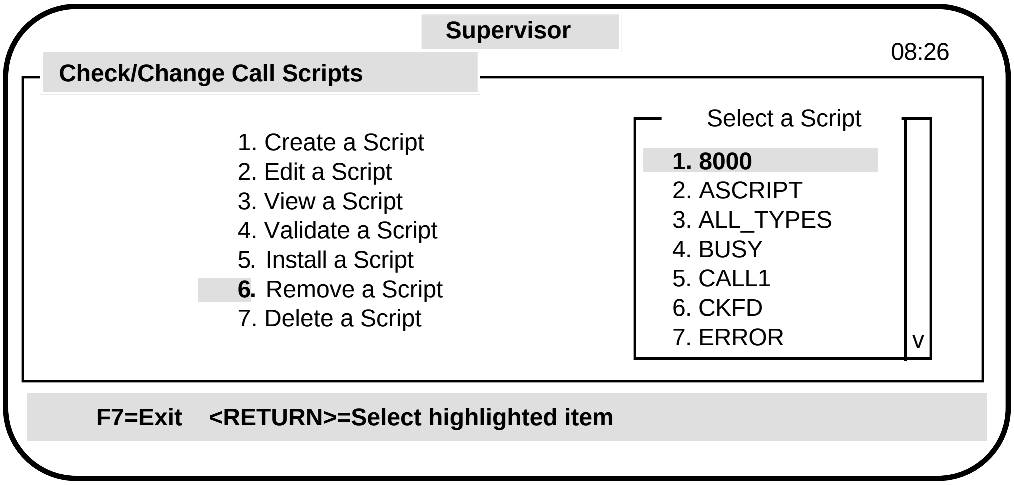
Note: A script may be successfully validated and not be successfully installed. If this happens, go into the error logs to view the error type. (See the *Customer Controlled Routing Diagnostic and Maintenance Guide,* NTP 553-3203-510.) Go back to the script editor to correct the problem, revalidate the script, and try to re-install the script.

Removing a script

Removing a script takes it out of the installed mode, but does not take it out of CCR. Calls pending in the queue are handled to completion. If a script is associated with a CDN, the association must be deleted from the Association Table before the script can be removed. For more information, see the “Associating CDNs and scripts” chapter.

1 Select the Remove a Script option from the Check/Change Call Scripts menu.

Figure 59
Remove a Script window



A pop-up menu lists all scripts that can be removed (Figure 59).

2 If the name of the script you want to remove appears in the list, select the script by highlighting the name of the script and pressing Return. A message appears, informing you that the script was removed successfully.

Or

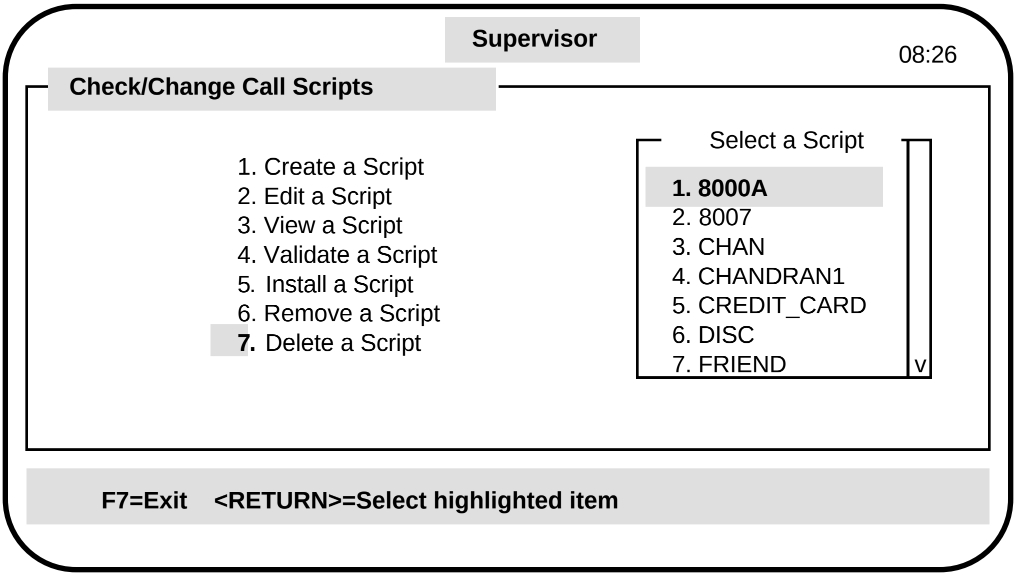
If the name of the script you want to remove does not appear in the list, the script is associated with a CDN. Delete the association (in the Association Table screen) and try again. If the script was removed successfully, the pop-up menu disappears and the Check/Change Call Scripts menu appears.

Deleting a script

A script can be deleted only if it has not been installed. If a script is associated with a CDN, the association must be deleted from the Association Table before the script can be removed and deleted. Deleting a script permanently erases it from CCR (deletes the script files from the disk).

1 Select the Delete a Script option from the Check/Change Call Scripts menu.

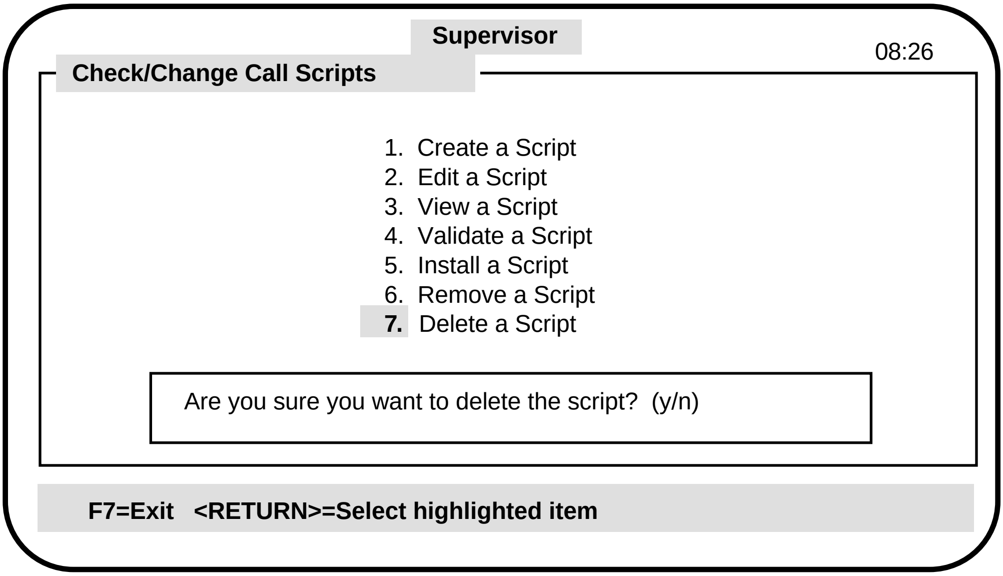
Figure 60
Delete a Script window



A pop-up menu appears listing scripts that can be deleted (Figure 60).

2 Select the script you want to delete by highlighting the script and pressing Return.

Figure 61
Delete a Script window



Are you sure you want to delete the script? (y/n)

A pop-up window, prompting you to confirm the deletion, replaces the pop-up menu (Figure 61).

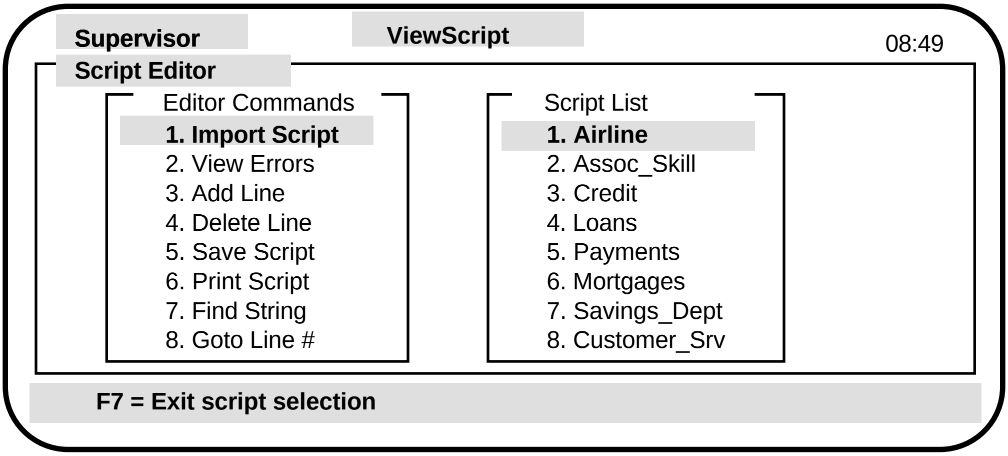
3 Verify that you want to delete this script by typing **y** and pressing Return.

When the delete process ends, the pop-up window disappears and a message appears, informing you that the script was deleted successfully.

Editing an installed and associated script

You cannot edit an installed script. To change an installed script, import a copy of it and make the changes on the imported copy. Then validate and install the new script before updating the Association Table. The new association and the new script take effect immediately on all new incoming calls. Those already queued receive the old script.

Figure 62
Import a Script window



To change an installed script, perform the following tasks:

1 Create a new script (see “Creating a script”).

2 Import the script you wish to modify (see “Editing a script”).

3 Make the changes in the new script (see “Editing a script”).

4 Validate the new script (see “Validating a script”).

5 Install the new script (see “Installing a script”).

6 Go to the Association Table.

7 Delete (or edit) the association from the old script.

8 Create an association for the new script.

9 Save and activate the modified Association Table (see the “Associating CDNs and scripts” chapter).

10 If you do not want to keep two copies, remove the old script (see “Removing a script” in this chapter).

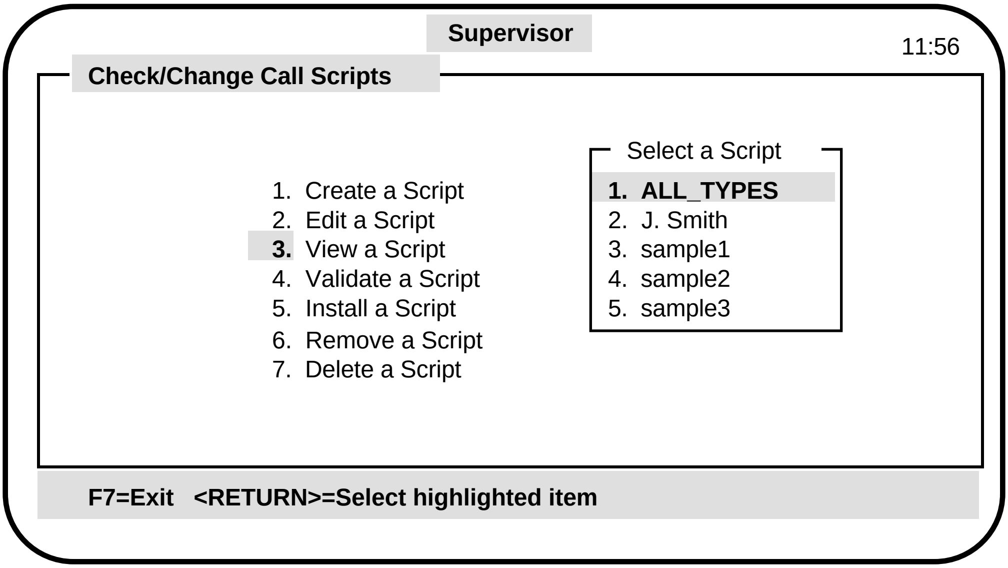
Viewing a script

Perform the following procedure to view a script.

1 Select the Check/Change Call Scripts option from the Main Menu.

2 Select option 3 (View a Script) on the Check/Change Call Scripts menu.

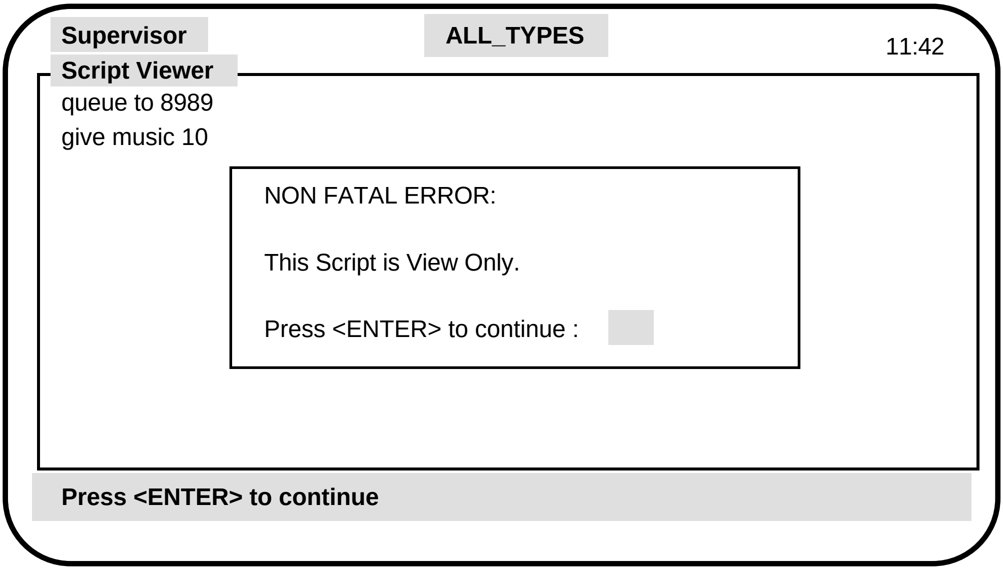
Figure 63
View a script window



A pop-up menu appears, showing a list of existing scripts (Figure 63).

3 Select the desired script. The script viewer screen appears, showing the desired script.

Figure 64
Trying to enter commands when viewing



```
Supervisor                    ALL_TYPES                     11:42
Script Viewer
queue to 8989
give music 10

          NON FATAL ERROR:

          This Script is View Only.

          Press <ENTER> to continue :

Press <ENTER> to continue
```

If you attempt to enter commands, you will receive an error message (as shown in Figure 64).

Sample scripts

Credit card company

Application overview

The financial institution has one 24-hour primary Customer Service answering group for 800 number calls from credit card customers. There is a backup group (Special Services Group) available to answer calls. There are three different numbers, distinguished by DNIS, for three levels of callers: Platinum, Gold, and Regular. The callers receive different levels of treatment as follows:

Platinum

- Queue to Customer Service Group at Priority 1
- Use the Special Services Group (backup group) at Priority 1 if no agents are available in the Customer Services Group
- Play a unique recorded announcement for these callers
- Play music

Gold

- Queue to Customer Service Group at Priority 2
- Use the Special Services Group at Priority 2 if the caller has waited 10 seconds in the Customer Services Group
- Play a unique recorded announcement for these callers
- Play music

Regular

- If it is a holiday or a weekend and there are twice as many calls queued to the Customer Service Group as there are agents logged into the Customer Service Group, then force the caller to busy
- Queue to Customer Service Group at Priority 3
- There are no backup answering positions for these callers
- If callers wait one minute, increase their priority from 3 to 2
- Play a unique recorded announcement for these callers

Variables used in the script

The following variables are used in the script:

Table 16
Variables used in credit card company sample script

Variable	Type	Class	Value*	Comments
cust_svc	ACD	Item	1001	Main Group
special_svc	ACD	Item	1002*	Backup Group
holiday	DATE	Set	01/01, 07/04, 12/25	
weekend	DAY	Set	Saturday, Sunday	
platinum_ran	RAN	Item	23	RAN Route #
gold_ran	RAN	Item	24	
regular_ran	RAN	Item	25	
platinum_dnis	DNIS	Item	5562	Platinum 800 #
gold_dnis	DNIS	Item	2323	Gold 800 #
regular_dnis	DNIS	Item	1517	Regular 800 #
soft_music	MUSIC	Item	19	Music Route #

* If you are using this as a practice script, remember to use real values for your ACD DNs.

Specific RANs needed

platinum_ran
"Thank you for calling. All of our agents are busy. Your call is important to us so please stay on the line."

gold_ran
"Thank you for calling. All of our agents are busy. Your call will be answered as soon as possible so please stay on the line."

regular_ran
"Thank you for calling. All calls will be answered in the order received."

Example script

```
/* Send the caller to the appropriate section for handling */

GOTO Platinum_Callers IF DNIS = platinum_dnis
GOTO Gold_Callers IF DNIS = gold_dnis

/* The following instruction will only be used if the call arrives to this
call script without a DNIS number or with an unexpected DNIS number. */

GOTO Regular_Callers

/* Call handling section for Platinum Cardholders */

/* Calls will be given to the Customer Service Group first. Calls will then be
queued to the Special Services Group if they were not immediately answered in
the Customer Service Group. Calls will then be simultaneously queued to both
groups until their calls are answered. If callers wait, they will hear a
special RAN followed by music */

SECTION Platinum_Callers
       QUEUE TO cust_svc WITH PRIORITY 1
       QUEUE TO special_svc WITH PRIORITY 1
       GIVE RAN platinum_ran
       GIVE MUSIC soft_music
       QUIT
```

```
/* Call handling section for Gold Cardholders */
/* Calls will be given to the Customer Service Group first. Calls will be
given a unique RAN followed by music. If calls wait more than 30 seconds in
the Customer Service Group, they will be queued to the Special Services Group.
Calls will then simultaneously be queued to both groups and will continue to
hear music until they are answered. */

SECTION Gold_Callers
       QUEUE TO cust_svc WITH PRIORITY 2
       GIVE RAN gold_ran
       GIVE MUSIC soft_music
SECTION Loop
       WAIT 2
       GOTO Queue_Secondary IF Age Of Call > 30
       GOTO Loop

/* This section will be reached only if the call waits more than 30 seconds.
*/

SECTION Queue_Secondary
       QUEUE TO special_svc WITH PRIORITY 2
       QUIT
```

```
/* Call handling section for Regular Cardholders */

/* If it is a holiday or a weekend and there are twice as many calls queued to
the Customer Service Group as there are agents logged into the Customer
Service Group, then the call is forced to busy. For example, as long as there
are 30 agents logged in and there are 20 calls in queue, then any additional
calls will be forced to busy. Calls that are not forced to busy are queued to
the Customer Service Group and given a unique RAN. If a call waits more than
60 seconds in the Customer Service Group queue, the priority of the call is
increased to level 2. */

SECTION Regular_Callers
       FORCE BUSY IF         (Day Of Year = holiday OR
                              Day Of Week = weekend) AND
                             (Total Queued Calls cust_svc >
                              (2 * Logged Agents cust_svc))
       QUEUE TO cust_svc WITH PRIORITY 3
       GIVE RAN regular_ran
SECTION Regular_Loop
       WAIT 2
       GOTO Change_Priority IF Age Of Call > 60
       GOTO Regular_Loop

/* This section will only be reached if the call waits more than 60 seconds
without being answered. */

SECTION Change_Priority
       QUEUE TO cust_svc WITH PRIORITY 2
       QUIT
```

Utility company

Application overview

The utility company has one 24-hour primary Customer Service answering group for all calls about service. During power outages, calls are pre-screened. Calls in the affected NPANXX areas hear a recording asking them to hang up if they are only calling to report an outage in the affected area. If the caller stays on the line, the call is presented to an agent. Calls from hospitals, based on their CLID, will always be answered ahead of all other calls.

Hospitals

- Queue to Customer Service Group at Priority 1
- Play a unique recorded announcement for these callers

Callers with NPANXX in Outage area

- Play RAN asking customer to hang up if they are calling to report an outage in their area
- If the customer stays on the line, Queue to Customer Service Group at Priority 2
- Play the recorded announcement for these callers
- Play music

Other Callers

- Queue to Customer Service Group at Priority 2
- Play music

Variables needed for the application

The following variables are used in the script:

Table 17
Variables used in utility company sample script

Variable	Type	Class	Value	Comments
cust_svc	ACD	Item	1001	Main Group
hospital_clid	CLID	Set	4155551000	Local Hospitals
			4085551234	
			4085552323	
			4085557389	
outage_area	NPANXX	Set	408732	This changes, based on current outages
			408733	
hospital_ran	RAN	Item	23	
outage_ran	RAN	Item	24	This changes, based on current situation
standard_ran	RAN	Item	25	
soft_music	MUSIC	Item	19	Music Trunk

Specific RANs needed

hospital_ran

"Providing power to hospitals is of utmost concern to us. Your call will be answered soon, so please stay on the line."

outage_ran

"Thank you for calling LG&E. We are currently aware of a power outage in the Lexington area. If you are calling to report this outage, please hang up. If you have other business about which you wish to speak to a customer service representative, please stay on the line and your call will be answered as soon as possible."

standard_ran

"All of our agents are busy, please stay on the line as all calls will be answered in the order they have been received."

Example script

```
/* This section sends the caller to the appropriate section for handling */

GOTO Hospital IF CLID = hospital_clid
GOTO Outage IF NPANXX = outage_area
GOTO Regular_Calls

/* Call handling section for Hospitals contained in the hospital_clid list.
This section gives the unique RAN for hospitals. */

SECTION Hospital
       QUEUE TO cust_svc WITH PRIORITY 1
       GIVE RAN hospital_ran
       QUIT

/* Call handling section for callers in the outage area. This section gives
callers a special announcement. If the caller stays on the line after the RAN
has completed, the call is handled as a regular call. */

SECTION Outage
       GIVE RAN outage_ran
       WAIT 6
```

```
/* Call handling section for Regular callers (not in the outage area or a
hospital) Notice that there is no QUIT at the end of SECTION Outage, so that
the call script flows into the this section */

SECTION Regular_Calls
       QUEUE TO cust_svc WITH PRIORITY 2
       GIVE RAN standard_ran
       GIVE MUSIC soft_music
       QUIT
```

Script formatting conventions

Follow a consistent format to ensure that scripts can be easily read and interpreted by you and others at a later date.

Commands

Type commands in all uppercase.

```
QUEUE TO main_acd
```

Intrinsics

Type intrinsics with initial caps followed by lowercase.

```
IF Age Of Call > 30
```

Logical Operators (AND, NOT, OR)

Type in all uppercase.

```
GOTO Night_Treatment IF Night Service main_acd AND
    Night Service backup
```

Parentheses

Remember that parentheses affect the order in which the expressions in a statement are processed. Expressions in parentheses are processed before other expressions in a statement. Check to make sure that the parentheses in your script correctly reflect the order of call processing you want.

Section names

Type with initial caps followed by lowercase.

```
SECTION Night_Treatment
```

(Remember that the word “SECTION” is a command and what follows is the section name.)

Statements

When writing scripts, be sure to format statements so that the way they read is the way they operate. For example, the following should be written in only one line:

```
GOTO Night_Treatment IF Night Service main_acd OR
   Day Of Year = holiday
```

Variables

Type variables in all lowercase.

```
GIVE RAN closed
```

Blank lines, indenting, adding comments

To improve readability, be sure to leave blank lines between SECTIONs, and before and after comments. Indent statements after SECTIONs to make it easier to identify SECTIONs. Also, indent statements that extend beyond one line. Comments help others understand your intention for a SECTION and—while not required—can be extremely helpful in understanding the original purpose of the SECTION when reviewing a script at a later date.

```
/* This section of the script queues calls to the main_acd
   group only if the main_acd queue is not in Night Service,
   or is not a holiday, or it is not a weekend. The
   variables holiday, weekend, and main_acd are defined in
   the Variable Table. */

   GOTO Night_Treatment IF Night Service main_acd OR
      Day Of Year = holiday

   QUEUE TO main_acd

   QUIT

/* This section of the script plays the recorded announcement
   informing the caller that the business is closed and
   disconnects the call. */

   SECTION Night_Treatment

      GIVE RAN closed

      FORCE DISCONNECT
```

Script writing tips

This section gives recommendations for writing effective scripts. The tips are presented in alphabetical order. They include:

- ACD DNs
- Check for most likely conditions first
- Check ACD DNs for Night Service
- Consider the caller
- Default treatments
- First treatments
- Giving tones
- GOTO commands
- High traffic conditions
- Improve call processing efficiency
- Loops
- Queue unconditionally
- Ranges
- SECTIONs
- Time comparisons
- Variables
- WAIT

ACD DNs

Remember to include an ACD DN for the following intrinsics:

- Logged Agents
- Total Queued Calls
- Oldest Call
- Night Service
- Idle Agents

Check for most likely conditions first

If you are checking multiple conditions in a script, it is more efficient to check for the conditions most likely to occur first. For example, if a script handles three types of callers differently, based on DNIS, it would be most efficient to check the most commonly called number followed by the second most commonly called number, followed by the third number.

```
QUEUE TO main_acd WITH PRIORITY 3 IF DNIS = 5604

QUEUE TO main_acd WITH PRIORITY 2 IF DNIS = 5610

QUEUE TO main_acd WITH PRIORITY 1 IF DNIS = 2512
```

In the above example 5604 is called most often, followed by 5610 and 2512.

Check ACD DNs for Night Service

Attempts to queue calls to an ACD queue that is in Night Service will be rejected, leaving callers in a ring no answer state. If there is any possibility that a queue will not be staffed (for example, if the queue does not operate 24 hours a day), start all your scripts with a check for Night Service.

ACD queues are put into Night Service when all agents have logged out by using the Make Set Busy (MSB) key, or the queue has been put into Night Service using the Night Service (NSVC) key on the supervisor telephone.

> ***Note:*** If the queue is in Transition Mode (by dialing “T” while pressing the NSVC key), all pending calls receive regular treatment, but no new calls can be queued to that ACD DN until it is taken out of Night Service. If an ACD DN enters Night Service without this transition (by dialing “N” while pressing the NSVC key), the switch removes all pending calls from the queue.

Use the Night Service intrinsic rather than the Logged Agents intrinsic to check queues. The Night Service intrinsic is true for all queues in Night Service—no matter how they got there. If the queue is in Transition Mode or Night Service by means of the supervisor NSVC key, Logged Agents could be greater than 0, indicating that agents are available. However, calls newly queued to that ACD DN will not be accepted by those agents. A valid test would be:

```
IF Night Service main_acd AND Night Service backup_group
```

There are several ways to follow a check for Night Service. These include:

- QUEUE TO an ACD DN still in day service
- ROUTE TO an attendant
- FORCE BUSY
- FORCE DISCONNECT (preceded by a GIVE RAN)

This ensures the best treatment to the caller and is the most efficient method of handling call processing.

Here is an example of a script beginning with a Night Service check:

```
GOTO Night_Treatment IF Night Service main_acd

FORCE BUSY IF Total Queued Calls main_acd > (2 * Logged
   Agents main_acd)

ROUTE TO attendant IF CLID = vip  /* attendant and vip
   are variables */

QUEUE TO main_acd

QUIT

SECTION Night_Treatment

   GIVE RAN closed

   FORCE DISCONNECT
```

Note that in the above example, even though the Night_Treatment SECTION comes at the end of the script, the call will be played a recorded announcement and disconnected right away if the main_acd group is in Night Service.

Consider the caller

Consider what the caller will hear (for example, ringing, silence, music or some other tone) at all points in the script. Note that the GIVE MUSIC command does not usually need to be repeated in a script because music persists after other commands (except GIVE SILENCE or GIVE RINGBACK). For example:

```
QUEUE TO main_acd WITH PRIORITY 3

GIVE RAN agents_busy

GIVE MUSIC local_station

WAIT 30

GIVE RAN agents_still_busy
```

In the above example, assuming an agent does not become available, callers receive the following treatments:

- they would be placed in the main_acd queue
- they would hear the entire RAN
- they would hear music for 30 seconds
- they would hear the entire RAN agents_still_busy
- they would hear music again until answered

Default treatments

If a call is initially queued to an ACD DN by the QUEUE TO command, ringback is automatically applied. If a queued call is no longer receiving ringback, it is automatically provided when the call is presented to an agent.

Remember that the default treatment after a caller hears RAN or has completed an IVR session is silence. If you want the caller to hear anything else (music or ringback) you must specify it using the GIVE command.

First treatments

CCR will not validate scripts that use GIVE SILENCE, WAIT, or FORCE DISCONNECT unconditionally as first treatment in a script. If the script's first treatment uses one of these commands with a condition, CCR allows the statement, but gives a warning message.

If you use the FORCE BUSY command, it should be the first treatment in the script.

Giving tones

CCR automatically gives ringback when a call enters a queue and when a call is presented to an agent. The caller often hears only a burst of ringback in these situations, depending on the delay—or lack of delay—in providing the next treatment. Because a script cannot control that delay, it does not completely control the tones heard by the caller. However, you can decide whether to accept the burst of ringback, to try to avoid it for most callers, or to ensure that it lasts long enough to sound natural to the caller. The examples given in this section illustrate different methods of giving tones, and describe the possible results of each method.

Example 1

```
QUEUE TO 8900

GIVE MUSIC music_route
```

This script gives music to a caller waiting for an agent to answer. When the QUEUE TO statement is executed, the caller may hear a burst of the ringback before hearing the specified music. The caller hears ringback again when the call is presented to an agent. Use this method of giving tones when you expect most calls to be answered immediately, rather than queued.

Example 2

```
GIVE MUSIC music_route

QUEUE TO 8900
```

Example 2 shows how to give tones immediately when the call is queued. It gives music before CCR has a chance to provide automatic ringback, and consequently avoids the burst of ringback. However, if the call is presented to an agent without delay, the caller may hear a burst of music before hearing the ringback provided when the call is presented to an agent.

Use the method of giving tones illustrated in Example 2 when you expect most calls to be queued rather than answered immediately.

Example 3

```
QUEUE TO 8900

WAIT 6

GIVE MUSIC music_route
```

Alternatively, give the caller a chance to hear a full cycle of ringback before the desired tones. Example 3 forces a six-second delay before giving tones to ensure a more natural-sounding ringback cycle.

GOTO commands

Make sure that whenever you use a GOTO command that you have created a SECTION with that label. Also, remember to write the GOTO command as one word, not two words.

High traffic conditions

If the script starts with a significant number of IF tests before the first treatment of a call is determined, the caller may hear a second or two of silence before CCR gives treatment (ringback or another tone). Under high traffic conditions, the four-second response timer may expire, causing calls to default intermittently. It would take more than 100 IF tests to present a problem.

If this situation is a problem, write the script so that it decides on the tone first. Begin a large number of IF test only when the call is safely hearing a tone and thus in no danger of defaulting. For example:

```
FORCE BUSY IF CLID = blacklist_callers

GIVE RINGBACK

GOTO california IF NPA = 415, 408, 510, 213, 818, 714,
    619

GOTO oregon IF NPA = . . .

GOTO arizona IF NPA = . . .
```

Improve call processing efficiency

Use the following tips to improve call processing efficiency:

- Avoid putting statements that do not need to be repeated in a loop. In the next example, the QUEUE TO statement is executed before the loop, which repeats the RAN. It would be possible to include the QUEUE TO statement in the loop, but this would cause the QUEUE TO statement to be repeated unnecessarily. A QUEUE TO statement that is repeated multiple times will be ignored, and will create a less efficient script.

  ```
  QUEUE TO main_acd WITH PRIORITY 3

  /* This section repeats a recorded announcement every 30
      seconds until the call is answered. */

  Section Play_RAN

      Wait 30

      GIVE RAN agents_busy

      GOTO Play_RAN
  ```

- Limit the number of ACD DNs on which the CCR application is drawing statistics. ACD statistics are requested by the CCR application for all ACD DNs used as parameters in queue intrinsics or as ACD variables in the Variable Table. To limit requests for statistics:

  — If an ACD variable is referenced from a script, but only in a QUEUE TO or a REMOVE FROM statement, insert the ACD DN as a constant in the script, rather than as a variable.

  — If an ACD variable is not used in any script, remove it from the Variable Table.

- If you queue incoming calls to a number of ACD DNs unconditionally, follow each QUEUE TO statement with a short WAIT to give the Meridian 1 a chance to find an Idle Agent before executing the next QUEUE TO statement. For more information, see the “QUEUE TO” section in the “Script commands” chapter.

Loops

A loop is a SECTION of a script that is repeated over and over, until a call is answered. It is useful for checking time intrinsics such as the Age of Call or for playing the same recorded announcement more than once to a caller. Loops should always include a WAIT statement, as shown in Example 1.

Example 1

```
QUEUE TO main_acd WITH PRIORITY 3

GIVE RAN

/* This section repeats the second recorded announcement
    after a 30 second pause until the call is answered. */

SECTION Play_2nd_RAN

    Wait 30

    GIVE RAN agents_still_busy

    GOTO Play_2nd_RAN
```

Take care when using a loop to check whether an intrinsic meets a given condition. This is important because an intrinsic (Age of Call, for example) is equal to a particular value for only one second in time. For example, the statement

```
IF Age Of Call = 10
```

is true only if the ACD DN happens to be checked when the call had waited exactly 10 seconds. A better way to do this is to set a condition using an operator that targets a range of time, and place the conditional statement in a loop that is repeated at frequent intervals, beginning at a time just prior to that specified in the condition.

In Example 2, the priority of the call does not change until the call has waited more than 2 minutes. The conditional QUEUE TO statement is required in the loop (SECTION Check_Age), which begins execution only when the call has waited almost 2 minutes. The first WAIT statement prevents the system from beginning the loop until the check on the ACD DN has some possibility of being true.

Example 2

```
QUEUE TO main_acd WITH PRIORITY 3

WAIT 110

/* This section of the script test the Age of Call every
   two seconds only until the call has been queued at
   priority 3 for 120 seconds.

SECTION Check_Age

   GOTO Change_Priority IF Age Of Call > 120

   WAIT 2

   GOTO Check_Age

/* This section of the script increases the priority of a
   call from priority 3 to priority 2 if the call waits
   more than 120 seconds without being answered. */

SECTION Change_Priority

   QUEUE TO main_acd WITH PRIORITY 2

   QUIT
```

Queue unconditionally

Calls should always be queued to at least one queue unconditionally. This will prevent calls from being left without being answered in case none of the specified conditions are met.

The following is a poor example:

```
GOTO Night_Treatment IF Night Service main_acd

FORCE BUSY IF Total Queued Calls main_acd > (2 * Logged
    Agents main_acd)

ROUTE TO attendant IF CLID = vip  /* attendant and vip
    are variables */

QUEUE TO main_acd WITH PRIORITY 3 IF DNIS = 5604

QUEUE TO main_acd WITH PRIORITY 2 IF DNIS = 5610

QUEUE TO main_acd WITH PRIORITY 1 IF DNIS = 2512

QUIT

SECTION Night_Treatment

    GIVE RAN closed

    FORCE DISCONNECT
```

In the above example, if a call has no DNIS number and does not contain a CLID in the VIP list, it will receive no treatment at all because all the commands are conditional.

A better script would include the statement

```
QUEUE TO main_acd WITH PRIORITY 4 IF DNIS <> 5604, 5610,
    2512
```

prior to the QUIT statement,. This would ensure that all calls *not* having the specified DNIS numbers would be queued.

Ranges

Ranges for variables include all numbers in the range including the starting and ending range values. For example, a range of CLID numbers shown as 1..4155552323 includes every CLID from 1 to 4155552323 (these include 1, 3, 10, 415, 4152, ..., 4155552323). Note that the starting and ending range values should usually have the same number of digits (for example, 4155552233.. 4155552134). Use caution in specifying ranges such as the above example (1..4155552323) to ensure that your range includes only the intended values.

SECTIONs

Use a SECTION to provide treatment when more than one action is required if a condition is met. For example, the correct way to write a script that treats VIP callers in a special way would be:

Example 1

```
GOTO VIP_Treatment IF CLID = vip_list

QUEUE TO main_acd WITH PRIORITY 3

GIVE RAN agents_busy

QUIT

/* This section of the call script gives special
   treatment to calls in the VIP variable list */

SECTION VIP_Treatment

   QUEUE TO main_acd WITH PRIORITY 1

   GIVE RAN special_callers

   QUIT
```

Example 2

The *less desirable* way to write the script would be:

```
QUEUE TO main_acd WITH PRIORITY 1 IF CLID = vip_list

GIVE RAN special_callers IF CLID = vip_list

QUEUE TO main_acd WITH PRIORITY 3

GIVE RAN agents_busy

QUIT
```

In the above example if you had forgotten to check IF CLID = vip_list in the second statement, all callers would have received the special RAN. For this reason, it is always better to use a SECTION to provide multiple actions for the same type of call.

Time comparisons

To be meaningful, time comparisons should include a >= or <=. For example, given the following statements:

```
IF Time Of Day = 08:00

IF Time Of Day >= 08:00 AND Time Of Day <= 09:00
```

The first expression is true for one minute; the second is true for one hour.

Variables

Variables make your scripts easier to modify. For example, if you use the variable main_acd for your main agent group (2500), and the numbering plan in the Meridian 1 software changes the agent group number to 2600, you need to update only the variable in the Variable Table; you do not have to modify all your scripts.

Variables cannot be key words or SECTION labels. They must be unique.

To avoid warning messages, define all variables prior to writing your script. If an ACD variable is not referenced at all by any script, take it out of the Variable Table. If an ACD variable is referenced by a script, but only in a QUEUE TO or a REMOVE FROM statement, insert the ACD DN as a constant in the script, rather than as a variable. All ACD variables draw statistics, but a QUEUE TO or a REMOVE FROM statement that references a constant rather than a variable does not draw statistics.

WAIT

Using a WAIT statement immediately prior to a QUIT statement has no effect on the script. For example:

```
GIVE RAN ran_route

WAIT 20

QUIT
```

has the same effect on the call as:

```
GIVE RAN ran_route

QUIT
```

Example script

The following is an example of a script that combines the above tips. In this case the call center is not open 24 hours a day.

```
/* Check to see if both groups that can answer calls are in
   Night Service and disconnect the caller if they are */

GOTO Night_Treatment IF Night Service main_acd AND Night
   Service backup_group

/* Send VIP calls to be handled in a special way. This is
   done at the beginning to ensure that VIP callers are
   never given a busy signal */

GOTO Special_Handling IF CLID = vip

/* Check to see if there are already more than twice as many
   calls queued as there are agents logged in and force the
   caller to busy if this is true */

FORCE BUSY IF Total Queued Calls main_acd > (2 * Logged
   Agents  main_acd)

/* Queue the caller to the main group and give an
   announcement followed by music */

QUEUE TO main_acd WITH PRIORITY 3

WAIT 6          /* Ensures at least a full cycle of the
                default ringback treatment, in the event
                that the call is not answered immediately */

GIVE RAN agents_busy

GIVE MUSIC 6
```

```
/* This section of the script increases the priority of non-
   VIP callers and queues them to the back_up group if they
   have waited more than 2 minutes. It also plays a second
   recorded announcement after a 20 second pause. The caller
   will hear music after the RAN completes. */

SECTION Check_Age

   QUEUE TO main_acd WITH PRIORITY 1 IF Age Of Call > 120

   QUEUE TO backup_group WITH PRIORITY 3 IF Age Of Call >
     120

   WAIT 20

   GIVE RAN agents_still_busy

   GOTO Check_Age

/* This section of the script will only be reached if the
   caller has a CLID that is contained in the vip variable
   list. These callers are queued with high priority to 2
   queues and given a special RAN. */

SECTION Special_Handling

   QUEUE TO main_acd WITH PRIORITY 1

   QUEUE TO backup_group WITH PRIORITY 1

   GIVE RAN you_are_special

   GIVE MUSIC 6

   QUIT

/* This section of the script plays the closed announcement
   and disconnects the caller. This will only happen if both
   the main group and the backup group are in Night Service.
   */

SECTION Night_Treatment

   GIVE RAN closed

   FORCE DISCONNECT
```

Associating CDNs and scripts

Accessing the Association Table screen

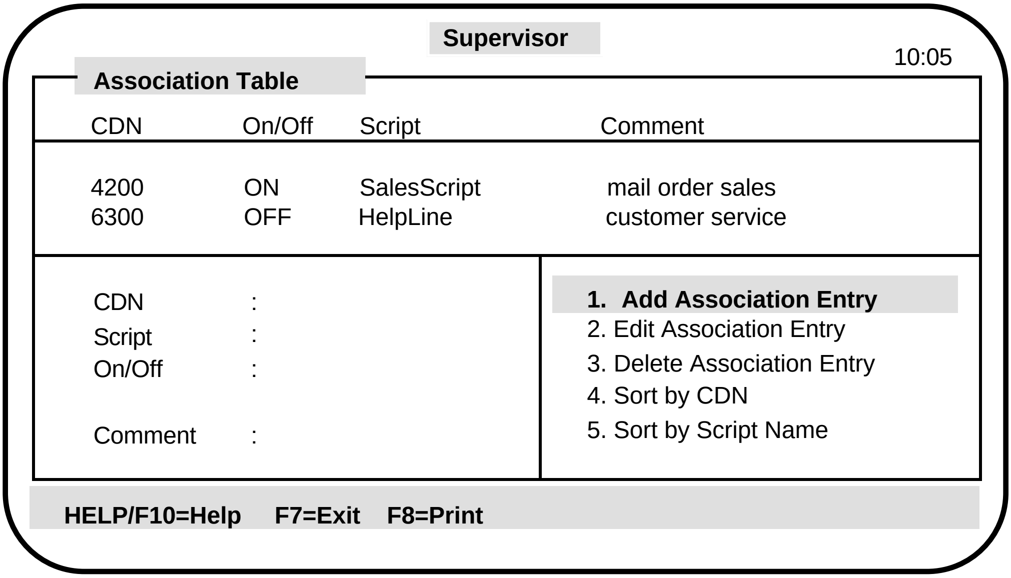
Once a script has been written, validated, and installed, it must be associated to a Control DN (CDN) so it can handle and treat incoming calls. One script can work for many CDNs, but each CDN can use only one script at a time.

The number of CDNs supported by CCR is limited only by the number of CDNs that can be configured on the Meridian 1.

To access the Association Table screen, follow the steps below:

1 Select the Check/Change Associations Table option from the Main Menu.

Figure 65
Add Association Entry window



Supervisor

10:05

Association Table

CDN	On/Off	Script	Comment
4200	ON	SalesScript	mail order sales
6300	OFF	HelpLine	customer service

CDN :
Script :
On/Off :
Comment :

1. Add Association Entry
2. Edit Association Entry
3. Delete Association Entry
4. Sort by CDN
5. Sort by Script Name

HELP/F10=Help F7=Exit F8=Print

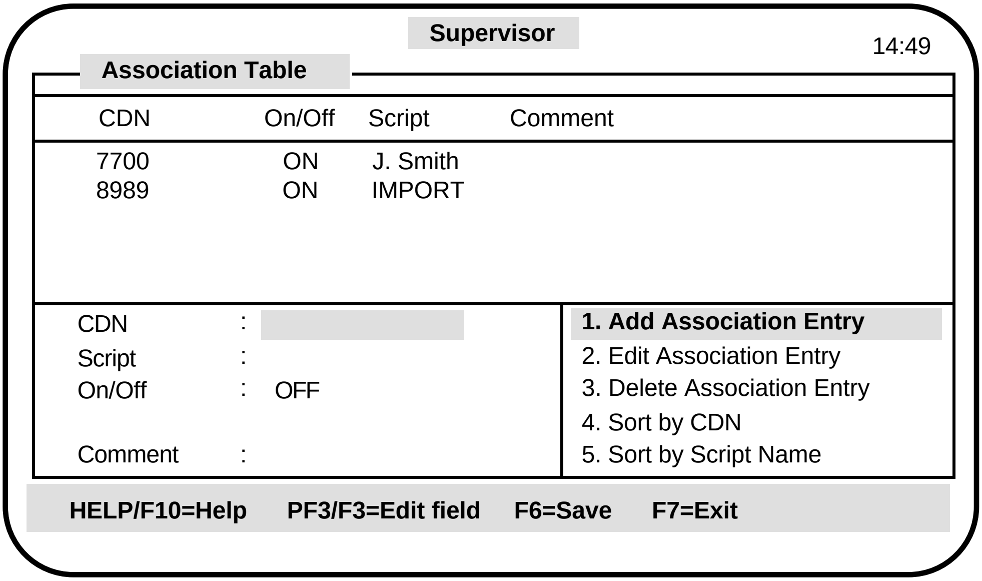
The Association Table screen appears, with the Add Association Entry highlighted as illustrated in Figure 65.

Adding an association

To add a new CDN/script association, follow the steps below:

1 Select the Add Association Entry option from the Association Table menu.

Figure 66
Add Association Entry window



CDN	On/Off	Script	Comment
7700	ON	J. Smith	
8989	ON	IMPORT	

CDN :
Script :
On/Off : OFF
Comment :

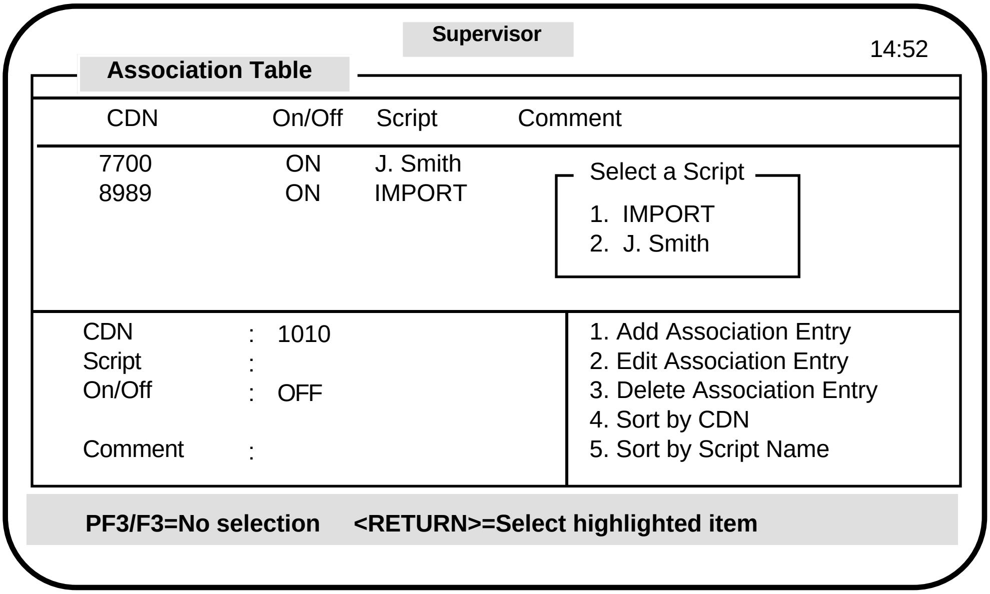
The CDN field is highlighted (Figure 66).

2 Type the CDN you want to associate. For example, type **1010**.

3 Press Return to move to the Script field.

4 Press PF3 to view a pop-up menu displaying the scripts available for association.

Figure 67
Select a Script pop-up menu



Supervisor

14:52

Association Table

CDN	On/Off	Script	Comment
7700	ON	J. Smith	
8989	ON	IMPORT	

Select a Script

1. IMPORT
2. J. Smith

CDN : 1010
Script :
On/Off : OFF

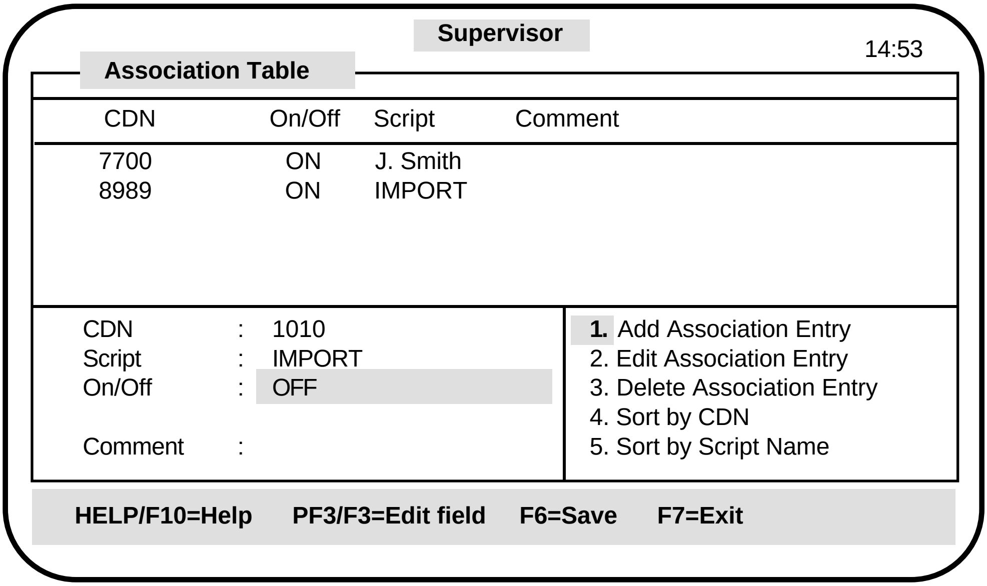
Comment :

1. Add Association Entry
2. Edit Association Entry
3. Delete Association Entry
4. Sort by CDN
5. Sort by Script Name

PF3/F3=No selection <RETURN>=Select highlighted item

5 Select the script to be associated and press Return.

Figure 68
Add Association On/Off screen



Supervisor 14:53

Association Table

CDN	On/Off	Script	Comment
7700	ON	J. Smith	
8989	ON	IMPORT	

CDN : 1010
Script : IMPORT
On/Off : OFF

Comment :

1. Add Association Entry
2. Edit Association Entry
3. Delete Association Entry
4. Sort by CDN
5. Sort by Script Name

HELP/F10=Help PF3/F3=Edit field F6=Save F7=Exit

The Association Table screen displays the selected script in the Script field and the highlight bar moves to the On/Off field.

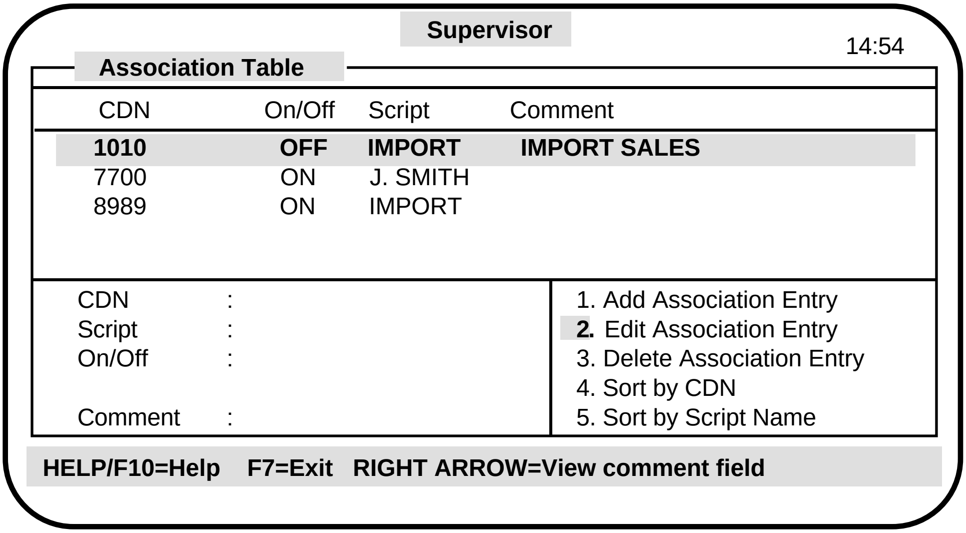
6 Type **ON** if you want the association active or type **OFF** if you want the association inactive and press Return.

CAUTION

For the association to become active, the On/Off field must read ON, and the CDN must be defined on the Meridian 1 with the overlay 23 CNTL prompt set to YES.

7 Type a comment describing the association and press Return. While optional, this information can be helpful.

Figure 69
Add a comment to Association Entry screen



Supervisor

14:54

Association Table

CDN	On/Off	Script	Comment
1010	**OFF**	**IMPORT**	**IMPORT SALES**
7700	ON	J. SMITH	
8989	ON	IMPORT	

CDN :
Script :
On/Off :

Comment :

1. Add Association Entry
2. Edit Association Entry
3. Delete Association Entry
4. Sort by CDN
5. Sort by Script Name

HELP/F10=Help F7=Exit RIGHT ARROW=View comment field

8 Press F6 to save this association into the Association Table.

9 Press F6 again to activate the Association Table.

Note: If the CDN you just entered is not valid—or if the AML was not up when you updated and activated the Association Table—an error box (similar to the call processing error box) appears. The message doesn't tell you which CDN is bad; it asks you to go to the CCR error log for details. To remove the error box from the screen, press Return.

⚠ **CAUTION**

It is recommended that scripts be tested with test traffic before live traffic is sent to the CDN.

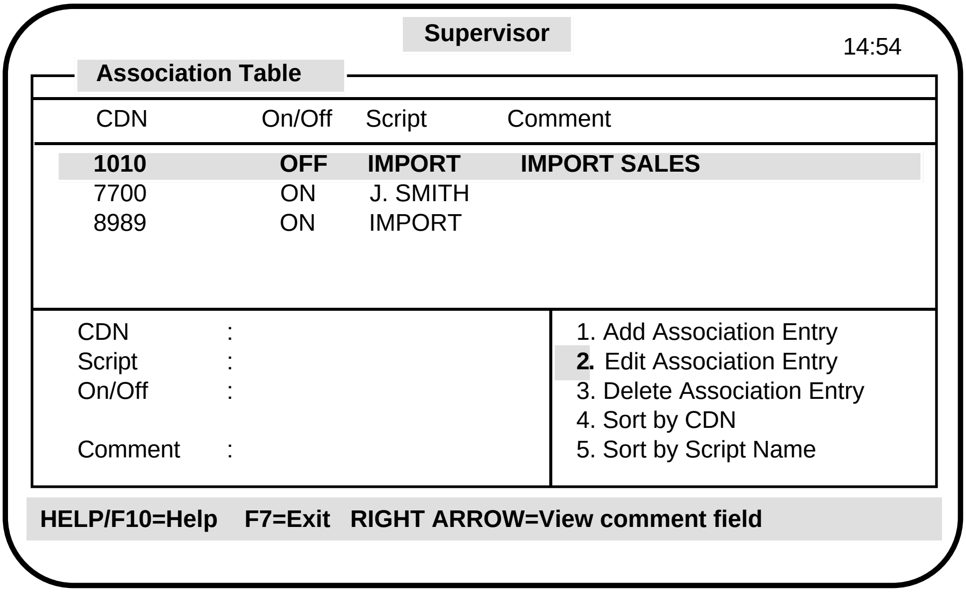
Editing an association

Associations can be changed at any time. All calls receive treatment according to the script currently associated with the CDN. A new association takes effect immediately upon activating it and new calls are treated according to the new association's script. However, existing calls use the previous script.

To modify an existing CDN/script association:

1 Select the Edit Association Entry option from the Association Table menu.

Figure 70
Edit Association Entry screen



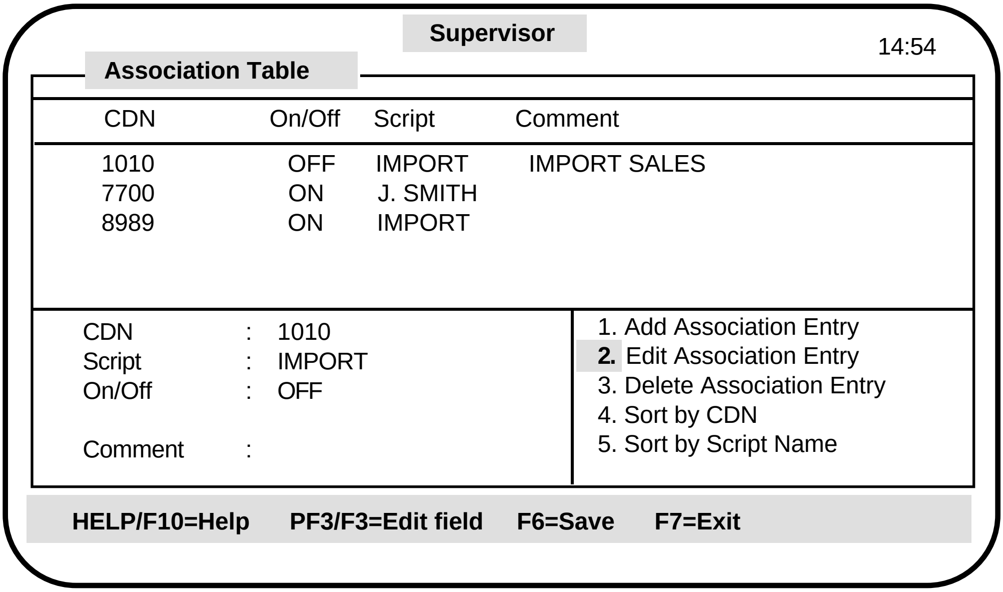
```
                        Supervisor
                                                        14:54
 Association Table

   CDN             On/Off   Script        Comment

   1010             OFF     IMPORT        IMPORT SALES
   7700             ON      J. SMITH
   8989             ON      IMPORT


  CDN          :                      1. Add Association Entry
  Script       :                      2. Edit Association Entry
  On/Off       :                      3. Delete Association Entry
                                      4. Sort by CDN
  Comment      :                      5. Sort by Script Name

 HELP/F10=Help   F7=Exit   RIGHT ARROW=View comment field
```

This places you in the selection pane with the first association highlighted (Figure 70).

2 Use the up and down arrow keys to scroll to the association you want to change.

3 Select the association and press Return.

Figure 71
Edit Association Entry screen



Supervisor 14:54

Association Table

CDN	On/Off	Script	Comment
1010	OFF	IMPORT	IMPORT SALES
7700	ON	J. SMITH	
8989	ON	IMPORT	

```
CDN        :  1010
Script     :  IMPORT
On/Off     :  OFF

Comment    :
```

1. Add Association Entry
2. Edit Association Entry
3. Delete Association Entry
4. Sort by CDN
5. Sort by Script Name

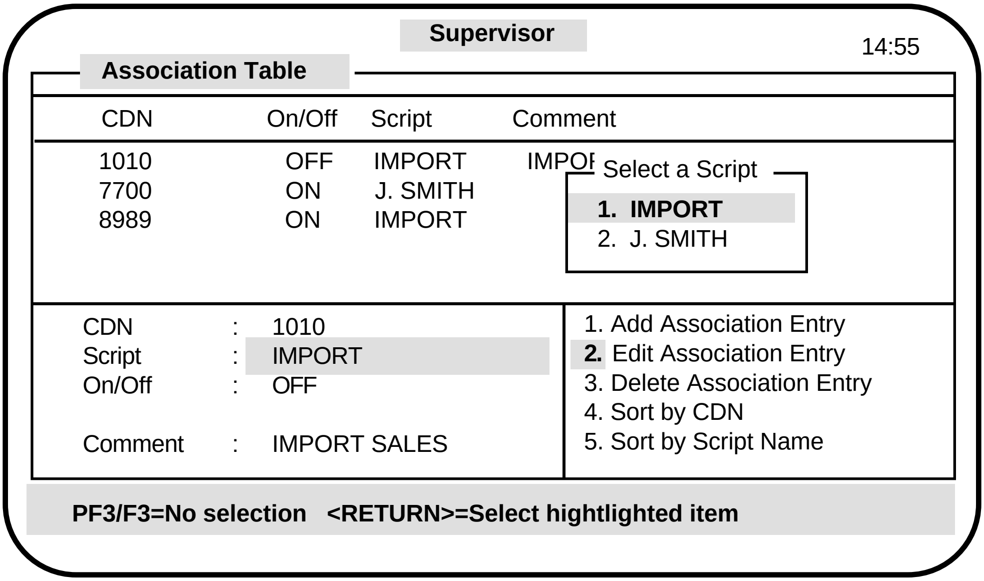
HELP/F10=Help   PF3/F3=Edit field   F6=Save   F7=Exit

4 To change the script associated with the CDN, press PF3/F3.

A pop-up menu displays the scripts available for association.

5 Select the script you want to associate with the CDN.

Figure 72
Select a Script pop-up menu



Supervisor

14:55

Association Table

CDN	On/Off	Script	Comment
1010	OFF	IMPORT	IMPOI
7700	ON	J. SMITH	
8989	ON	IMPORT	

Select a Script

1. IMPORT
2. J. SMITH

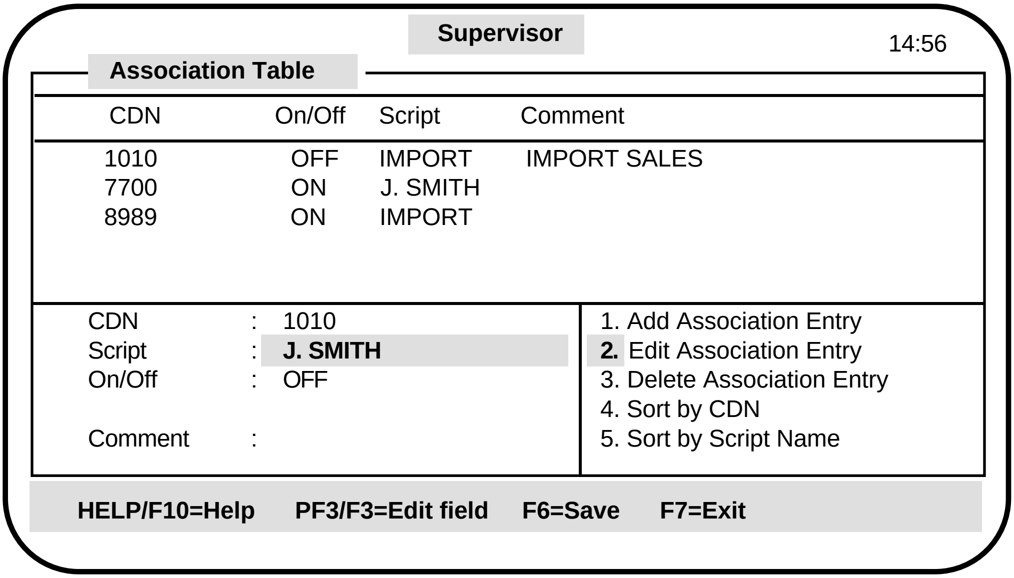
CDN	:	1010
Script	:	IMPORT
On/Off	:	OFF
Comment	:	IMPORT SALES

1. Add Association Entry
2. Edit Association Entry
3. Delete Association Entry
4. Sort by CDN
5. Sort by Script Name

PF3/F3=No selection <RETURN>=Select hightlighted item

You are now in the edit pane. The fields are completed and the Script field is highlighted (Figure 72).

Figure 73
Edit an association entry window



```
                                   Supervisor
                                                                 14:56
   Association Table
     CDN          On/Off   Script       Comment
     1010          OFF     IMPORT       IMPORT SALES
     7700          ON      J. SMITH
     8989          ON      IMPORT

   CDN       :  1010                   1. Add Association Entry
   Script    :  J. SMITH               2. Edit Association Entry
   On/Off    :  OFF                    3. Delete Association Entry
                                       4. Sort by CDN
   Comment   :                         5. Sort by Script Name

   HELP/F10=Help   PF3/F3=Edit field   F6=Save   F7=Exit
```

6 Use the arrow keys to move to the field you want to edit and press Return.

Note: You cannot edit the CDN field. To change a CDN, you must add a new association and delete the old association.

7 Press F6 to save this association into the Association Table.

8 Press F6 again to activate all the associations in the Association Table.

Deleting an association

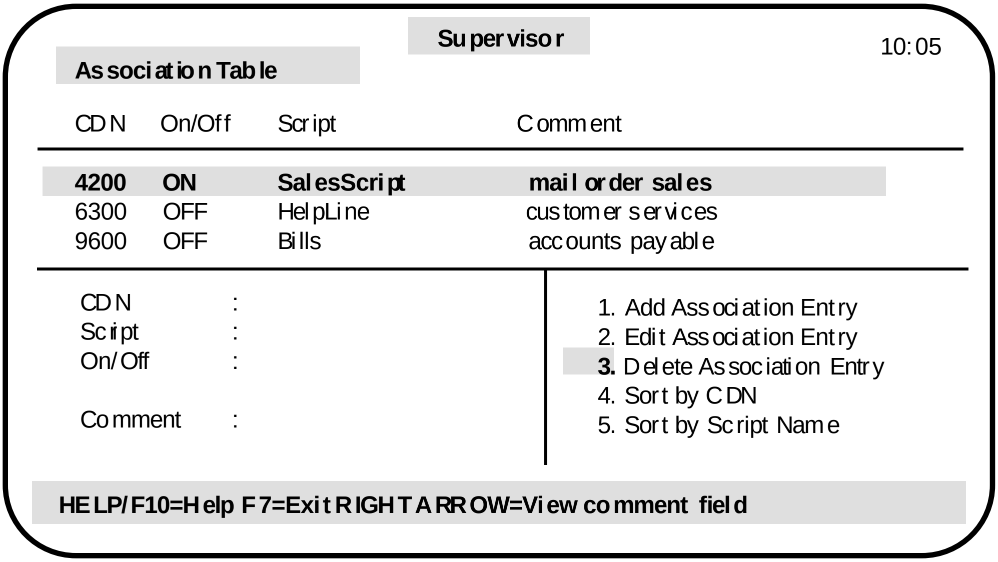
CAUTION

Deleting an association means that new calls directed to the non-associated CDN receive default treatment.

Associations can be deleted at any time. Scripts referencing a deleted association remain in effect until all existing calls handled by those scripts are completed. To delete an association, follow these steps:

1 Select Delete Association Entry from the Association Table menu.

Figure 74
Delete Association Entry window



```
                                   Supervisor
 Association Table                                           10:05

 CDN    On/Off    Script               Comment
 ----------------------------------------------------------------
 4200   ON        SalesScript          mail order sales
 6300   OFF       HelpLine             customer services
 9600   OFF       Bills                accounts payable
 ----------------------------------------------------------------
 CDN         :                     |  1. Add Association Entry
 Script      :                     |  2. Edit Association Entry
 On/Off      :                     |  3. Delete Association Entry
                                   |  4. Sort by CDN
 Comment     :                     |  5. Sort by Script Name

 HELP/F10=Help F7=Exit RIGHT ARROW=View comment field
```

This places you in the selection pane (Figure 74).

2 Scroll to the association you wish to delete and press Return. A message appears, informing you that the association was deleted.

3 Press F6 to activate the updated Association Table.

Sorting the Association Table

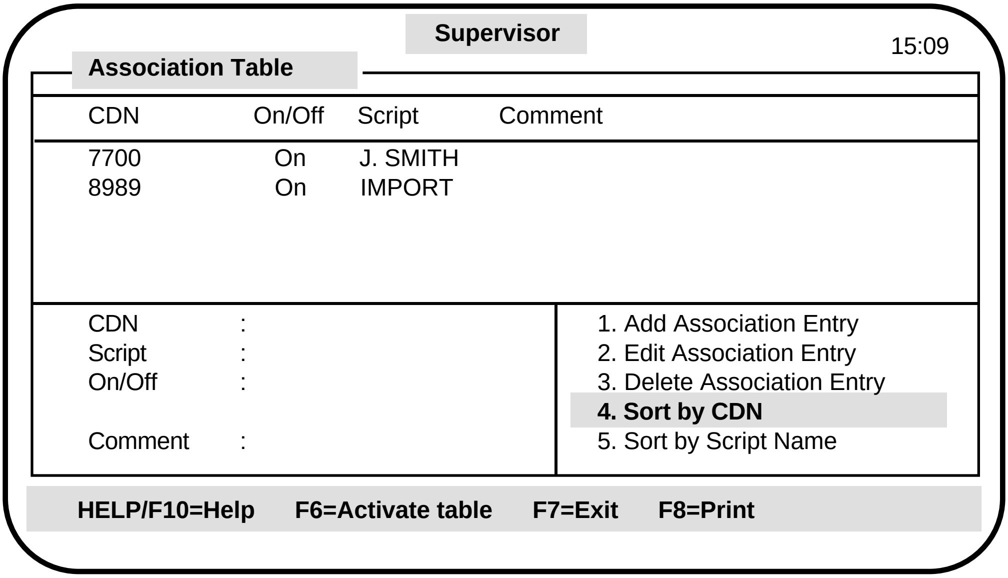
There are two ways to sort association entries: Sort by CDN (shown in Figure 75) or Sort by Script Name (shown in Figure 76).

Sorting by CDN

On the Association Table screen, select the Sort by CDN option.

The selection pane is immediately updated with the entries sorted numerically by CDN.

Figure 75
Sort by CDN screen



Supervisor 15:09

Association Table

CDN	On/Off	Script	Comment
7700	On	J. SMITH	
8989	On	IMPORT	

CDN :
Script :
On/Off :

Comment :

1. Add Association Entry
2. Edit Association Entry
3. Delete Association Entry
4. **Sort by CDN**
5. Sort by Script Name

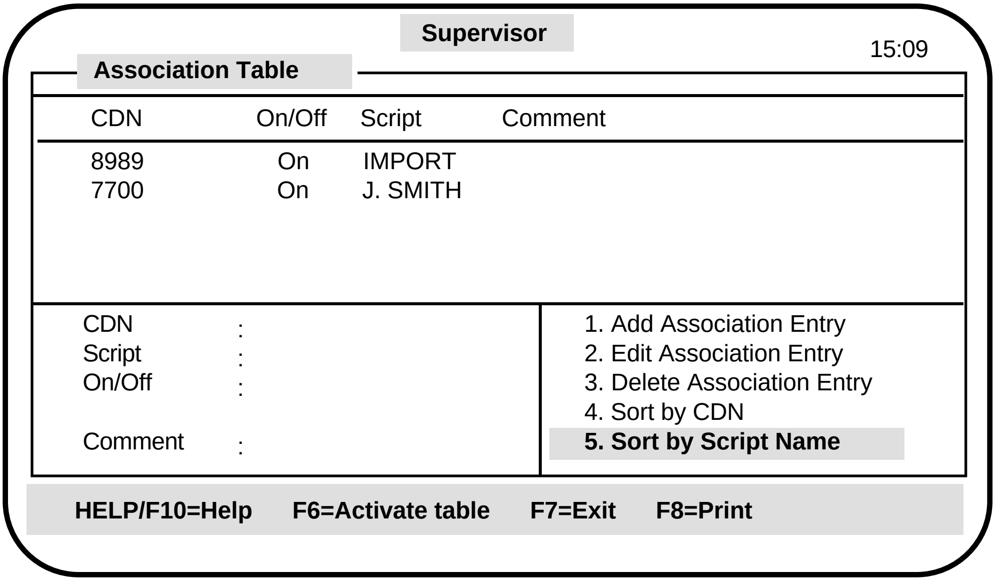
HELP/F10=Help F6=Activate table F7=Exit F8=Print

Sorting by script name

On the Association Table screen, select the Sort by Script Name option.

The selection pane is immediately updated with the entries sorted alphabetically by script name.

Figure 76
Sort by Script Name screen



```
                          Supervisor
                                                          15:09
   Association Table
   CDN          On/Off   Script      Comment
   8989           On     IMPORT
   7700           On     J. SMITH

  CDN          :                   1. Add Association Entry
  Script       :                   2. Edit Association Entry
  On/Off       :                   3. Delete Association Entry
                                   4. Sort by CDN
  Comment      :                   5. Sort by Script Name

  HELP/F10=Help   F6=Activate table   F7=Exit   F8=Print
```

CCR messages

This chapter contains information about the error, warning, information and fatal messages that occur during validation of scripts.

To look up messages by message ID, refer to the *Customer Controlled Routing Diagnostic and Maintenance Guide* (NTP 553-3203-510).

Error messages

Error messages prevent you from validating your script and must be addressed before you can proceed. Here is a list of error messages and their definitions.

Table 18
Error messages

Error message	Description	Example
Busy after treatment	The FORCE BUSY command is given after a treatment (ringback, silence, RAN , music, IVR and queue to).	GIVE RAN 12 FORCE BUSY
Constant too long	The number of digits is too long for a constant. Any entry larger than 32 characters will get this error message.	QUEUE TO 8900 if CLID > 12345678901234567890123123456789012345678901 23

— continued —

Table 18
Error messages, continued

Error message	Description	Example
Constant out of range for type	The value for the type is out of the allowed range.	GIVE RAN 123456789 ***Note:*** RAN is only allowed a value from 0-511.
Illegal character	An illegal character is used. Legal characters are - : , . = < > * + ( ) / Integers 0 1 2 3 4 5 6 7 8 9 Alpha A-Z, a-z	QUEUE TO 890& QUEUE TO 890$ ***Note:*** & and $ are illegal characters.
Illegal constant	Invalid entries are used to express time and date.	QUEUE TO 8900 IF Time Of Day = 12/2 QUEUE TO 8900 IF Day Of Year > 3/2:10
Illegal quit or disconnect	A script quits abruptly ,or the script is empty except for a single occurrence of QUIT, or a QUIT is given before a control. (Commands that give control are FORCE BUSY, FORCE DISCONNECT, ROUTE TO, QUEUE TO.)	GIVE RAN 12 QUIT

— continued —

Table 18
Error messages, continued

Error message	Description	Example
Illegal Infinite Loop	There is an illegal infinite loop with no WAIT, RAN or IVR statement within the loop.	SECTION A WAIT 10 GOTO A SECTION B GIVE RAN 6 GOTO B SECTION C GIVE IVR 8989 GOTO C ***Note:*** The above scripts will produce "Possible Infinite Loop" warnings.
Illegal use of set	A set is incorrectly used to search a set. Only an item may be used.	QUEUE TO 8900 IF 2345, 4567 = 1234..6789 ***Note:*** The left side is a set and therefore incorrect.
— continued —		

Table 18
Error messages, continued

Error message	Description	Example
Label declared twice	The same label is used to define two different SECTIONs. Labels must have unique names to identify SECTIONs.	GOTO Holiday IF Day Of Year = 12/25 GOTO Holiday IF Day Of Year = 1/1 SECTION Holiday GIVE MUSIC 20 .. .. SECTION Holiday GIVE MUSIC 21 .. .. ***Note:*** The SECTION Holiday is declared twice.
Label not defined	A GOTO <label> is given without a SECTION <label> being present in the script.	QUEUE TO 8900 GOTO Holiday ***Note:*** There is no SECTION Holiday.
Mismatched types in expression	Two operands are not of the same type.	Operand1 < Operand2 ***Note:*** Each operand can be a variable, an intrinsic or a constant, but both Operand1 and Operand2 must be the same data type. In this case, both must be type INTEGER.

— continued —

Table 18
Error messages, continued

Error message	Description	Example
Name or word too long	A name or word is greater than 32 characters.	QUEUE TO abcdefghijklmnopqrstuvwxyz1234 56789010
Name is not a variable	A name is used in a script that has not been entered into the Variable Table.	QUEUE TO my_acd ***Note:*** If my_acd is not defined in the Variable Table in lower case, the validation process will not recognize the variable.
Nested comment found	A "/*" is followed by a "/*".	/*/*this is a comment*/
Option not available	The IVR package is not available on the SL-1 or the IVR treatment does not exist.	
Script contains no code	A comment and no other intrinsics or commands exist in the script.	
Syntax error	The language key words are miss-typed.	QUEUE MO 8900 ***Note:*** This statement should read "QUEUE TO 8900."

— continued —

Table 18
Error messages, continued

Error message	Description	Example
Too many errors-can’t continue	There are more than 100 errors in the script.	
Unclosed comment error	A comment is not ended with “*/”.	/*This is an example of an unclosed comment.
Unexpected end of file	The latter part of a FORCE, GIVE, GOTO or SECTION command has not been entered into the script. The line in the script must be the last line in the file.	QUEUE TO 8900 WAIT 20 FORCE ***Note:*** The error will be generated for the FORCE. This command is also the last line in the script.
Variable of wrong type	A variable is entered into the Variable Table and is used incorrectly in a script.	QUEUE TO presidents_dn ***Note:*** A DN type cannot be queued to. A QUEUE TO command expects an ACD type.
? or @ cannot be used in a range	A wildcard or placeholder character has been used in a range.	QUEUE TO 8800 if CLID = 22@..33@ QUEUE TO 800 if CLID = 22?..33?

Warning messages

Warning messages alert you to possible problems in the script writing process. They do not prevent validation and, in some cases, you may want to ignore them.

Warning messages appear if you have the Validate Flag field set to Flag_on in the Profile Maintenance screen. They can be turned off by indicating Flag_off as the value (this is not recommended for new users).

Here is a list of warning messages and their definitions.

Table 19
Warning messages

Error message	Description	Example
Tests on time values not recommended	Tests on time are for a single minute during the day.	QUEUE TO 8900 IF Time Of Day = 8:00 QUEUE TO 8900 IF Time Of Day <> 8:00 ***Note:*** The = operator selects the time range from 8:00 up to, but not including, 8:01. The <> operator selects any time other than from 8:00 up to, but not including, 8:01.
Can be reached without control	It is possible for an incoming call to receive no control.	
Can be reached without treatment	It is possible for an incoming call to receive no treatment.	

— continued —

Table 19
Warning messages, continued

Error message	Description	Example
Code cannot be reached	A script command is written after a ROUTE TO, FORCE BUSY or FORCE DISCONNECT.	QUEUE TO 8900 FORCE BUSY QUEUE TO 8901 ***Note:*** The QUEUE TO 8901 will not be executed because the incoming call has been given a FORCE BUSY.
Label not used	A SECTION is used in the script and there is no corresponding GOTO statement calling the SECTION.	QUEUE TO 8900 SECTION Christmas GIVE RAN 23 ***Note:*** The warning is given for “Christmas”.
Possible infinite loop	There is a possible infinite loop using the GOTO and SECTION commands.	QUEUE TO 8900 SECTION Holiday GIVE MUSIC 23 WAIT 10 GIVE RAN 3 GOTO Holiday ***Note:*** The warning is given for line 6 “GOTO Holiday”. There is no condition set for ending the loop.
Variable name used as label	A variable name is used as a SECTION or GOTO in the script.	SECTION Christmas_Week ***Note:*** “Christmas_Week” is defined in the Variable Table.

Information messages

Information messages help you improve your scripts.

Table 20
Information messages

Message	Description	Example
Min/Max steps before control/ treatment	Provides the minimum statement and maximum statement executed in a script before a call can receive control and treatment, respectively. If a control or treatment statement is conditional, the control/treatment is given only if the condition is true. The minimum is the first statement that could provide control/treatment. If the statement is conditional and the condition is true, the minimum number of statements provides control/treatment. If the statement is conditional and the condition is false, the next possible control/treatment statement is executed, until the script executes either a condition that is true or the maximum. The maximum is the first non-conditional statement to provide control/treatment. Control commands are: QUEUE TO, ROUTE TO, FORCE BUSY and FORCE DISCONNECT. Treatment commands are: GIVE RINGBACK, GIVE SILENCE, GIVE RAN, GIVE MUSIC, GIVE IVR, ROUTE TO and QUEUE TO. Note these statement-numbering rules: • Comments and blank lines are not counted for INFO messages. • The number of statements—not the number of lines—is counted. • Statements are counted in the order that they would be executed.	/*This is a test script */ QUEUE TO 4567 IF Day Of Week = Friday GIVE MUSIC 21 QUEUE TO 8900 This script generates this message: Min/Max steps before control/treatment 1/3 1/2. ***Note:*** 1/3 indicates that the control minimum is statement 1 (the first possible control statement to be executed). In this example, statement 1 queues the call to 4567 only if it is Friday. 1/3 also indicates that the control maximum is statement 3. If it is not Friday, statement 3 queues the call to 8900. 1/2 indicates that the treatment minimum is statement 1 (the first possible treatment statement to execute). In this example, statement 1 queues the call to 4567 and provides default treatment only if it is Friday. 1/2 also indicates that the treatment maximum is statement 2. If it is not Friday, statement 1 does not queue the call or provide default treatment; statement 2 provides music.

— continued —

Table 20
Information messages, continued

Message	Description	Example
IVR got cancelled	A call was receiving IVR but was interrupted. This occurs only with the GIVE INTERRUPTIBLE IVR treatment.	

Fatal messages

Fatal messages tell you why CCR can no longer continue processing a script, when the error is other than a script command.

> ***Note:*** Contact your system administrator if you get one of these messages.

Table 21
Fatal messages

Error message	Description
Out of memory	System memory has been used up.
Error opening script source file	An error is detected upon opening a script for reading.
Read error	An error is detected upon reading a script file from disk.
Error opening output file	An error is detected upon opening a script file for writing.
Error writing output file	An error is detected upon writing a script file to disk.
CDN acquire fatal failure	A fatal error occurred when acquiring the CDN. CCR restarts itself and tries to correct the problem.
Error ready options data	An error is detected while attempting to read the CCR options file.

Message filtering

Because of the large number of calls that it handles, CCR filters call-processing background errors to prevent the log file from filling up and the system from degrading while handling these errors.

This means that CCR tracks unique errors logged (and displayed to all logged-in users) for each command type during a single hour. Errors that exceed the limits set for each command type are not logged and displayed. Every hour (on the hour) the CCR message filtering process is refreshed—that is, the list of message types is cleared and CCR starts it over again.

In general, for every script command type, one error may be logged per hour. In addition to the script command types, statistics messages sent in the background also need to be logged and filtered. This means the following:

- For commands with parameters, CCR logs up to 50 unique errors per hour. Parameterized commands include:
  - QUEUE TO
  - GIVE RAN
  - GIVE MUSIC
  - GIVE IVR
  - REMOVE FROM
- For commands without parameters, CCR logs up to 10 unique errors per hour. Non-parameterized commands include:
  - GIVE RINGBACK
  - GIVE SILENCE
  - FORCE BUSY
  - FORCE DISCONNECT
  - ROUTE TO

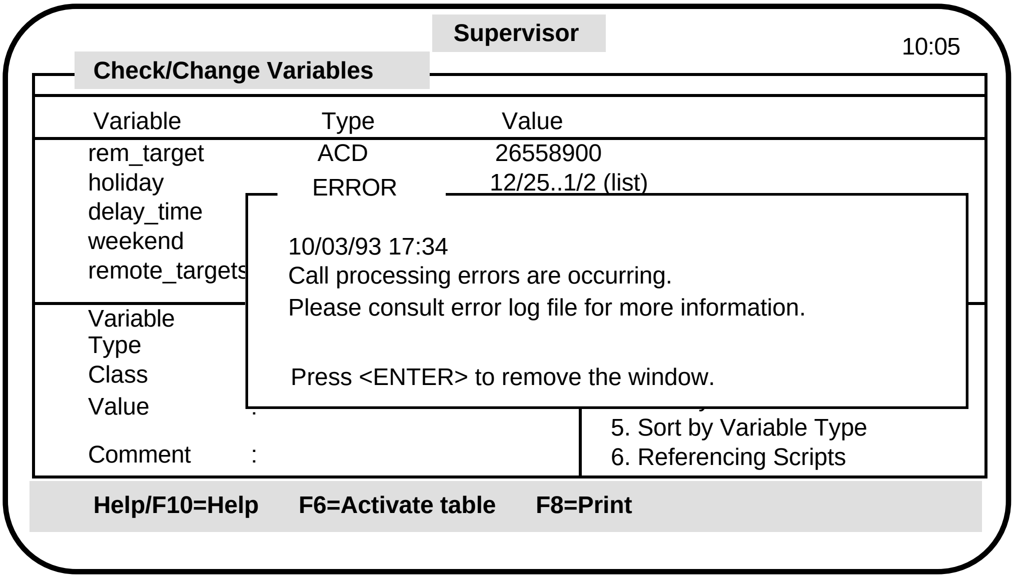
For every error logged, CCR logs the time, the command type, the applicable call parameters (for example, CDN, ACD DN, route number), and the reason for the failure.

Background error notification

To alert you of call-processing background errors, CCR interrupts your operation to display an Error Notification window (Figure 77). However, to prevent a flood of error notifications from engulfing you, CCR message filtering ensures that you will see only one of these call processing windows per hour. Note that all logged-in users see the same Error Notification window at the same time. Note also that if no users are logged in, CCR sends an unsolicited message to the system console.

To consult the error log, log in as maint on the console (if you are authorized to do so), type **view** and press Return. Then select the CCR error log. (Refer to the description of the view command in the *Customer Controlled Routing Diagnostic and Maintenance Guide* (NTP 553-3203-510.)

Figure 77
Error Notification window



```
Variable          Type          Value
rem_target        ACD           26558900
holiday                         12/25..1/2 (list)
delay_time
weekend
remote_targets

Variable
Type
Class
Value         :
Comment       :
```

ERROR

10/03/93 17:34
Call processing errors are occurring.
Please consult error log file for more information.

Press <ENTER> to remove the window.

Key words

The following list of key words are reserved words within CCR’s scripting language and should not be used as user-defined names or labels.

AGE
AGENTS
AND
BUSY
CALL
CALLS
CLID
DAY
DISCONNECT
DNIS
FORCE
FRIDAY
FROM
GIVE
GOTO
IDLE
IF
INTERRUPTIBLE
IVR
LOC
LOGGED
MONDAY
MUSIC
NIGHT
NOT
NPA
NPANXX
NXX
OF
OLDEST
OR
PRIORITY
QUEUE
QUEUED
QUIT
RAN
REMOVE
RINGBACK
ROUTE
SATURDAY
SECTION
SERVICE
SILENCE
SUNDAY
THURSDAY
TUESDAY
TIME
TO
TOTAL
TREATMENT
TUESDAY
WAIT
WEDNESDAY
WEEK
WILDCLID
WITH
YEAR

List of terms

ACD

See Automatic Call Distribution.

ACD DN

The DN that identifies an ACD group. Calls made to this DN will be distributed to ACD agents belonging to that group, according to ACD algorithms.

Age Of Call

This intrinsic allows you to route or treat a call based on the age (in seconds) of the call.

ANI

See Automatic Number Identification (Service) (ANI).

Application Module (AM)

Application Module. Nortel's Motorola-based Application Processor that is packaged within an Application Equipment Module (AEM) in a Meridian 1 system.

Application Module link (AML)

A Nortel internal and proprietary link that connects the Meridian 1 (via ESDI or MSDL) to the Application Module.

association

This is a mapping between a script and CDNs. The Associations Table tells the system which script controls calls entering all CDNs associated with it.

Automatic Call Distribution (ACD)

This is a feature offered by the Meridian 1 that queues and distributes incoming calls to waiting agents. Calls are queued until an agent is available. If multiple agents are available, calls are distributed on an equitable basis. ACD has many additional features, including recorded announcement (RAN), music while queued, night treatment, overflows, statistics/reports, and networking with ISDN.

Automatic Number Identification (Service) (ANI)

The provision of calling party information (typically telephone number or billing/account number) to the called party.

Base Operating System (BOS)

Motorola’s SYSTEM V/68 Base Operating System (UNIX). UNIX System V Release 3 Version 7.1 (V/68 R3V7.1) is used with the CCR Module.

Call Force

This ACD feature controls how calls are presented to agents. An ACD queue is configured with Call Force in overlay 23. If FORC=NO, the call rings until the agent manually answers. During the presentation cycle, the caller hears ringback. If FORC=YES, the switch automatically answers the call on the agent's behalf, bypassing the presentation cycle.

Calling Line Identification (CLID)

Calling Line Identification (CLID) identifies the number of the calling party. If trunking facilities are capable of delivering CLID or ANI digits, these digits are passed to the PBX when the CO routes a call.

case-sensitive

Case-sensitive means that there is a difference between an uppercase and a lowercase of the same letter. You must type uppercase letters in uppercase on your keyboard, and lowercase letters in lowercase. It is important to key in your data in the exact combination of uppercase or lowercase characters. Inputting in the wrong case could make your entry invalid for some fields (for example, password).

CCR

See Customer Controlled Routing.

CDN

See Control DN.

class

Refers to the way items are categorized by the CCR application. Class is an item or set.

CLID

See Calling Line Identification.

comments

Text in a script, ignored by CCR, that explains the intended script processing.

Control DN (CDN)

A Control DN is a special DN, configured in your Meridian 1 system, to which no agents are assigned. To control calls in the CDN you need to create a script; otherwise, the calls are put into the default mode. The script is associated with one or more CDNs, and all calls entering those CDNs are handled by the same script.

Customer Controlled Routing (CCR)

CCR is a flexible call-routing application that allows you to define, for each ACD DN, how calls are handled and routed through a Meridian 1 system.

By using scripts, you can define individualized treatment for your calls. When a call is queued, the treatment given to the waiting call is determined by instructions contained in the script. The set of user-friendly call processing commands can be combined to create various call-routing schemes and treatments to meet your specific business needs.

data type

See type.

Day Of Week

This intrinsic allows you to route or treat a call based on the day of the week.

directory number (DN)

The number that identifies a telephone set on a PBX or in the public network. It is the number that a caller dials in order to establish a connection to the addressed party. The DN could be a local PBX extension (local DN), a public network telephone number, or an ACD DN that is the pilot or group number for an ACD queue.

Directory Number Identification Service (DNIS)

A service provided on a trunk. DNIS identifies to the called system the last one to 31 digits of the number actually dialed by the calling party. The DNIS digits are sent as in-bank DTMF tones on non-ISDN trunks, or using dial pulses on dial pulse (DIP) trunks. On ISDN PRA trunks, DNIS is carried in the called party IE field of the Q.931 Setup message. As an intrinsic, DNIS allows you to route or treat a call based on the DNIS (Dialed Number Identification Service) number for the call being handled.

DN

DN can refer to the data type used for variables that represent internal DNs (other than ACD DNs) and external numbers (any number that can be dialed on the Meridian 1 to make an external call). Also see directory number (DN).

Flexible Digit Strings (FDS)

This feature, provided as part of the CCR Release 3 software, allows expressions of CLID comparisons to include wildcards or placeholders, to facilitate parsing of variable length public dialing plans, as commonly found outside the North American dialing plan.

FORCE BUSY

This command gives a busy signal to the caller.

FORCE DISCONNECT

This command lets you terminate a call.

GIVE IVR

This command lets you give IVR (Interactive Voice Response) treatment to a call.

GIVE MUSIC

This command lets you play music to the caller.

GIVE RAN

This command lets you give a RAN (Recorded Announcement) to the call.

GIVE RINGBACK

This command gives the call ringback tone.

GIVE SILENCE

This command lets you turn off MUSIC, RINGBACK.

global access

Users with global access can create, change, delete, and print any profiles, scripts, variables, and associations. Only one user can have global access at a time. The first global user logged in receives the global access.

GOTO

This command is used to control the flow through a script.

Idle Agents

This intrinsic allows you to route or treat a call based on the number of agents logged in and idle at the specified ACD DN.

IF

This command lets you specify a condition and execute the associated command only if the condition is true.

Interactive Voice Response (IVR)

A system/facility that plays voice menus to callers, and acts upon user input (typically, DTMF digits from a touch tone phone). Sometimes called VRU, Voice Response Unit.

intrinsics

Intrinsics are real time information provided by the system about time, date, calls, and Meridian 1 queues.

IVR

See Interactive Voice Response.

key words

These are reserved words within CCR that are part of the scripting language and cannot be used as variables or SECTION names by the user.

language

The language type is selected from a pop-up menu. Any language listed in that pop-up menu is valid. Screens, help text, and most messages appear in the language specified. Script commands are in English only.

LOC

The LOC is the first three digits for private network calls (for example, the digits 343 in 6-222-343-5343). This intrinsic allows you to route or treat a call based on the LOC of the call being handled.

Logged Agents

This intrinsic allows you to route or treat a call based on the number of agents logged in to the specified ACD DN.

Meridian Link

Meridian 1’s host interface. Meridian Link supports X.25 and LAPB communication protocol for host connectivity.

Night Service

This intrinsic allows you to make decisions on whether or not a command is to be executed based on the Night Service status at the specified ACD DN.

none access

Users with none access cannot view scripts, profiles, associations or variables.

NPA

This intrinsic allows you to route or treat a call based on the NPA (Numbering Plan Area or Area Code) of the call being handled.

NXX

This intrinsic allows you to route or treat a call based on the NXX (Local Exchange Code) of the call being handled.

NPANXX

This intrinsic allows you to route or treat a call based on “NPANXX” of the call being handled.

Oldest Call

This intrinsic allows you to route or treat a call based on the amount of time, in seconds, that the oldest call at the specified ACD DN has waited.

operator

An operator is the symbol used to express a mathematical expression. An example of an operator is the + symbol for addition.

password

This is associated with the User ID. The password is optional and must contain 4–8 characters. The password is case-sensitive.

PBX

See private branch exchange.

placeholder

A placeholder character (?) represents one digit of any value and is used in CLID digit-string comparisons. For example, 555? represents any 4-digit CLID that starts with 555 (such as 5550, 5551, 5552, and so on).

printer

This is selected from a pop-up menu. Any printer listed in the pop-up screen can be selected as the default printer.

private branch exchange (PBX)

A CPE telephone switch, typically used by a business to service its internal telephone needs. A PBX typically offers many more advanced features than are generally available on the public network. A PBX interfaces with the public network central office (CO) via circuits known as trunks. PBX is also used as name for a family of telephone sets: 500 (rotary) and 2500 (touch tone) sets.

profile

User profiles define the level of access end users have in CCR. Global access allows the user to create and change profiles. View access displays operating privileges for the profile only. A user with No access cannot access any profile screen.

Profile Maintenance

This option lets you define operating privileges for each user.

QUEUE TO

This command lets you queue calls to a specified ACD DN.

QUIT

This command ends execution of commands.

range

This refers to a limited series of consecutive items. Ranges may be included in sets to define conditions for routing and treatment of calls. *See also* standard range and wrapped range.

REMOVE FROM

This command lets you remove calls from an ACD DN queue.

ROUTE TO

This command lets you route a call to any dialed number either on or off the switch.

script

This is a collection of statements defining call routing and treatment.

SECTION

This command labels the target of a GOTO command.

set

A list of values, a range of values, or any combination of items, lists, or ranges.

standard range

This refers to a range that is generally used in a given situation. (Compare with "Wrapped range.") Examples of standard ranges are:

total_week = Monday..Sunday

work_week = Monday..Friday

total_day = 00:00..23:59

work_day = 08:30..16:30

statement

A line of script that causes the call to be routed or treated. A statement must contain at least one command. It may also contain other script language elements, such as variables or intrinsics, for example, QUEUE TO acd_main, where QUEUE TO is the command and acd_main is the variable.

Time Of Day

This intrinsic allows you route or treat a call based on the time of day.

Total Queued Calls

This intrinsic allows you to route or treat a call based on the number of calls queued at the specified ACD DN.

type

Refers to the kind of data the variable name represents. INTEGER is an example of a data type.

User ID

This is the ID number for a person using CCR. The User ID must be a four-digit number between 0002 and 9999.

User Name

This is the name that appears at the top of the profile screen, for example, “Supervisor.” It is also associated with the User ID. The User Name can be no longer than 32 characters, including blanks.

validate flag

This allows access to warning messages. A pop-up menu offers Flag_on or Flag_off. If you want to see script warning messages, select Flag_on. If you do not want to see script warning messages, select Flag_off (not advisable for new users). Script errors are always displayed.

variable

This is a user-defined name representing a value or set of values. In the Variable Table, you define variables, such as after_hours, and apply real values to them, like 17:00 to 06:00.

view access

Users with view access can customize only their own profiles. They can only display and print scripts, associations, and variables.

WAIT

This command lets you delay execution of a command.

wildcard

A wildcard character (@) represents 1–32 digits of any value and is used in CLID digit-string comparisons. For example, 555@ represents any CLID that starts with 555.

wrapped range

This refers to a range definition that will exceed standard range limits. Compare with "standard range". Examples of wrapped ranges are:

- total_week = Sunday..Saturday
- weekend_shift = Friday..Sunday
- fiscal_year = 08/01..07/31
- night_shift = 21:30..05:00

Index

Meridian 1

Customer Controlled Routing

Customer Controlled Routing Release 3C

User Guide

Northern Telecom reserves the right to make changes in equipment, design or components as progress in engineering or manufacturing may warrant.

P0884611
Release: Standard 1.0
September 1998
Printed in U.S.A.

NORTEL
NORTHERN TELECOM