P0911816

Nortel Networks Symposium Call Center Server

DMS/MSL-100

Scripting Guide

Product release 3.0 Standard 1.0 April 2000

NORTEL NETWORKS™

How the world shares ideas.

P0911816

Nortel Networks Symposium Call Center Server

for the DMS/MSL-100
Scripting Guide

Publication number:	P0911816
Product release:	3.0
Document release:	Standard 1.0
Date:	April 2000

Printed in the United States of America

Information is subject to change without notice. Nortel Networks reserves the right to make changes in design or components as progress in engineering and manufacturing may warrant.

The process of transmitting data and call messaging between the DMS/MSL-100 and the Nortel Networks Symposium Call Center Server is proprietary to Nortel Networks. Any other use of the data and the transmission process is a violation of the user license unless specifically authorized in writing by Nortel Networks prior to such use. Violations of the license by alternative usage of any portion of this process or the related hardware constitutes grounds for an immediate termination of the license and Nortel Networks reserves the right to seek all allowable remedies for such breach.

Publication history

April 2000

This document is the first standard release of the *Nortel Networks Symposium Call Center Server Scripting Guide* for the DMS/MSL-100.

Contents

Section A: Creating scripts **89**

Section B: Importing and exporting scripts **105**

Section C: Administering scripts **111**

6 Basic script commands 125

Section A: Basic general commands **127**

Section B: Basic call processing commands **141**

Chapter 1

Getting started

In this chapter

Overview

Introduction

The *Nortel Networks Symposium Call Center Server Scripting Guide* for the DMS/MSL-100 provides an overview of the functions of call center scripts and explains the scripting process. The guide explains how to

- plan the scripts used in your call center
- create, modify, and delete script variables
- create, validate, activate, and delete call center scripts
- view applications, and change application thresholds
- use sample scripts for your call center

Skills you need

Introduction

This guide is intended for individuals responsible for designing, writing, and maintaining the scripts used in Symposium Call Center Server.

This section describes the skills and knowledge you need to use this guide effectively.

Nortel Networks product knowledge

Knowledge of, or experience with, the following Nortel Networks products can be of assistance when creating scripts for Symposium Call Center Server:

- Symposium Call Center Server
- the DMS-100 family of switches or the MSL-100 switch

PC experience or knowledge

Knowledge of, or experience with, the following PC products can be of assistance when administering Symposium Call Center Server:

- Microsoft Windows 95, Windows 98, or Windows NT 4.0 Workstation

Other experience or knowledge

Other types of experience or knowledge that can be of use include

- programming
- flowcharting
- analytical skills
- knowledge of call center operations and call routing requirements

Chapter 2

Understanding and planning your scripts

In this chapter

Overview

Introduction

A script is an application containing instructions that determine the sequence of steps that a call follows once it arrives at the Symposium Call Center Server. These steps include call treatments (such as music or ringback), or call routing (such as skill-based routing).

Scripts perform two major functions: they define the path a call follows and they provide treatments to a call as it moves through Symposium Call Center Server. Scripts also enable the Symposium Call Center Server to track and record information about each step in a call’s progress. You can use this information to analyze how your call center is functioning and make decisions on how best to improve service.

Using scripts in your call center

Introduction

To use scripts effectively, you must fully understand the objectives of the call center. Generally, a call center has three major objectives:

- Maximize call center efficiency.
- Maximize caller satisfaction.
- Analyze how your call center is functioning, and make decisions on how best to improve service.

Maximize call center efficiency

To maximize the efficiency of your call center, you must accomplish the following goals:

- Increase productivity.
- Improve service.
- Decrease costs.
- Handle unusual situations.

In an efficient call center, agents process calls that they are qualified to handle. You must design a script so that incoming calls are presented to the agents best prepared to deal with the requirements of the call. This is the basis for skill-based routing: determine a caller’s requirements and route the call to an agent who has the knowledge to deal with it effectively.

Callers should wait for as short a time as possible before speaking with an agent. This accomplishes two things: the caller is less likely to hang up while waiting in queue, and agents spend as little time as possible waiting to answer calls. When both of these conditions are met, costs decrease and profits increase.

Occasionally, an incoming call does not follow the path specified in the script, for example, the call is returned to the queue or is disconnected. Design scripts to prevent such situations or to deal with them in the event they do occur. The script designer must consider unexpected conditions and use scripting tools to resolve them.

Maximize caller satisfaction

Callers should speak to a qualified agent immediately. However, due to large call volumes and a limited number of agents, this is not always the case. You can, however, try to reduce the amount of time each caller waits in queue.

Caller satisfaction is extremely important. Callers waiting in queue do not want to hear silence until their call is answered. They want to know what is happening to their call. If callers begin to doubt that their call is being handled properly, they might hang up.

There are several ways to ensure maximum caller satisfaction. You can

- prioritize calls based on your most important callers
- give callers options while waiting in queue
- inform callers how long they can expect to wait in queue
- inform callers of their position in queue
- let callers speak with an agent of their choice
- let callers speak with an agent in the language of their choice

Well-designed scripts enable you to accomplish these tasks.

Track and report on call information

Scripts allow you to track call-related information and store it in a database for later analysis. If scripts are well-designed, tracking call data uses minimal system resources. Take time to plan and design your scripts to track the information you need. You can use this information later in reports that enable you to analyze how your call center is functioning, and make decisions on how best to improve service. For example, you might want to know the average amount of time agents spend answering calls or the number of abandoned calls.

How call routing and call treatment work

Introduction

Specific scripts are executed when certain types of calls enter the Symposium Call Center Server. These scripts should deal with specific call requirements and route the calls to an agent prepared to deal effectively with these requirements.

As a script designer, you must write scripts to ensure that calls are routed to the qualified agents as quickly as possible. A call is not always answered immediately by an agent. You can, however, provide treatments to the calls while they wait in queue. These treatments can include informing callers of the estimated amount of time before their call is answered, or you can choose to have callers hear music while they wait in queue.

Call routing and call treatment methods

You can route calls by queuing them to

- specific or multiple skillsets
- specific agents
- other call center destinations

Examples of call treatment provided to callers include

- music
- ringback tones
- silence
- recorded announcements
- expected wait time in the queue

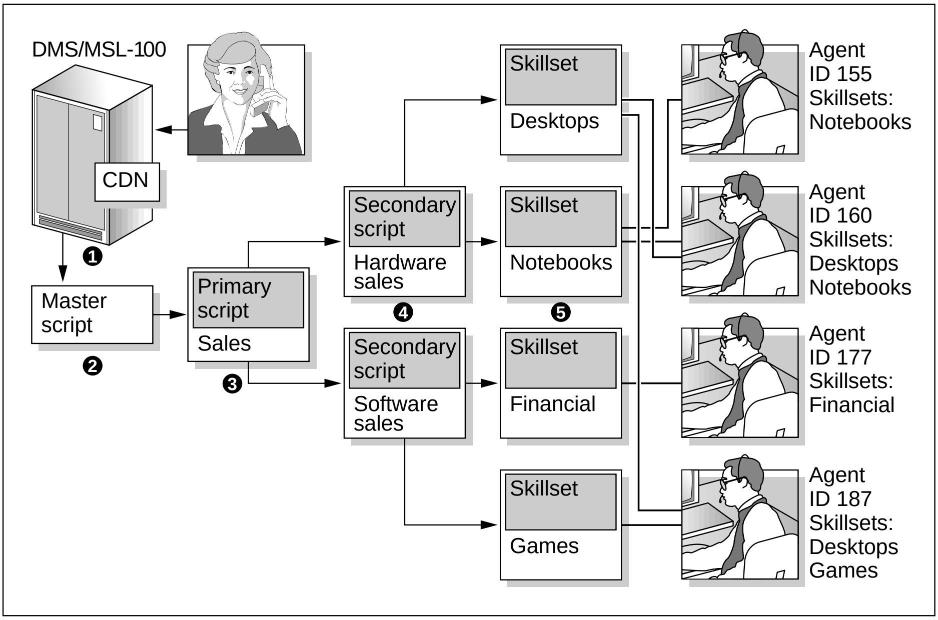
The process of call routing and providing call treatment

When a call enters the Symposium Call Center Server, the call starts the execution of the Master script. Typically, the Master script routes calls and provides treatments based on information such as Dialed Number Identification Service (DNIS), Calling Line Identification (CLID), or the trunk route information. Then the Master script directs the call to primary scripts based on this information. For example, if a CLID number is determined to be on the list of VIP callers, the call can be queued to an agent or skillset reserved for callers requiring preferential treatment.

Additionally, such items as time of day, day of week, day of year, or call center activity can determine how the call is handled.

Example of a typical call routing situation

The following illustration shows how a typical call is handled when it enters a call center. The text following the illustration provides an explanation of what happens at each step in the call flow process.



G101503

1. The incoming call arrives at the DMS/MSL-100 switch, where it is directed to a CDN. The DMS/MSL-100 then notifies Symposium Call Center Server of the call.

2. Symposium Call Center Server takes control of the call. The call begins to follow the path specified in the Master script. The Master script determines

the call type based on DNIS, CLID, and other information, and directs the call to a primary script.

In this example the Master script determines that this is a sales call. The call is directed to the primary script, “Sales.”

3. The primary script can now provide treatments to the call, the call can be queued to a skillset, or the call can be directed to secondary script.

   In this example, the primary script performs a test that determines the caller is interested in hardware sales. The primary script directs the call to the secondary script, “Hardware Sales.”

4. The secondary script can provide additional treatments to the call, the call can be queued to the appropriate skillset or, if necessary, it can be directed to another secondary script.

   In this example, the secondary script performs a test that determines the caller is interested in purchasing a notebook computer. The primary script queues the call to the skillset, “Notebooks.”

5. When an agent in the skillset “Notebooks” becomes available, the call is presented to the agent.

   In this example, the call is presented to either Agent 155 or Agent 160, both of whom have the “Notebooks” skillsets. The call is presented to the first agent available to handle the call.

What happens if a call is not queued?

You can take steps to ensure that calls are successfully queued to the appropriate skillset. If a call is not queued to a skillset or a specific agent when it reaches the end of the script, you can ensure that the call is queued to a default skillset. If the default skillset is out of service, you can inform the caller of this through a recorded announcement (default RAN). After the announcement, the call is queued to the default ACD-DN of the CDN.

For more information about queuing calls to a default skillset or configuring default RANs, refer to the *Administrator’s Guide*.

Types of scripts

Introduction

All Symposium Call Center Server scripts belong to one of the three basic types:

- system-defined
  - Master
- user-defined
  - primary
  - secondary
- sample

System-defined scripts

The Master script is system-defined. This means that it comes with Symposium Call Center Server and cannot be deactivated or deleted. However, you can change its contents to suit your call center's needs, and activate the new version.

Master script

The Master script (Master_Script) is the central point of entry for every call that enters the Symposium Call Center Server. The Master script is system-defined—that is, it comes with the Symposium Call Center Server, and it cannot be deactivated, renamed, or deleted. However, you can change its contents to suit your call center's needs, and activate the new version. The Master script performs the following functions:

- It directs incoming calls to primary scripts based on conditions such as the Dialed Number Identification Service (DNIS), Calling Line ID (CLID), time of day, or any other criteria that you choose.
- It acts as the scheduler for scripts. It invokes primary scripts according to real-time call center conditions.

User-defined scripts

Primary and secondary scripts are user-defined. This means that they do not come with Symposium Call Center Server. You create these scripts on the system. Only a Symposium Call Center Server user with the appropriate privileges can change these scripts.

Primary and secondary scripts

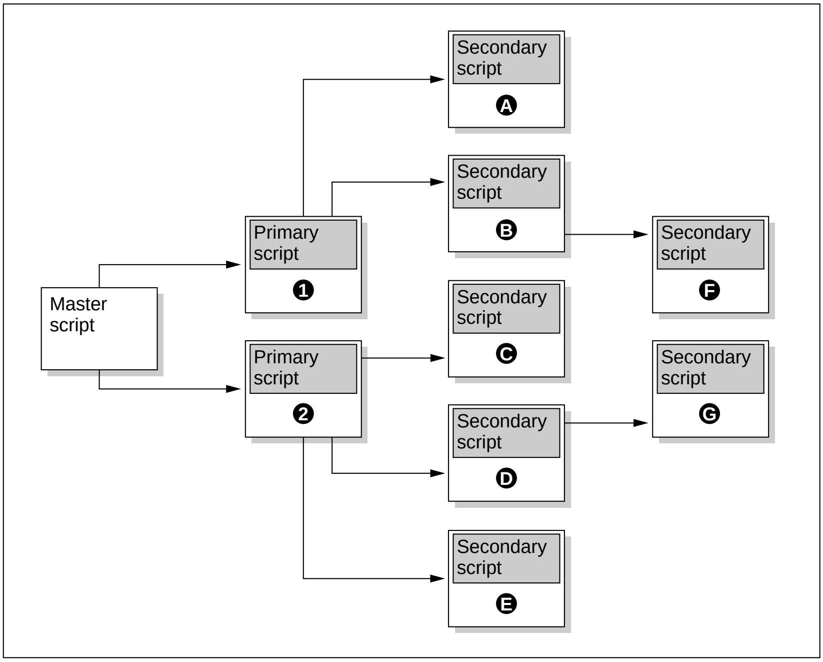
A primary script is executed or referenced in the Master script to perform a specific function. Primary scripts contain sets of instructions that relate to a particular type of call (sales), caller (nuisance), or set of conditions (such as time of day or day of week). A primary script can route calls to appropriately skilled agents, or it can send the control of routing to a secondary script.

A secondary script is any script that is referenced from a primary script or any other secondary script. For example, consider the following situation.

Example of referenced scripts

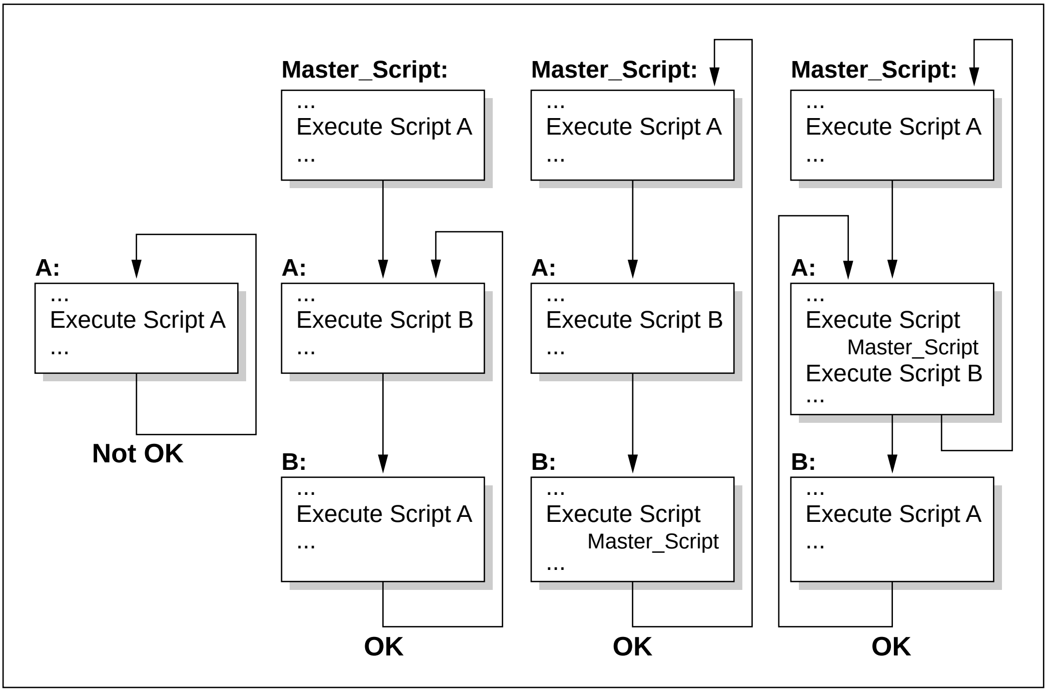
In this example, a caller is interested in purchasing a notebook computer from a computer retailer. You place commands in the Master script that direct the call to a primary “Sales” script. You use conditional tests written into the “Sales” script to determine if the caller’s intention is to purchase a notebook. The “Sales” script then directs the call to a secondary script, “Notebook Sales.” If conditional tests written into the “Notebook Sales” script determine that the caller intended to purchase a notebook from a specific vendor, then the call is directed to another secondary script (for example, “Vendor_Name_Sales”). This script then presents the call to an agent qualified to deal with the specific vendor’s notebook products.

For more information about how scripts reference each other, see the following illustrations. The referencing of scripts (also known as branching), shown in the first illustration, is often referred to as a tree.


Master script
Primary script 1
Primary script 2
Secondary script A
Secondary script B
Secondary script C
Secondary script D
Secondary script E
Secondary script F
Secondary script G

G101504

A script can be referenced from many scripts—that is, there can be several scripts that branch to the same script (for example, the referenced script performs a function required by many other scripts). When this happens, the more complicated structure is called a web, as shown in the next illustration.



G101506

Sample scripts

Sample scripts come with the Symposium Call Center Server client as text files. They are designed to help you create scripts for typical call center situations. You can import or copy the contents of these scripts to help you create your own scripts. You can find the sample script files in the following directories on the client computer:

- C:\Nortel\client\en\script\samples, where C: is the drive on which the client is installed. The “samples” directory contains two subdirectories. For information about the sample scripts contained in these directories, see Chapter 12, “Using sample scripts,” in this guide.

Building blocks of scripts

Introduction

Scripts contain the instructions that tell Symposium Call Center Server how to process incoming calls. These instructions consist of commands, scripting keywords and parameters (some of which are optional), and expressions.

Commands

Commands perform distinct functions, such as routing a call to a specific destination, playing music or recorded announcements to a caller, or disconnecting a caller. Commands are made up of combinations of intrinsics, constants, variables, and expressions.

Expressions

Expressions enable Symposium Call Center Server to create and compare data. To create customized calculations for comparing known facts with conditional situations, you can use mathematical expressions such as addition (+), subtraction (–), division (/), and multiplication (*); logical conjunctions such as AND, OR, and NOT; and comparisons such as less than (<), greater than (>), less than or equal (< =), greater than or equal (> =), and not equal (< >).

Example

In the following section of a script, the total number of calls waiting for the sales skillset is compared to the number of agents logged on to that skillset. In this case, if the total number of calls waiting for the sales skillset is three times the total number of agents logged on to that skillset, then the caller should be given an announcement stating that heavy call volumes can delay servicing of the call.

```
IF (QUEUED CALL COUNT sales_sk) > (3* LOGGED AGENT COUNT
sales_sk) THEN
   GIVE RAN long_delay_ran_gv
END IF
```

Intrinsics

Intrinsics are words or phrases that you use in scripts to represent a value or a set of values about the Symposium Call Center Server system. They contain system-wide information about skillsets, agents, time, and call traffic. Use intrinsics in a script to access system information, which is then used in formulas and decision-making statements.

Example

In the following section of a script, the intrinsic Average Speed Answer checks whether calls are being answered more quickly, on average, by the support skillset than by the service skillset. If they are, then incoming calls are queued to the support skillset.

```
IF (AVERAGE SPEED ANSWER support_sk < AVERAGE
SPEED ANSWER service_sk) THEN

   QUEUE TO SKILLSET support_sk

END IF
```

Variables

Script variables are user-defined words that you can insert in a script in place of a value or a set of values. There are two types of variables:

- Global variables are script variables that you can use in any script on the system.

Example

Create a variable named “business_hours_gv,” and assign the values “8:00 .. 17:00” to that variable. You can then use this variable in several scripts, updating them all at once (for example, if you change to summer hours) by updating the variable.

- Call variables are script variables with a value that can change for each call. These variables follow the call through the system and are passed from one script to another with the call.

Example

If your call center uses an optional third-party IVR system, ask callers to provide some information about themselves—for example, an account number—at the beginning of the call. This information is stored in a call variable, and the value is used only for this particular call.

You define variables on the client. Variables are referenced by more than one script. Therefore, you can change the value of a variable while the system is active. For example, if you change your call center hours of business, you can change the values of the "business_hours_gv" variable and update all of your scripts at once.

Tip: To help you identify types of variables when you are writing and editing your scripts, include information about the variable type in its name. For example, you can name a global variable for a greeting RAN "greeting_ran_gv" or name a call variable for caller-entered data "caller_data_cv."

For more information about variables, see Chapter 4, "Working with script variables."

Skillset

A skillset is an area of expertise that an agent or a group of agents possess that corresponds to a specific call type. Skillsets match callers' specific requirements with agents best prepared to meet their needs. For example, if you expect your call center to receive calls requesting information about servicing notebook computers, create a skillset (such as skillset "notebook_service_sk") and assign agents to this skillset who are knowledgeable about servicing notebook computers.

Planning your scripts

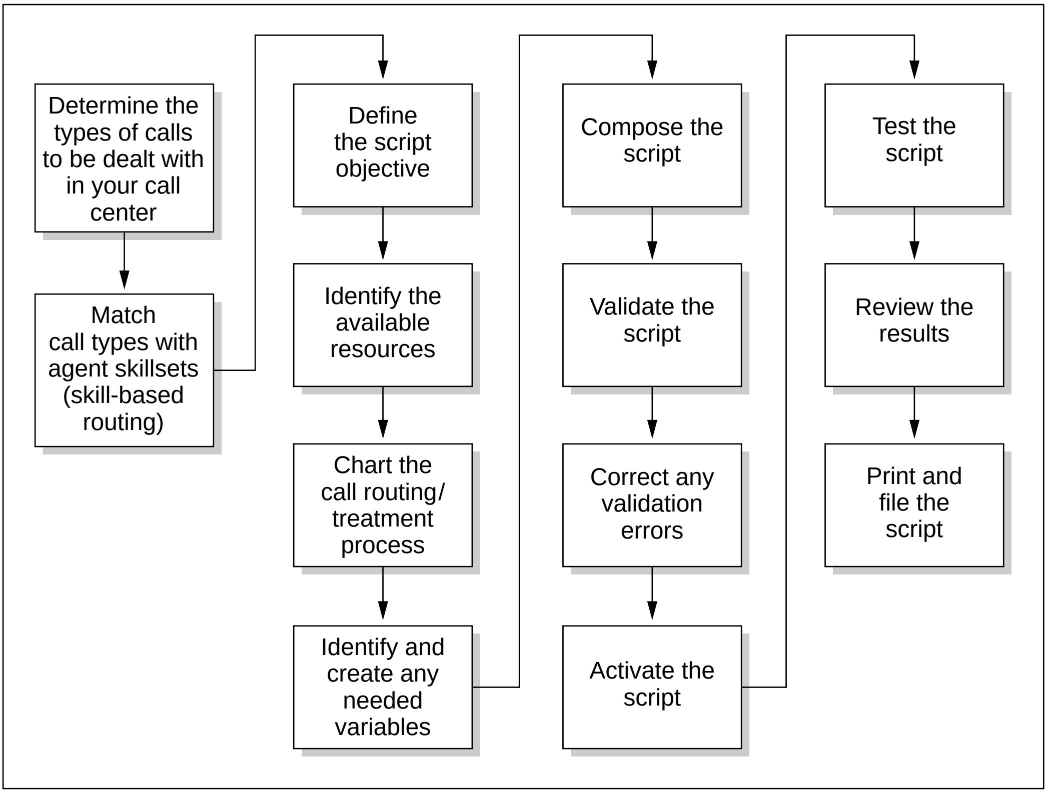
Introduction

An efficient call center is one in which you have successfully matched callers and their specific requirements with agents qualified to handle their calls. If you are aware of the types of callers who place calls to your call center, and the specific information or services they require, then you can begin to write effective scripts.

Efficient servicing of your callers ensures an efficient call center. You can accomplish this with well-written scripts. If you understand these objectives, you can begin to draft the process of how to treat the different types of calls entering your call center.

Scriptwriting process

The following flowchart shows an overview of the scriptwriting process.


Determine the types of calls to be dealt with in your call center
Match call types with agent skillsets (skill-based routing)
Define the script objective
Identify the available resources
Chart the call routing/ treatment process
Identify and create any needed variables
Compose the script
Validate the script
Correct any validation errors
Activate the script
Test the script
Review the results
Print and file the script

G101505

Determine the types of calls entering your call center

Once you have determined the types of calls entering your call center, you can create skillsets that correspond to these call types, and assign agents to the skillsets.

Understand skill-based routing

The concept of matching qualified agents with related call types is the basis for skill-based routing. You must be able to match callers requirements with an agent prepared to answer their questions. At the same time, you must be aware of when to provide treatments to calls and how call information is tracked. When you understand these issues, you can write efficient scripts.

Define the script’s objective

Each script should meet an objective of your call center. You might require the script to perform one or more functions. Determine what purpose the script must accomplish (for example, routing a specific caller to a specific agent). Keep in mind that each script can be referenced by or might need to reference other scripts. Consider the following questions:

- What kinds of scripts do you need for your system?
- How many scripts do you need to meet these requirements?
- How do the scripts interact with one another?

To help plan your scripts, you can create a flowchart or an illustration to outline the intended logic of the script (for example, a script tree or web). A flowchart helps you to visualize the sequence of steps that a call follows once it enters the Symposium Call Center Server. A flowchart also allows you to determine when you give treatments to the call and when you can collect data for reporting purposes.

Identify the available resources

Symposium Call Center Server includes many resources that you can use in your scripts to control what happens to a call once it enters the call center. Knowledge of these resources and how they work together helps you to design a logical path that calls follow. Before you write scripts, you must be familiar with the following resources:

- CDNs
- RAN routes
- music routes
- skillsets
- number of agents in each skillset
- call center working hours and holidays
- call treatments
- CLIDs and DNISs
- whether your call center uses an optional third-party IVR system or screen-pop application

You can get this information from your call center administrator or, for more information about setting up these resources, refer to the following guides:

- *Setup Guide*
- *Symposium and DMS Switch Guide*
- *Symposium and MSL-100 Switch Guide*

Chart the call routing process

Create a flowchart

You can create a flowchart that illustrates the call routing and call treatment process. The flowchart on the following page shows an example of call routing and call treatment. The text below explains the call flow process.

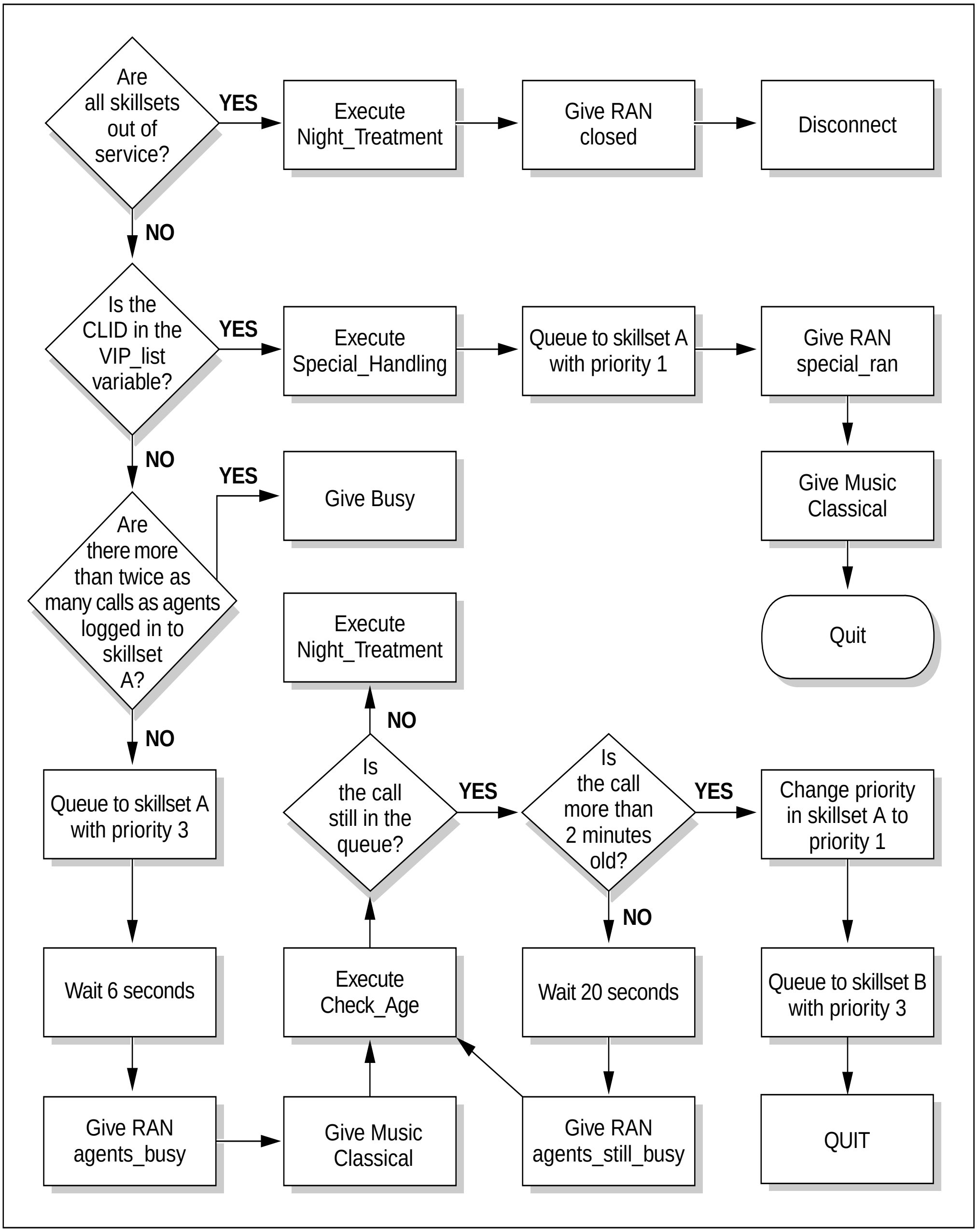
A call arrives at the DMS/MSL-100 switch and has been forwarded to the Symposium Call Center Server. The call has gone through the Master script and has arrived at either the primary or a secondary script (depending on how the Master script has been written).

The script first checks to see if agents are available in the requested skillets to answer the call. If not, the call is disconnected.

The script then checks to see if the caller is on the list of VIP callers. If so, the caller is given preferential treatment. If not, a test is performed to determine if there are more than twice as many calls currently queued as there are agents logged on to the preferred skillset, “skillset A.” If so, the caller receives a busy tone. If there are less than twice as many calls currently queued as there are agents logged on to “skillset A,” the call is queued to “skillset A.”

Once the call is queued to “skillset A,” the script then performs a number of conditional tests and treatments to the call until it is answered.

To see the sample script that this flowchart represents and a more detailed explanation of the call routing process, see page 58.


Are all skillsets out of service?
YES
Execute Night_Treatment
Give RAN closed
Disconnect
NO
Is the CLID in the VIP_list variable?
YES
Execute Special_Handling
Queue to skillset A with priority 1
Give RAN special_ran
NO
YES
Give Busy
Give Music Classical
Are there more than twice as many calls as agents logged in to skillset A?
Execute Night_Treatment
Quit
NO
NO
Queue to skillset A with priority 3
Is the call still in the queue?
YES
Is the call more than 2 minutes old?
YES
Change priority in skillset A to priority 1
NO
Wait 6 seconds
Execute Check_Age
Wait 20 seconds
Queue to skillset B with priority 3
Give RAN agents_busy
Give Music Classical
Give RAN agents_still_busy
QUIT

G101377

Describe in writing

If you have graphically planned the logic of a script and you are satisfied with its intended function, you can choose to write the script on paper before creating it in the Scripts Editor (for more information about the Scripts Editor, see "The Scripts Editor" on page 27).

Identify and create needed variables

A variable is a placeholder you create that stores a value or set of values. For example, you can create a variable named "holidays_gv" to store information on the days when your call center is closed. You use variables to test for conditions that can affect the treatments given to the call or the data (both call information and caller entered) collected from a call as it moves through Symposium Call Center Server. You must define all of your variables before you write your scripts. If you define a variable that is not used in any script, delete the variable.

For more information about variables, see Chapter 4, "Working with script variables."

Compose your scripts

When you compose a script, you follow a four-part process: enter the script in the Scripts Editor, validate the script, activate the script, and test the script. For more information about composing scripts, see "Scripting tools and procedures" on page 26.

Using a common secondary script to reduce script maintenance and system processing power

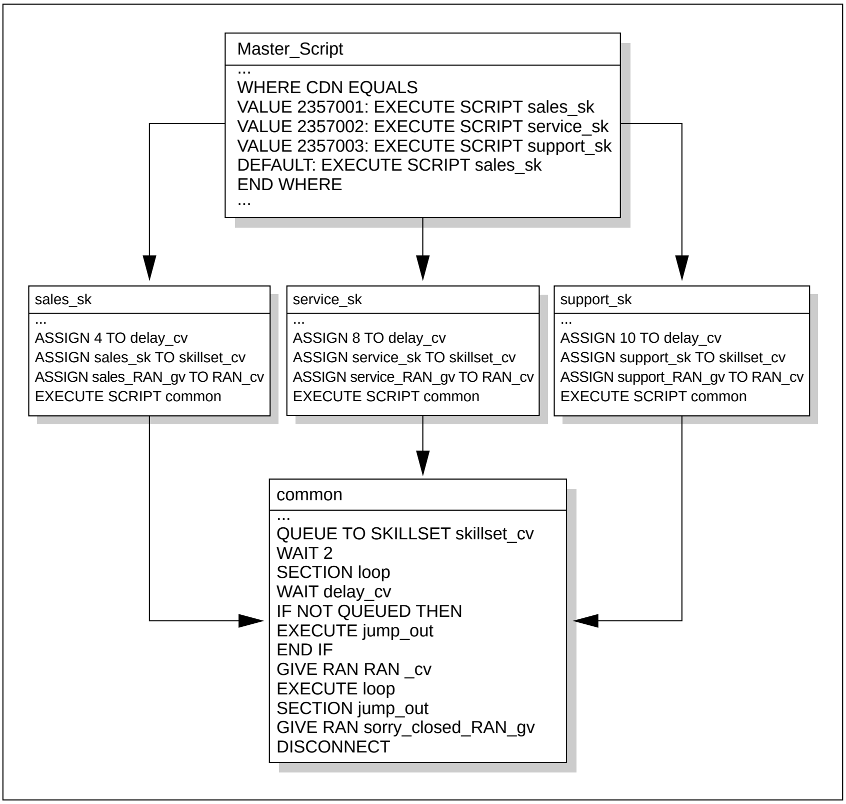
If you create a number of primary scripts that perform a similar function, maintaining these scripts can be time consuming. Each script includes many script elements common to all scripts. If you decide to modify a script element common to all of the scripts, you must edit and then reactivate each script.

As a more efficient alternative, consider combining all of the common elements in one secondary script, and use multiple primary scripts to define any unique call variables.

For example, if your call center receives three types of calls—sales, service, and support—create three primary scripts and one common secondary script to handle these call types. The three primary scripts are used only to define call variables associated with each call type (sales, service, or support). The secondary script contains all routing instructions and treatments common to each call type.

Example

The illustration below shows a Master script referencing three primary scripts that direct calls to a common secondary script. In the example, the three primary scripts (“sales_sk”, “service_sk”, and “support_sk”) are used to define the call variables “delay_cv”, “skillset_cv”, and “RAN_cv”. The value of the CDN tested in the Master script determines which primary script executes. Once the call variables are defined in the primary script, the secondary script “common” is executed using the values assigned in the primary script.



```
Master_Script
...
WHERE CDN EQUALS
VALUE 2357001: EXECUTE SCRIPT sales_sk
VALUE 2357002: EXECUTE SCRIPT service_sk
VALUE 2357003: EXECUTE SCRIPT support_sk
DEFAULT: EXECUTE SCRIPT sales_sk
END WHERE
...
```

```
sales_sk
...
ASSIGN 4 TO delay_cv
ASSIGN sales_sk TO skillset_cv
ASSIGN sales_RAN_gv TO RAN_cv
EXECUTE SCRIPT common
```

```
service_sk
...
ASSIGN 8 TO delay_cv
ASSIGN service_sk TO skillset_cv
ASSIGN service_RAN_gv TO RAN_cv
EXECUTE SCRIPT common
```

```
support_sk
...
ASSIGN 10 TO delay_cv
ASSIGN support_sk TO skillset_cv
ASSIGN support_RAN_gv TO RAN_cv
EXECUTE SCRIPT common
```

```
common
...
QUEUE TO SKILLSET skillset_cv
WAIT 2
SECTION loop
WAIT delay_cv
IF NOT QUEUED THEN
EXECUTE jump_out
END IF
GIVE RAN RAN _cv
EXECUTE loop
SECTION jump_out
GIVE RAN sorry_closed_RAN_gv
DISCONNECT
```

G101507

Scripting tools and procedures

Introduction

This section describes the tools that you use to view, create, and edit scripts. This section also describes script states, validation, and activation.

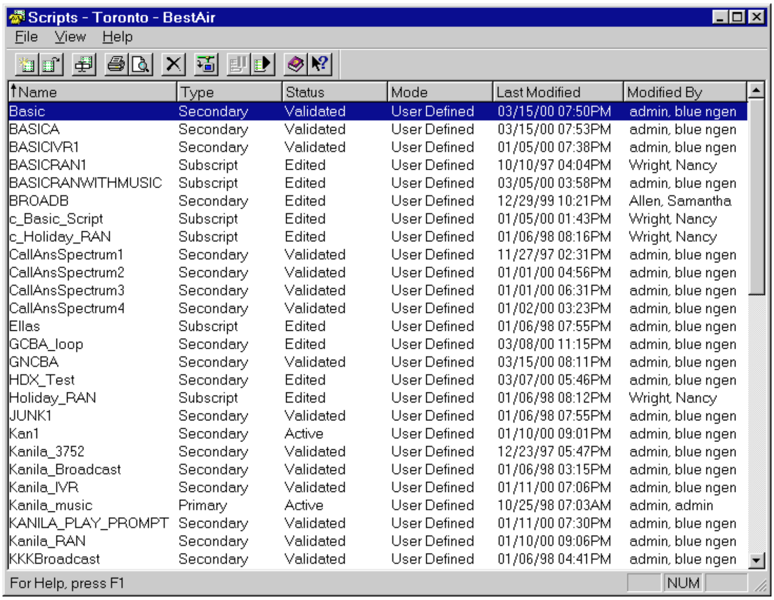
Viewing, creating, and editing scripts

Use the following tools to view, create, and edit scripts:

- the Script Manager
- the Scripts Editor
- the Script Command Reference page

The Script Manager

Use the Script Manager to list the scripts on the system. From this window, you can add or remove a script and access the Scripts Editor to change a script. You can also activate or deactivate a script directly from this window.

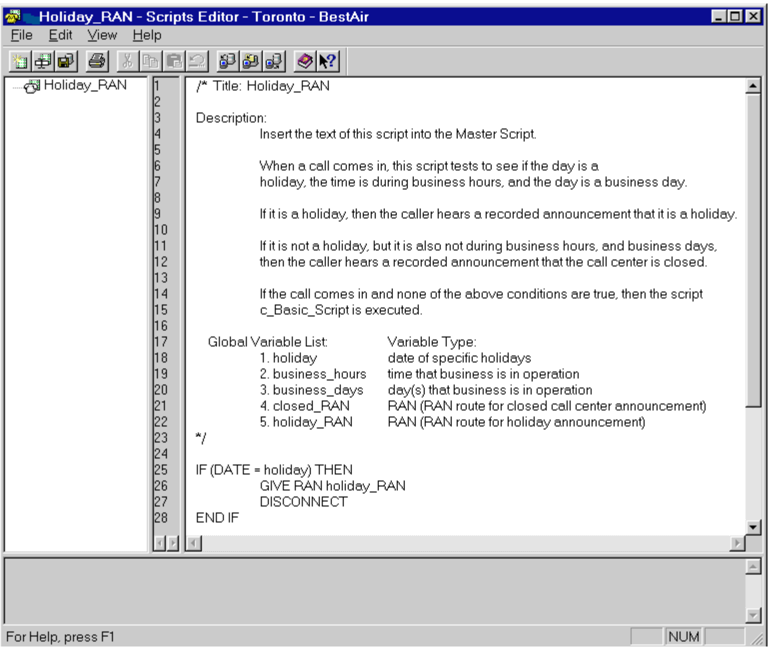


↑Name	Type	Status	Mode	Last Modified	Modified By
Basic	Secondary	Validated	User Defined	03/15/00 07:50PM	admin, blue ngen
BASICA	Secondary	Validated	User Defined	03/15/00 07:53PM	admin, blue ngen
BASICIVR1	Secondary	Validated	User Defined	01/05/00 07:38PM	admin, blue ngen
BASICRAN1	Subscript	Edited	User Defined	10/10/97 04:04PM	Wright, Nancy
BASICRANWITHMUSIC	Subscript	Edited	User Defined	03/05/00 03:58PM	admin, blue ngen
BROADB	Secondary	Edited	User Defined	12/29/99 10:21PM	Allen, Samantha
c_Basic_Script	Subscript	Edited	User Defined	01/05/00 01:43PM	Wright, Nancy
c_Holiday_RAN	Subscript	Edited	User Defined	01/06/98 08:16PM	Wright, Nancy
CallAnsSpectrum1	Secondary	Validated	User Defined	11/27/97 02:31PM	admin, blue ngen
CallAnsSpectrum2	Secondary	Validated	User Defined	01/01/00 04:56PM	admin, blue ngen
CallAnsSpectrum3	Secondary	Validated	User Defined	01/01/00 06:31PM	admin, blue ngen
CallAnsSpectrum4	Secondary	Validated	User Defined	01/02/00 03:23PM	admin, blue ngen
Ellas	Subscript	Edited	User Defined	01/06/98 07:55PM	admin, blue ngen
GCBA_loop	Secondary	Edited	User Defined	03/08/00 11:15PM	admin, blue ngen
GNCBA	Secondary	Validated	User Defined	03/15/00 08:11PM	admin, blue ngen
HDX_Test	Secondary	Edited	User Defined	03/07/00 05:46PM	admin, blue ngen
Holiday_RAN	Subscript	Edited	User Defined	01/06/98 08:12PM	Wright, Nancy
JUNK1	Secondary	Validated	User Defined	01/06/98 07:55PM	admin, blue ngen
Kan1	Secondary	Active	User Defined	01/10/00 09:01PM	admin, blue ngen
Kanila_3752	Secondary	Validated	User Defined	12/23/97 05:47PM	admin, blue ngen
Kanila_Broadcast	Secondary	Validated	User Defined	01/06/98 03:15PM	admin, blue ngen
Kanila_IVR	Secondary	Validated	User Defined	01/11/00 07:06PM	admin, blue ngen
Kanila_music	Primary	Active	User Defined	10/25/98 07:03AM	admin, admin
KANILA_PLAY_PROMPT	Secondary	Validated	User Defined	01/11/00 07:30PM	admin, blue ngen
Kanila_RAN	Secondary	Validated	User Defined	01/10/00 09:06PM	admin, blue ngen
KKKBroadcast	Secondary	Validated	User Defined	01/06/98 04:41PM	admin, blue ngen

For Help, press F1

The Scripts Editor

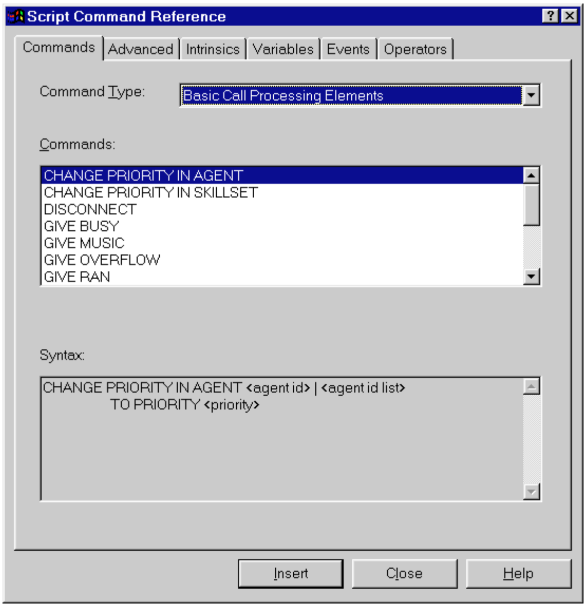
In the Scripts Editor, you can create, edit, validate, and activate the scripts that generate call processing for your call center. The Scripts Editor provides a text-based editor for customizing your scripts. In the Scripts Editor, you can import and export scripts and copy portions of other scripts into the current script.



```
/* Title: Holiday_RAN

Description:
            Insert the text of this script into the Master Script.

            When a call comes in, this script tests to see if the day is a
            holiday, the time is during business hours, and the day is a business day.

            If it is a holiday, then the caller hears a recorded announcement that it is a holiday.

            If it is not a holiday, but it is also not during business hours, and business days,
            then the caller hears a recorded announcement that the call center is closed.

            If the call comes in and none of the above conditions are true, then the script
            c_Basic_Script is executed.

  Global Variable List:         Variable Type:
            1. holiday          date of specific holidays
            2. business_hours   time that business is in operation
            3. business_days    day(s) that business is in operation
            4. closed_RAN       RAN (RAN route for closed call center announcement)
            5. holiday_RAN      RAN (RAN route for holiday announcement)
*/

IF (DATE = holiday) THEN
            GIVE RAN holiday_RAN
            DISCONNECT
END IF
```

The Script Command Reference page

From the Scripts Editor, you can also access the Script Command Reference page, which provides the ability to view and select available script commands, variables, events, intrinsics, and operators for pasting into scripts. Although you can type commands into your scripts manually, if you use the Script Command Reference page, all parameters that you must replace are automatically inserted. This can help to reduce errors in your scripts.



Command Type: Basic Call Processing Elements

Commands:

CHANGE PRIORITY IN AGENT
CHANGE PRIORITY IN SKILLSET
DISCONNECT
GIVE BUSY
GIVE MUSIC
GIVE OVERFLOW
GIVE RAN

Syntax:

```
CHANGE PRIORITY IN AGENT <agent id> | <agent id list>
          TO PRIORITY <priority>
```

For information on the “building blocks” of scripts (commands and expressions), refer to the following chapters:

- Chapter 4, “Working with script variables”
- Chapter 6, “Basic script commands”
- Chapter 7, “Advanced script commands”
- Chapter 9, “Intrinsics”
- Chapter 10, “Script expressions”

The Validation Options dialog box

From the Scripts Editor, you can access the Validation Options dialog box. Validation Options provides an effective tool to help the scriptwriter create or edit scripts. Set validation options so the application can inform you when you are breaking scriptwriting rules. These rules are designed to eliminate run-time errors that can result in improper routing of calls in Symposium Call Center Server.

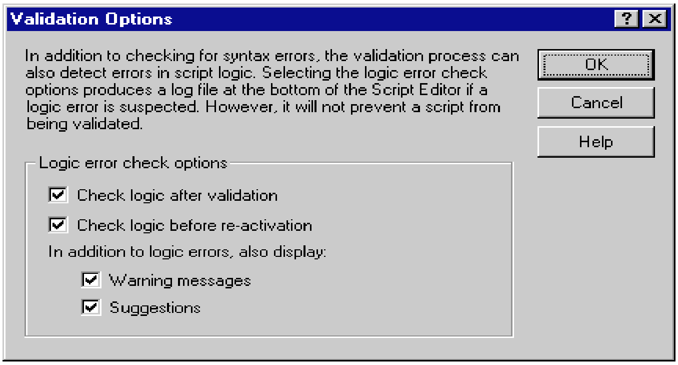
Validation options can be configured so you are informed of broken scriptwriting rules automatically after a script has successfully validated, or before an activated script is edited and then reactivated. You can also configure validation options to display error messages.

Note: If you have configured validation options, breaking a scriptwriting rule results in an error message being displayed. For a list of error messages that can appear, see "Appendix A, "Troubleshooting."

To configure validation options

1 From the Scripts Editor, choose View → Validation Options.

Result: The Validation Options dialog box appears.



Validation Options

In addition to checking for syntax errors, the validation process can also detect errors in script logic. Selecting the logic error check options produces a log file at the bottom of the Script Editor if a logic error is suspected. However, it will not prevent a script from being validated.

Logic error check options

☑ Check logic after validation

☑ Check logic before re-activation

In addition to logic errors, also display:

☑ Warning messages

☑ Suggestions

2 Configure validation options by selecting the appropriate boxes.

3 Click OK.

To disable validation options

1 From the Scripts Editor, choose View → Validation Options.

Result: The Validation Options dialog box appears.

2 Deselect any options chosen in the boxes.

3 Click OK.

Script states

A script can be in three "states": edited, validated, and activated.

Edited

The script has been created or edited, and saved, but has not been validated.

Validated

The script syntax has been checked for errors, and the script is ready to be activated.

Activated

The script is active in the system.

When you finish writing or updating a script in the Scripts Editor, you must validate and then activate it before it can begin to process calls. When you no longer want the script to process calls, you can deactivate it. You can also edit and reactivate a script that is in Activated state. Revalidation is performed once the script is activated.

You do not need to validate or activate a script to save it. You can also change and revalidate it at any time after you create it.

Validation

Before you put a script into service, or “activate” it, you must check it to ensure that the syntax and semantics are correct. This process is called validation. If the script does not contain any errors, validation results in an executable version of the script. If the script contains errors, validation results in a list of those errors, and the corresponding lines of the script where the errors occurred.

Note: Script validation detects only syntax errors. It cannot detect errors in logic.

Activation

An activated script processes calls or is ready to process calls. To activate a script, it must first be validated. (The system validates a script automatically before it is activated, if you have not already done so.)

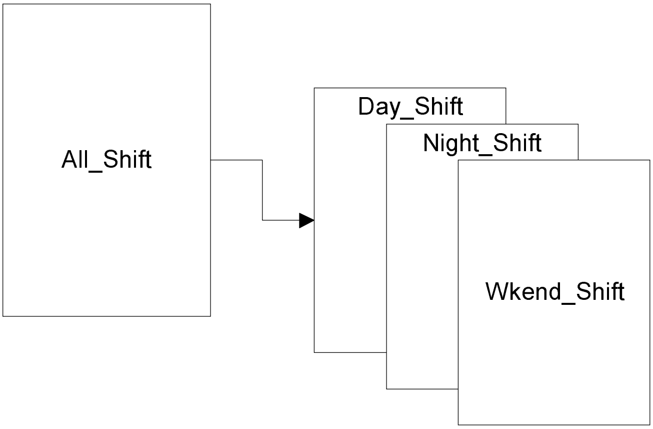
Note: If a script is activated but is not referenced by the Master script (directly, or indirectly through other scripts), then it does not process any calls.

Activating secondary scripts

When you activate a script, all scripts that it references are automatically activated, too (if they are not already activated). Therefore, you must validate all referenced scripts before you activate your script. If you do not, activation of your script fails.

Example

The All_Shift script references the Day_Shift, Night_Shift, and Wkend_Shift scripts. You must validate all secondary scripts before you can activate the All_Shift script.



Each script has been successfully validated by the administrator. When the administrator activates the All_Shift script, the system automatically activates the Day_Shift, Night_Shift, and Wkend_Shift scripts.

Changing an activated script

You can make changes to a script while it is activated. You then have a choice between activating the script with the changes immediately, or saving the script with a new name and not putting the changes into service right away. If you choose to put the new version of the script into service immediately, any calls that were already in progress when the script was changed continue to be handled by the original version of the script. The updated version of the script handles new calls.

Note: If an error occurs while the script is being activated, then the original script is used to process calls. You are not allowed to save your changes with the same name.

Deactivation

A deactivated script does not process new calls. If calls already in the system are using the script when it is deactivated, then they continue to be processed by the script until they are completed.

You cannot deactivate or delete a script while it is being referenced by another activated script.

Although you cannot deactivate the Master script, you can change the contents and activate the new version. Validation is performed once the script is activated.

Chapter 3

Scripting rules and conventions

In this chapter

Overview

Introduction

This chapter contains tips to help you plan how best to write and organize your scripts. This chapter covers the following topics:

- "Script formatting conventions" on page 36 outlines the formatting conventions to use when you write scripts. Follow these conventions so your scripts are easy to understand.
- "Script rules" on page 41 lists the rules that you must follow when you write scripts. If you do not follow these rules, errors can result when you validate your scripts.
- "Scriptwriting tips" on page 44 offers recommendations to help you write efficient scripts.
- "Script example using formatting conventions and rules" on page 58 is a sample script that demonstrates the use of these formatting conventions and tips.
- "Logging on to a Symposium Call Center Server system" on page 61 provides instructions to connect to the Symposium Call Center Server.

Before you begin

Before you begin to write scripts or create variables, Symposium Call Center Server must be installed and configured.

In addition, all system resources such as RAN routes, music routes, voice ports, call treatments, and CDNs must be set up and acquired. For more information, refer to the *Symposium and DMS Switch Guide or the Symposium and MSL-100 Switch Guide.*

Additionally, all agents, skillsets, and thresholds must be created. For more information, refer to the *Administrator's Guide.*

If you plan to use a third-party IVR system to collect caller entered data, you must set that system up before you can use the caller entered data in your scripts. See the documentation that came with your IVR system for more information.

Finally, If you plan to use a third-party screen-pop application, you must set up Meridian Link, TAPI, and CTI. For more information about Meridian Link, refer to the *Software Installation and Upgrade Guide*. For information about setting up TAPI and CTI, refer to the documentation that came with your applications.

Note: The voice ports and position IDs for the IVR system on the switch must be identical to the configuration on the Symposium Call Center Server.

Script formatting conventions

Introduction

This section provides formatting conventions to use when you write your scripts. To ensure that your scripts are easily read and understood by you and others, follow a consistent format.

Commands

Type commands in all uppercase letters.

```
QUEUE TO SKILLSET service_sk
```

Intrinsics

Type intrinsics in all uppercase letters.

```
IF (AGE OF CALL > 30)
```

Logical expressions

Type logical operators (And, Not, Or) in all uppercase letters.

```
IF (TIME OF DAY = business_hours_gv)
AND (DAY OF WEEK = business_days_gv) THEN
   GIVE RAN open_ran_gv
END IF
```

Parentheses

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 that you want. For more information on how parentheses affect your scripts, see “Order of operations” on page 286.

Section names

Type section names with initial capital letters followed by lowercase letters.

```
SECTION Night_Treatment
```

(Remember that the word “section” is a command, and what follows is the section name.)

Skillset names

Type skillset names in all lowercase letters or in mixed case with the first letter capitalized.

```
sales
```

```
or
```

```
Sales
```

Tip: If you follow the skillset name with an underscore and the letters sk (for example, “sales_sk” or “Sales_sk”), this quickly identifies any skillsets displayed in the Script Editor.

Variables

Type variables in all lowercase letters or in mixed case with the first letter in lowercase.

```
GIVE RAN closed
```

or

```
GIVE RAN openHours
```

Tip: To help you identify types of variables in your scripts, include information about the variable type in its name. For example, you can name a global variable for a greeting RAN, “greeting_ran_gv,” or name a call variable for caller-entered data, “caller_data_cv.”

Comparison expressions using If-Then-Else-End If

If and Then should always be on the same line, unless the statement is too long to fit on a single line. Indent commands following the If statement using the tab key. When an If statement is followed by a second If statement (known as a nested If), indent the second If statement. When using multiple If statements in a script, each IF must have a matching End If.

```
IF (CLID = vip_list_gv) THEN

   EXECUTE Vips_Section

ELSE

   IF (CLID = special_list_gv) THEN

       EXECUTE Special_Section

   END IF

END IF
```

Blank lines, indenting, and comments

To improve the readability of your scripts, leave blank lines between sections and before and after comments. Indent commands inside sections to make it easier to identify sections. Indent commands that extend beyond one line.

Comments help others understand your intentions for a section. While comments are not required, they can be extremely helpful in understanding the original purpose of the section for those who review the script at a later date.

Note: Tabs and extra spaces within a line are ignored. Blank lines are also ignored.

ATTENTION

Comments included in your scripts must begin with an opening marker (/*) and end with a closing marker (*/). If you do not include both an opening and a closing marker, the script does not validate.

Example

The following example shows the proper use of comments, indenting, and blank lines between sections.

```
/* This section of the script queues calls to the general
skillset during regular business hours.*/

IF (DAY OF YEAR = holiday_gv)

OR (DAY OF WEEK = weekend_gv) THEN

       EXECUTE Night_Treatment

END IF

QUEUE TO SKILLSET general_sk

QUIT

SECTION Night_Treatment

   GIVE RAN closed_ran_gv

   DISCONNECT
```

The perfect basic script

The following example shows a script with all recommended formatting conventions applied. This example clearly distinguishes elements such as commands, variables, and skillsets. Write all of your scripts in this manner.

Example

```
IF (DATE = holidays_gv)

OR (DAY OF WEEK = weekend_gv)

OR (TIME OF DAY = after_hours_gv)

OR OUT OF SERVICE skillset_sk THEN

   GIVE RAN closed_ran_gv

   DISCONNECT

END IF


QUEUE TO SKILLSET skillset_sk

WAIT 2

GIVE RAN agents_busy_ran_gv

GIVE MUSIC soft_music_gv
```

```
SECTION WaitLoop
   WAIT loop_time_gv
   IF NOT QUEUED THEN
       IF OUT OF SERVICE skillset_sk THEN
             GIVE RAN sorry_day_closed_ran_gv
             DISCONNECT
       ELSE
             QUEUE TO SKILLSET skillset_sk
             WAIT 2
       END IF
   END IF
   EXECUTE WaitLoop
```

Script rules

Introduction

This section describes the rules to follow when you create your Symposium Call Center Server scripts. If you do not follow these rules, you will receive errors when you validate the script, and the call does not receive the treatment you intend.

Note: Most, but not all, of these script element restrictions are detected during script validation (before the script is activated). However, the validator cannot detect every possible situation that could cause a problem.

First command rule

Do not use the following commands as the first command executed for a call in a script. The call must first be given treatment through the use of any other command.

- WAIT
- QUIT
- GIVE SILENCE
- REMOVE FROM AGENT
- CHANGE PRIORITY IN AGENT
- REMOVE FROM SKILLSET
- CHANGE PRIORITY IN SKILLSET

Call rejection

If one of the preceding commands is encountered as the first command, the call is routed to the default ACD-DN configured for the CDN. (The call is not be queued to the default skillset or RAN.) The Symposium Call Center Server logs an error to the alarm monitor and event browser.

Lists

For many commands and skillset intrinsics, you can list up to 20 skillsets or agents. Entries in lists must be separated by commas.

Example

```
QUEUE TO SKILLSET sales_sk, service_sk, support_sk
```

Parentheses rule

Parentheses are allowed in script commands to group elements and formulas in expressions. Each open parenthesis must have a matching closing parenthesis.

Variables rule

When using a command that changes the value of a variable, that variable must be defined as a call variable.

Event Handler rules

Rule 1

If you use an Event Handler command in a script, it must be the script's first command. The script must also have a closing End Handler command.

Rule 2

The Event Handler command only applies to the script in which it appears. If the script calls a secondary script, the Event Handler no longer applies to the call. If you want the Event Handler to apply to secondary scripts, you must repeat it at the beginning of each script.

Section and Execute rules

Rule 1

Any loop that is created in a script through the Section and Execute commands must have a Wait command inside it.

Rule 2

Each Execute command must have a section label defined in the script as its target.

Note: The reverse is not true. Each section label does not need an Execute command to target it.

If-Then-End If rule

The If-Then-End If command can have multiple commands between Then and End If.

If-Then-Else-End If rule

The If-Then-Else-End If command can have multiple commands between Then and Else, as well as multiple commands between Else and End If. The Else branch executes only when the If condition is not true.

Where-Equals rule

The value used in the Where-Equals command must be an item value or an expression that evaluates to an item value. This command must be closed with an End Where command. The Default clause of the command is optional; however, Nortel Networks recommends that it always be used.

Scriptwriting tips

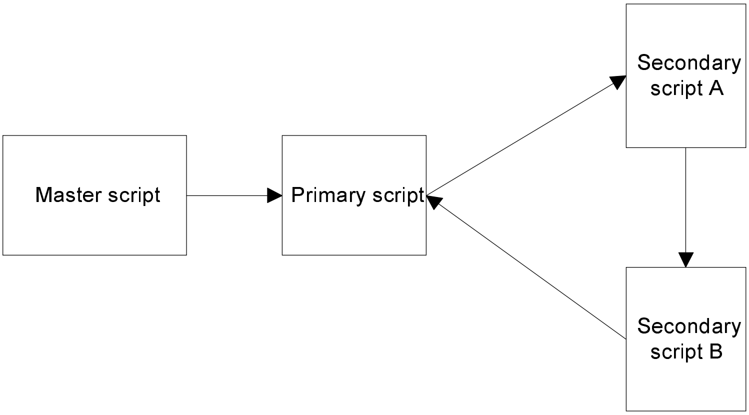
Introduction

This section gives recommendations for writing efficient scripts.

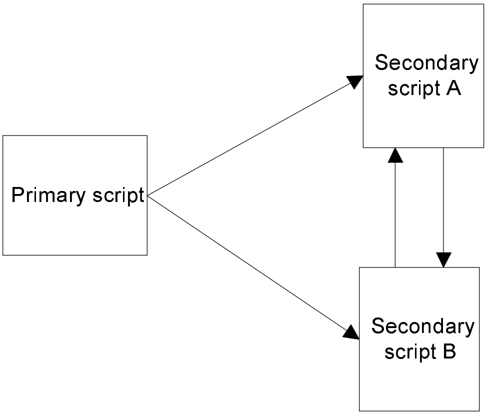
Avoid circular dependencies

When two scripts reference each other, either directly or indirectly through other scripts, they create a circular dependency. Symposium Call Center Server allows circular dependencies (with the exception that a script cannot call itself directly), but they are not recommended.

The following diagrams illustrate two ways in which circular dependencies can occur. In the first diagram, the circular dependency is created by a secondary script (script B) referencing a primary script.



In the following illustration, the circular dependency is created when two secondary scripts reference each other.



Since you cannot deactivate a script if it is being referenced by another active script, it is complicated to deactivate a script that is in a circular dependency. For more information, see "To deactivate circular-dependent scripts" on page 121.

Avoid unnecessary commands

Avoid using unnecessary commands that can decrease the efficiency of your system.

Examples

- Do not repeat a Give Music command when a call returns to a queue because music resumes automatically.
- Once a Queue to Skillset command executes, there is no need to repeat it for the same skillset.
- After Queue to Skillset, use a Wait 2 (or more) command before the next Give RAN command, if possible. If an agent is immediately available, the two-second delay prevents the system from starting to execute the Give RAN command.
- Do not use the Log command in the normal call processing path. Older events in the log file are overwritten by new events being logged. Events logged by this command can reduce the amount of historical data in the log file.

Call priority and agent priority

It is important to understand how Symposium Call Center Server uses call priorities (in the scripting language) and agent priorities (in agent to skillset assignments).

In general, avoid using call priorities (that is, the With Priority option), because they can prevent lower-priority calls from being answered.

If you use the With Priority option, Nortel Networks recommends that you either queue the calls to a backup skillset as well, or change the priorities of old calls (using the Age Of Call intrinsic) so they do not stay in queue forever.

Agent priorities do not cause calls to be left in queue. If you use agent priority and calls are waiting to be answered, Symposium Call Center Server presents calls to agents regardless of their priority.

Note: There is no relationship between call priority and agent within skillsets priority.

The following table compares call priority to agent to skillset priority.

	Call priority	Agent to skillset priority
Where is the value set?	Call priority is set or changed in the script (using the Queue To Skillset, Queue To Agent With Priority, or Change Priority commands) and can be different for each call.	Agent to skillset priority is only set in user administrator or agent to skillset assignments on the Symposium Call Center Server client PC.
Call or agent related?	Call priority is related only to the call. It is not related to agents.	Agent to skillset priority cannot be set or changed in the script, and is not call-related.

Call priority	Agent to skillset priority
Call priority is used only when there are no idle agents at the moment when the Queue To command is executed. Call priority is used at that time to decide in which position Symposium Call Center Server should place the call. When an agent becomes available, the call with the highest priority and longest wait time is presented.	Agent to skillset priority is used only when there are idle agents at the moment that the Queue To command executes. Symposium Call Center Server uses agent to skillset priority at that time to decide which agent to present the call to (in that skillset/ agent list). The agent with the highest priority and longest idle time is presented with the call.

For more information about setting agent to skillset priorities, see the *Administrator's Guide*.

Check for the most likely conditions first

If a script checks for multiple conditions, it is more efficient to check for the condition that is the 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 most commonly called number.

Example

In the following example, 2135555604 is called most often, followed by 2135555610 and 2135555612.

```
WHERE DNIS EQUALS

   VALUE 2135555604: QUEUE TO SKILLSET sales_sk WITH
PRIORITY 3

   VALUE 2135555610: QUEUE TO SKILLSET sales_sk WITH
PRIORITY 2

   VALUE 2135555612:QUEUE TO SKILLSET sales_sk WITH
PRIORITY 1

   DEFAULT: QUEUE TO SKILLSET sales_sk WITH PRIORITY 4

END WHERE
```

Check whether skillsets are out of service

Attempts to queue calls to a skillset queue that is out of service are rejected. If there is any possibility that a skillset is not staffed (for example, if the skillset does not operate 24 hours a day), use an Out-of-Service check in your scripts.

Example

```
IF OUT OF SERVICE general_sk, backup_sk THEN
   EXECUTE Night_Treatment
END IF
```

Consider what the caller hears

Consider what the caller hears (for example, ringing, silence, music, or some other tone) at all points in the script. Note that the Give Music command does not need to be repeated in a script, because music resumes after most commands (exceptions are Give Silence and Give Ringback).

Example

In the following example, if an agent does not become available, callers receive the following treatments:

- They enter the queue in the general skillset (they hear ringback when the call enters the queue).
- They hear the entire RAN agents_busy_ran_gv as long as no agents are available to answer the call.
- They hear music for 20 seconds.
- They hear the entire RAN agents_still_busy_ran_gv as long as no agents are available to answer the call.
- They hear music again until the call is answered.

```
QUEUE TO SKILLSET general_sk WITH PRIORITY 3
WAIT 2 /* Check for idle agent*/
GIVE RAN agents_busy_ran_gv
GIVE MUSIC local_station_gv
WAIT 20
```

```
GIVE RAN agents_still_busy_ran_gv
```

Default treatments

Some commands automatically have treatments associated with them. For example, if a call receives no treatment before it is queued to a skillset by the Queue To Skillset 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.

If no treatment has been given to the caller earlier in the script, the following commands automatically give a ringback tone when a call enters a queue and when a call is presented to an agent:

- QUEUE TO SKILLSET
- QUEUE TO AGENTSEND INFO
- SEND REQUEST
- LOG

Remember that the default treatment after a caller hears a RAN is silence (unless a Give Music command was given before the RAN). If you want the caller to hear anything other than silence (music or ringback), you must insert the appropriate command.

Emergency situations

You can use variables to handle emergency situations at your call center. For example, you can create a variable named “emergency” and set its value to True. For an example of how you can use this variable in a script, see Chapter 12, “Using sample scripts.”

In some emergency situations (for example, ones that requires evacuation), you might not have time to change the value of the emergency variable before you leave the call center. For these situations, you could do the following procedure:

To create an emergency skillset

1 Create an emergency agent. Give the agent a login ID that is easy to remember, such as 0911.

2 Assign the agent to the emergency skillset.

3 Include the following lines in all loops in all scripts:

```
IF NOT OUT OF SERVICE emergency_sk THEN
    <statements>
    EXECUTE Emergency_Situation
END IF
```

As an alternative, you can also use the following lines:

```
IF (LOGGED AGENT COUNT emergency_sk > 0) THEN
    EXECUTE Emergency_Situation
END IF
```

4 When an emergency situation arises, you can quickly log on the emergency agent to any phoneset. The emergency skillset is immediately placed in service when the agent logs on. The statements you provided for emergency situations take effect immediately.

Giving tones

If no treatment has been given to the caller earlier in the script, Symposium Call Center Server automatically gives a ringback tone when a call enters a queue and when it 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.

Example 1

The following example gives music to a caller waiting for an agent to answer. When the Queue To statement is executed, the caller hears a burst of the ringback before hearing the specified music. The caller hears ringback again when the call is presented to an agent.

```
QUEUE TO SKILLSET service_sk

WAIT 2

GIVE MUSIC pop_music_gv
```

Example 2

The following example shows how to give the caller a full cycle of ringback before the next treatment. This method forces a six-second delay before giving tones to ensure a more natural-sounding ringback cycle.

```
QUEUE TO SKILLSET sales_sk

WAIT 6

GIVE MUSIC classical_music_gv
```

High traffic conditions

If your script starts with a significant number of commands that do not give a treatment (such as Where-Equals or If tests, or Execute or Execute Script commands) before the first call treatment, the caller hears a second or two of silence before the Symposium Call Center Server gives treatment. Under high traffic conditions, the four-second response time (imposed by the switch) can expire, causing calls to default intermittently.

Note: It would take more than 50 If tests to present a problem. Nortel Networks recommends using no more than 50 If tests to avoid possible script execution problems.

If you think that you might encounter this problem, write the script so that it presents the treatment first (such as Give Music or Give Ringback). Begin a large number of If tests only when the caller is hearing a tone.

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 (for example, every 30 seconds).

Avoid infinite loops

To avoid endless loops, ensure that the initial Queue To Skillset command worked. For example, in "Example 1" on page 53, the Queued intrinsic checks (in each loop) whether the call is still queued.

When a call is in a loop, it is very important to ensure that the call is still queued to the skillset, and that there are agents logged on to the skillset. In regular script processing (that is, when a call is not in a loop), if the call is not queued when it reaches the end of the script, Symposium Call Center Server automatically checks whether the call is queued. If it is not, the call receives default treatment. In a loop, however, this check is never performed because there are always commands to execute.

Before queuing a call to a skillset or list of skillsets, use the Out of Service intrinsic to test the state of the skillset or use the Logged Out Agent intrinsic if you use the Queue to Agent command to queue the call.

Note: Use the Out of Service intrinsic rather than Logged Agent Count to test a skillset state. The Out Of Service intrinsic handles the transition mode (when the skillset is going out of service, but some agents are still logged in). The Logged Agent Count intrinsic does not handle the transition mode.

Any loop that you create in a script using the Section and Execute commands must contain a Wait command.

Avoid repeating commands

Avoid putting commands that do not need to be repeated inside a loop. In the following example, the Queue To Skillset statement is executed before the loop, which repeats the RAN. It is possible to include the Queue To Skillset statement in the loop, but this would cause the Queue To Skillset statement to be repeated unnecessarily. A Queue To Skillset statement that is repeated multiple times is ignored, but creates a less efficient script.

Example 1

```
QUEUE TO SKILLSET service_sk WITH PRIORITY 3

WAIT 3

/*This section checks to ensure that the call is queued,
then repeats a second recorded announcement after a 30
second pause until the call is answered.*/

SECTION Play_2nd_RAN

   WAIT 30

   IF NOT QUEUED THEN

       IF NOT OUT OF SERVICE service_sk THEN

             QUEUE TO SKILLSET service_sk WITH PRIORITY 1

             WAIT 2

       ELSE

             EXECUTE Help_Me_Now

       END IF

   END IF


   GIVE RAN agents_still_busy_ran_gv

   EXECUTE Play_2nd_RAN

SECTION Help_Me_Now

   ...
```

Checking for conditions

Use caution 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 skillset happens to be checked when the call has waited exactly ten 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.

Example 2

In the following example, the priority of the call does not change until the call has waited more than two minutes. The condition Age of Call (in the Check_Age section) removes the call from the Check_Age loop to the Change_Priority section, which begins only when the call has waited two minutes. The first Wait statement delays the script from beginning the loop until the system has had time to check if there is an agent available.

```
QUEUE TO SKILLSET service_sk WITH PRIORITY 3

WAIT 2

/* This section of the script tests the age of the call
every 10 seconds. If the call has been in queue longer than
120 seconds, the script jumps to another section to
increase the call’s priority in queue. It also checks
whether the call is queued, and if not, it jumps to another
section.*/

SECTION Check_Age

   WAIT 10

   IF NOT QUEUED THEN

       EXECUTE Help_Me_Now

   END IF

   IF (AGE OF CALL > 120) THEN

       EXECUTE Change_Priority

   END IF

   EXECUTE Check_Age

SECTION Change_Priority

   CHANGE PRIORITY IN SKILLSET service_sk TO PRIORITY 2

   WAIT 2
   EXECUTE WaitLoop
SECTION Help_Me_Now
```

Ranges

Ranges for variables include all numbers in the range, including the start and end range values. For example, a range of 1 .. 4155552323 includes every CLID from 1–415555 2323. The starting and ending values should usually have the same number of digits (for example, 4155552134 .. 4155552688).

For time ranges, the whole minute at the end of the time range is included. Therefore, the time range 08:00 .. 09:00 is actually 61 minutes. For a 60-minute period, use the range 08:00 .. 08:59.

Use caution in specifying ranges 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.

Example

This script uses a special section to treat VIP callers; otherwise, calls are queued to the general skillset. If the call is not answered within 30 seconds, the caller receives a message informing him or her that the call center is closed.

```
IF (CLID = vip_list_gv) THEN
   EXECUTE Vip_Treatment
ELSE
   QUEUE TO SKILLSET general_sk WITH PRIORITY 3
   WAIT 2
   EXECUTE WaitLoop
END IF
/* This section of the script gives special treatment to
calls in the VIP variable list */
```

```
SECTION Vip_Treatment
   QUEUE TO SKILLSET general_sk WITH PRIORITY 1
   WAIT 2
   GIVE RAN special_callers_ran_gv
/* This section of the script first checks to see if the
call has been queued, then it checks if there are any
available agents for the general skillset. If there are no
available agents the caller receives a message indicating
the call center is closed. Otherwise the call is queued to
the general skillset.*/
SECTION WaitLoop
   WAIT 30
   IF NOT QUEUED THEN
       IF OUT OF SERVICE general_sk THEN
             GIVE RAN day_closed_ran_gv
             DISCONNECT
       ELSE
             QUEUE TO SKILLSET general_sk WITH PRIORITY 1
             WAIT 2
       END IF
   END IF
   EXECUTE WaitLoop
```

Time comparisons

To be meaningful, time comparisons should include > = or < =. For example, given the following commands

```
IF (TIME OF DAY = 08:00)
```

and

```
IF (TIME OF DAY > = 08:00)
AND (TIME OF DAY < 09:00)
```

the first expression is true for one minute, while the second expression is true for one hour. A better way to express this would be to use a range, as in the following example:

```
IF (TIME OF DAY = 08:00 .. 08:59)
```

Variables

Variables make your scripts easier to modify. For example, if you use the variable "business_hours" to represent the hours that your company is open (say 9 a.m. to 5 p.m.), and you expand your hours to 8 a.m. to 6 p.m., you only need to update the variable value. You do not have to modify all of your scripts.

Naming restrictions

Do not use keywords, skillset names, or section labels to name variables. Variables must have unique names.

To avoid errors, define all of your variables before you write your scripts. If you define a variable that is not referenced by a script, delete it after you have written all of your scripts.

For more information about variables, see Chapter 4, "Working with script variables."

Script example using formatting conventions and rules

Introduction

The following script example combines many of the tips given in this chapter. (Note: In this case, the call center is not open 24 hours a day.)

Sample script

```
/* Check to see if both skillets that can answer calls are
out of service and disconnect the caller if they are. */

IF OUT OF SERVICE general_sk, backup_sk THEN

   EXECUTE Night_Treatment

END IF

/* 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. vip_list is a variable. */

IF (CLID = vip_list_gv) THEN

   EXECUTE Special_Handling

END IF

/* Check to see if there are already more than twice as many
calls queued as there are agents logged in and give the
caller a busy signal if this is true. */

IF (QUEUED CALL COUNT general_sk) > (2 * LOGGED AGENT

COUNT general_sk) THEN

   GIVE BUSY

END IF

/* Queue the caller to the general skillset and give an
announcement followed by music. */

QUEUE TO SKILLSET general_sk WITH PRIORITY 3
```

```
WAIT 6

GIVE RAN agents_busy_ran_gv

GIVE MUSIC classical_music_gv

/* This section of the script tests the age of the call
every 20 seconds. If the call has been in the queue longer
than 2 minutes, the script jumps to another section that
increases the call’s priority in queue. If the call is not
queued, the caller hears a message informing them that the
call center is closed.*/

SECTION Check_Age

/* Check if call still queued - if not, Call Center must
have closed */

   WAIT 20

   IF NOT QUEUED THEN

       EXECUTE Night_Treatment

   END IF

   IF (AGE OF CALL > 120) THEN

       EXECUTE Raise_Priority

   END IF

   GIVE RAN agents_still_busy_ran_gv

   EXECUTE Check_Age

SECTION Raise_Priority

   CHANGE PRIORITY IN SKILLSET general_sk TO PRIORITY 1

   WAIT 2

   QUEUE TO SKILLSET backup_sk WITH PRIORITY 3

   WAIT 2

SECTION Keep_RAN_Loop

/* Check if call still queued - if not, Call Center must
have closed */

   WAIT 20

   IF NOT QUEUED THEN
```

```
       EXECUTE Night_Treatment

   END IF

   GIVE RAN agents_still_busy_ran_gv

   EXECUTE Keep_RAN_Loop

/* This section of the script is reached if the caller has
a CLID that is in the vip variable list. These callers are
queued with high priority to two queues and given a special
RAN. */

SECTION Special_Handling

   QUEUE TO SKILLSET general_sk WITH PRIORITY 1

   WAIT 2

   QUEUE TO SKILLSET backup_sk WITH PRIORITY 3

   WAIT 2

   GIVE RAN you_are_special_ran_gv

   GIVE MUSIC classical_music_gv

   EXECUTE Keep_Ran_Loop

/* This section of the script plays the closed announcement
and disconnects the caller. This will only happen if both
the general skillset and the backup skillset are out of
service. */

SECTION Night_Treatment

   GIVE RAN closed_ran_gv

   DISCONNECT
```

Logging on to a Symposium Call Center Server system

Introduction

Before you can create or edit scripts for Symposium Call Center Server, you need to log on to the system.

Assumptions

This procedure assumes the following details:

- The site and systems you want to access have been set up and configured in the SMI Workbench.

  To set up sites and systems, refer to the *Software Installation and Upgrade Guide*.
- You know the user ID and password to log on to Symposium Call Center Server. If you do not know this information, contact your system administrator or your call center administrator.

Logging on for the first time

If you are logging on to Symposium Call Center Server for the first time after the system has been installed, refer to the *Setup Guide*. Otherwise, follow this procedure to log on to the system.

Once you have logged on to the system, you can create a desktop shortcut to reduce the number of steps involved in the procedure. For information on how to create a desktop shortcut, refer to the *Administrator’s Guide*.

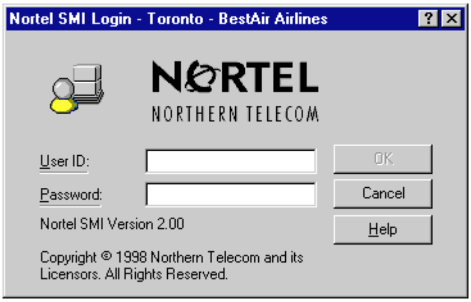
To log on to the system

1. From the Start menu, select Programs → SMI Workbench.

   Result: The SMI Workbench window appears.

2 Double-click the icon of the system to which you want to connect.

Result: The Login dialog box appears.



Nortel SMI Version 2.00

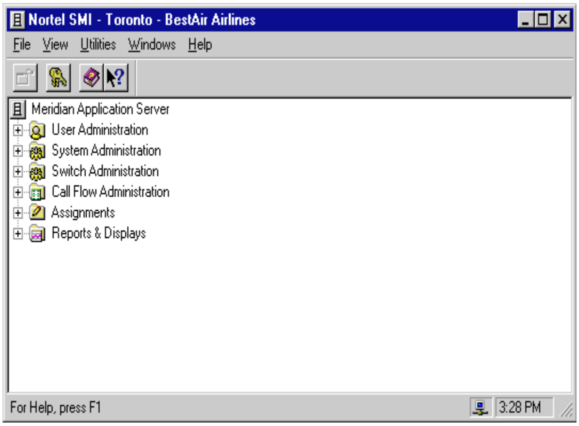
Copyright © 1998 Northern Telecom and its Licensors. All Rights Reserved.

3 Enter your user ID and password.

Note: If you do not know this information, contact your system administrator.

4 Click OK.

Result: The SMI window appears.



For Help, press F1

Result: If a connection could not be made, an error message appears.

Chapter 4

Working with script variables

In this chapter

Overview

Introduction

Before you create your Symposium Call Center Server scripts, create the variables you plan to use for your system. This chapter explains how to create and assign values to variables, how to change the values assigned to variables, and how to delete variables.

What are script variables?

Script variables are placeholders that you can use in scripts to represent values that are defined outside of the script. More than one script can use the same variable. To change the value of a variable, you only have to change the definition of the variable—scripts are not interrupted. Changes can be implemented easily, without interrupting call center operation.

All script variables are either global variables or call variables.

What are global variables?

Global variables are script variables that you can use in any script on the system. The value of a global variable can be changed only in the Script Variables window. It cannot be changed in the script.

Example

BestAir Airlines has a global variable named “business_hours_cv” that is assigned a value of 8:00 a.m. to 5:00 p.m. BestAir uses this variable in its script as follows:

```
IF (TIME OF DAY = business_hours_cv) THEN
   GIVE RAN open_ran_gv
END IF
```

Updating global variables

When you change the value of a global variable, calls that are already active in the Symposium Call Center Server system do not use the new value unless they branch to a new script (using the Execute Script command). New calls use the new global variable values immediately.

What are call variables?

Call variables are script variables whose value can change on a call-by-call basis. The value of a call variable follows the call through the system and is passed from one script to the next with the call. The initial value of a call variable is assigned in the Script Variables window. Each incoming call uses this value unless it is changed in the script by one of the following commands:

- ASSIGN TO
- GET RESPONSE

Example

You can use the Assign To command in a script to give a value to a variable for the duration of a call. The following example assigns the value of the total number of active calls, divided by the call rate, plus 10 to the variable int_var.

```
ASSIGN (TOTAL ACTIVE CALLS / CALL RATE + 10) TO int_var
```

Updating call variables

When you make changes to the initial value of a call variable in the Script Variables window, calls that are already active in the Symposium Call Center Server system do not use the new value (call variable values stay constant throughout the life of the call). The new value takes effect for new calls.

You can create up to 20 call variables. If you attempt to create a 21st variable, you will receive an error message.

There is no limit to the number of global variables that you can create.

Call variables and blind transfer calls

When a call is transferred, the Symposium Call Center Server combines the call variables of the original call and those of the consultative call. This creates a list of call variables that includes some of the original call variables and some of the consultative call’s variables. This only happens for blind transfers.

When a call first arrives and begins execution of a master script, it starts with an empty list of call variables. Then, as it goes through the scripts, each call variable that it uses is copied into the script’s private list of variables. If the scripts are written so that the original incoming call accesses some call variables, and the consultation call accesses different call variables, when the transfer is complete (blind transfer only), a single list is created that contains both the original call’s call variables plus the consultative call’s call variables. The original call starts again at the top of the master script, but this time with the combined list of call variables. If the same call variable is used by both the original call and the consultant call, the value of the consultant call is applied to the variable once the call is complete.

Sets of values in variables

Some variables can have a set of values instead of only a single value. For example, you can create a variable for holidays that includes all holidays for the year. Variables with sets of values can include up to 60 items.

Notes:

1. Call variable must be a single item only.
2. Replacement of a specific agent with an agent variable does not require you to modify your scripts if the agent leaves the call center and is replaced by another agent.

Opening the Script Variables window

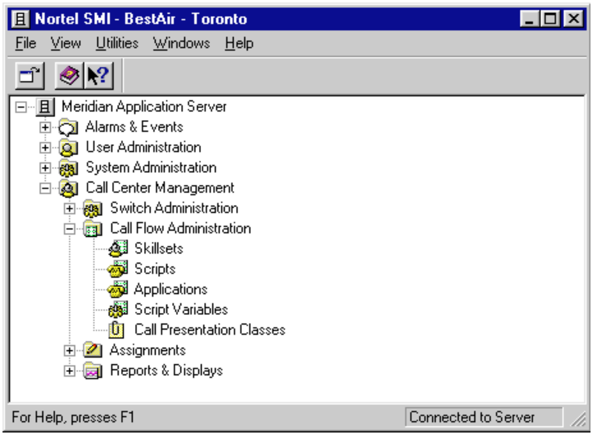
Introduction

Use the Script Variables window to list the script variables on the system. From this window, you can create or delete a script variable. You can also access the Script Variable Properties dialog box where you can change script variable properties.

Note: You cannot delete Script variables while they are referenced by any activated scripts.

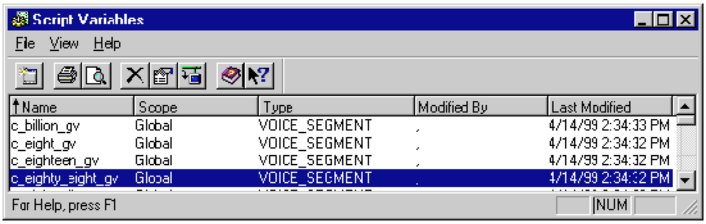
To open the Script Variables window

1 From the SMI window, choose Call Flow Administration.



2 Double-click Script Variables.

Result: The Script Variables window appears.



Name	Scope	Type	Modified By	Last Modified
c_billion_gv	Global	VOICE_SEGMENT	,	4/14/99 2:34:33 PM
c_eight_gv	Global	VOICE_SEGMENT	,	4/14/99 2:34:32 PM
c_eighteen_gv	Global	VOICE_SEGMENT	,	4/14/99 2:34:32 PM
c_eighty_eight_gv	Global	VOICE_SEGMENT	,	4/14/99 2:34:32 PM

Types of variables

Introduction

The following table lists the types of variables you can create.

Data type	Class allowed	Leading zero allowed	Valid value	Example
Agent ID	Item or set	Yes	16-digit number	93453
Boolean	Item	No	True or False	True
CDN	Item or set	No	10 digits (negatives not allowed) If you enter any number of digits from 1–10, the script compiles without errors. However, for the script to work properly, the CDN must be exactly 10 digits.	8900
CLID	Item or set	Yes	1–32 digits (negatives not allowed)	4165552244
Day	Item or set	No	Monday to Sunday	Tuesday
Day of month	Item or set	No	1–31	23
Date	Item or set	No	Jan 1 to Dec 31 January 1 to December 31	Sep 3 September 3
Digits	Item	No	1–32 digits, numbers only. This is the variable type used for the call variable that stores caller entered data.	5552356
DN	Item or set	Yes	1–32 digits (negatives not allowed)	5552356

Data type	Class allowed	Leading zero allowed	Valid value	Example
DNIS	Item or set	Yes	10 digits (negatives not allowed) If you enter any number of digits from 1–10, the script compiles without errors. However, for the script to work properly, the DNIS must be exactly 10 digits.	3427872334
Integer	Item	No	Number from –1999999999–199999999	22938
Language	Item	No	French, German, English, Spanish, Portuguese, Chinese, Japanese	French
Month	Item or set	No	Jan to Dec January to December	Mar March
Music	Item	No	Music route, 0–512	73
Priority	Item	No	Numbers 1–6	5
RAN	Item	No	RAN route, 0–512	72
Seconds	Item	No	0–65535	10
Skillset	Item or set	No	1–30-character string (no spaces allowed)	French_sales
String	Item	No	1 to 80 characters	Log message
Time	Item or set	No	0:00 to 23:59	11:15
Wildcard	Item or set	No	1- to 32-digit number containing wildcard @ or placeholder ? symbols	416@ 41?

Creating script variables

Introduction

Follow this procedure to create script variables in Symposium Call Center Server.

Before you begin

Before you begin to create variables, all system resources such as RAN routes, music routes, voice ports, call treatments, and CDNs must be set up. For more information about setting up these resources, refer to the *Switch Guide.*

Additionally, all agents and skillsets must be created. For more information about creating agents and skillsets, refer to the *Administrator's Guide.*

Naming script variables

When you name script variables, ensure that you meet the following requirements:

- Script variable names must be unique. They cannot be the same as skillset names, script language keywords, or intrinsics. For a list of script language keywords, see Appendix B, "Scripting keywords."
- Script variable names must begin with an alphabetic character and cannot contain spaces.
- Valid characters for script variable names are A–Z, a–z, 0–9, and _ (underscore).

Tips:

1. When possible, give variables generic names so you can reuse them in different scripts.
2. To help you identify types of variables when you are writing and editing your scripts, include information about the variable type in its name. For example, name a global variable for a greeting RAN "greeting_RAN_gv" or name a call variable for caller-entered data "caller_data_cv."

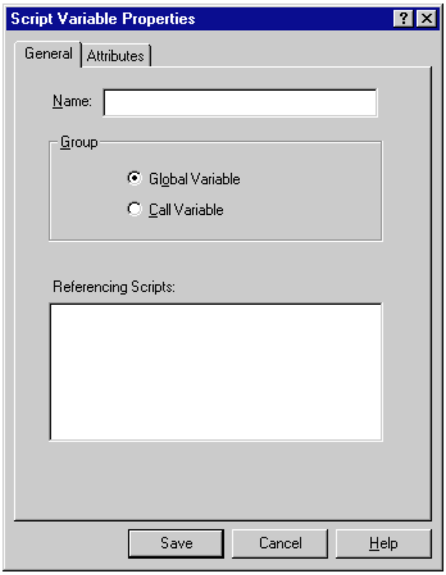
To add variables

1 From the SMI window, choose Call Flow Administration → Script Variables.

Result: The Script Variables window appears.

2 Choose File → New.

Result: The Script Variables Properties window appears.



Group
● Global Variable
○ Call Variable

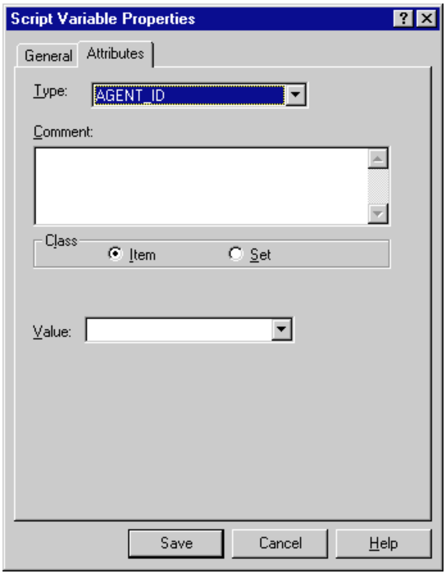
3 On the General property page, enter information in the following fields:

Name: Enter the name of the variable. For information about naming variables, see “Naming script variables” on page 73.

Scope: Select Global or Call Variable.

For more information about global and call variables, see “What are global variables?” on page 66 and “What are call variables?” on page 67.

4 Click the Attributes tab.



Type: AGENT_ID

Class: ● Item ○ Set

5 Enter information in the following fields:

Type: The data type of the variable. For more information, see “Types of variables” on page 71.

Comment*:* A description of the variable.

Class: Whether the variable has a single value (Item) or a set of values (Set).

6 Choose one of the following actions:

a. To assign a single value to the variable, see “To assign single values” on page 76.

b. To assign more than one value to the variable (that is, a set), see “To assign a set of values” on page 77.

c. To assign a range of values to the variable (that is, a set), see “To assign a range of values” on page 78.

Assigning values to variables

Introduction

Item class variables can have only a single value. Set class variables can have a set of values (for example, assign several agent IDs to a variable) or a range of values. For example, assign a range of days (Monday to Friday) to a Set class variable.

To assign single values

1 From the SMI window, choose Call Flow Administration ➝ Script Variables.

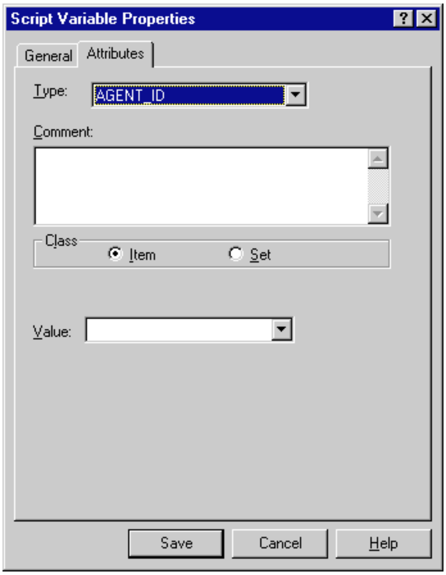
Result: The Script Variables window appears.

2 Select the script variable to which you want to assign a value. For this example, we are adding a value to the variable skillset_A.

3 Choose File ➝ Properties.

Result: The Script Variables Properties window appears.

4 Click the Attributes tab.



Script Variable Properties

Type: AGENT_ID

Comment:

Class ● Item ○ Set

Value:

5 For Class, select Item if it is not already selected.

6 In the Value box, type or select the value you want to assign to the variable.

7 Click Save.

To assign a set of values

1 From the SMI window, choose Call Flow Administration → Script Variables.

Result: The Script Variables window appears.

2 Select the script variable to which you want to assign a set of values. For this example, we are assigning two agents (Harfrey Ng and Ed Simpson) to an agent ID variable.

3 Choose File → Properties.

Result: The Script Variables Properties window appears.

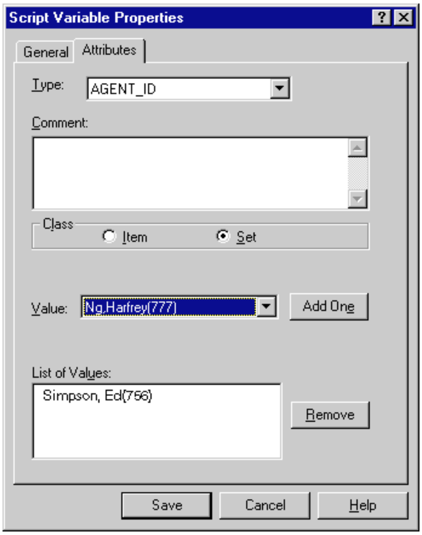
4 Click the Attributes tab.

5 For Class, select Set if it is not already selected.

6 In the Value box, type or select the values that you want to assign to the variable.

Note: You can include up to 60 items in the set.

7 Click Add One.



Type: AGENT_ID

Comment:

Class ○ Item ● Set

Value: Ng,Harfrey(777)

List of Values:
Simpson, Ed(756)

8. Repeat steps 6 and 7 for each value you want to include in the set.
9. Click Save.

To assign a range of values

If the class of a variable is Set, and the Type is:

- Day

or

- Day of Month

or

- Month

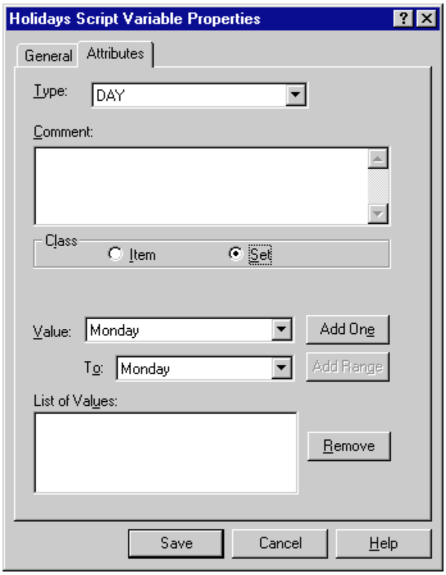
you can assign a range of values to the variable.

1. From the SMI window, choose Call Flow Administration → Script Variables.

   Result: The Script Variables window appears.
2. Select the script variable to which you want to assign a range of values. For this example, we are assigning a range of values (Monday to Friday) to the variable business_days.
3. Choose File → Properties.

   Result: The Script Variables Properties window appears.

4 Click the Attributes tab.



Holidays Script Variable Properties

Type: DAY

Comment:

Class ○ Item ● Set

Value: Monday

To: Monday

List of Values:

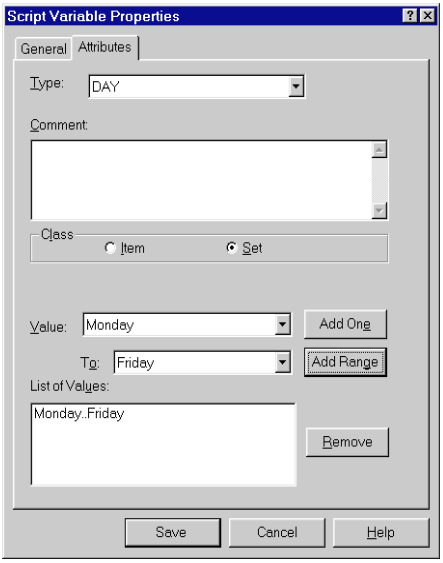
5 In the Value box, type or select the beginning of the range.

6 In the To box, select the end of the range.

Note: The To box appears only if the variable class is set to Set and Type is Day, Day of Month, or Month.

7 Click Add One or Add Range.

Note: These buttons appear only if the script variable class is set to Set and the Type is Day, Day of Month, or Month, or if there is a predefined list of values from which to choose.



Script Variable Properties

Type: DAY

Comment:

Class: ○ Item ● Set

Value: Monday

To: Friday

List of Values: Monday..Friday

8 Click Save.

To remove values from sets

1 From the SMI window, choose Call Flow Administration → Script Variables.

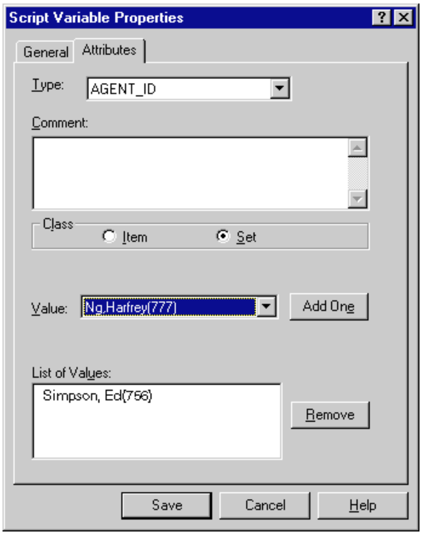
Result: The Script Variables window appears.

2 Select the script variable from which you want to remove a value.

3 Choose File → Properties.

Result: The Script Variables Properties window appears.

4 Click the Attributes tab.



Type: AGENT_ID

Class: ○ Item ● Set

Value: Ng,Harfrey(777)

List of Values: Simpson, Ed(756)

5 In the List of Values box, select the value that you want to remove.

6 Click Remove, then click Save.

Checking variables for referencing scripts

Introduction

Follow this procedure to check whether a variable is referenced by any active scripts.

If a script variable is referenced by any active scripts, you cannot change its properties (except for the value), or rename or delete it.

To check for referencing scripts

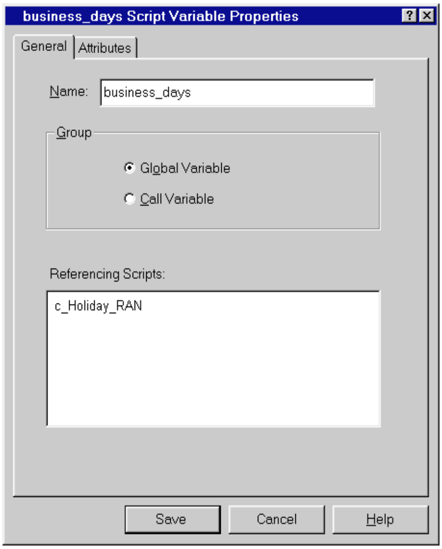
1 From the SMI window, choose Call Flow Administration → Script Variables.

Result: The Script Variables window appears.

2 Select the script variable that you want to check.

3 Choose File → Properties.

Result: The Script Variables Properties window appears.



business_days Script Variable Properties

Name: business_days

Group

● Global Variable

○ Call Variable

Referencing Scripts:

c_Holiday_RAN

The Referencing Scripts section lists the names of scripts that reference this variable.

4 If any activated scripts appear in the list, take one of the following actions:

   a. Deactivate the script. For more information, see “Activating and deactivating scripts” on page 117.

   b. Remove the reference to the variable from the referencing script.

5 Click Save to close the Script Variable Properties page.

Changing script variable properties

Introduction

Follow this procedure to change the properties of a script variable. You cannot change the class of a script variable, you can only change the value. For more information, see “Checking variables for referencing scripts” on page 82.

Note: You cannot change the variable name or group type. If you want to change the name or group type, you must delete the variable and create it again.

To change variable properties

1. From the SMI window, choose Call Flow Administration → Script Variables.

   Result: The Script Variables window appears.

2. Select the script variable that you want to change.
3. Choose File → Properties.

   Result: The Script Variables Properties window appears.

4. Click the Attributes tab.
5. Change information in the following fields as necessary:

   Comment: A description of the variable.

   Class: Whether the variable has a single value (Item) or a set of values (Set).

   Value: The new value of the variable.

   List of Values: If you chose Set, enter the new list of values.

   Note: You cannot change the variable type. If you want to change the type, you must delete the variable and create it again.

6. Click Save.

Deleting script variables

Introduction

Follow this procedure to remove a script variable from the system.

You cannot delete a script variable if it is referenced by any active scripts. For more information, see “Checking variables for referencing scripts” on page 82.

To delete variables

1 From the SMI window, choose Call Flow Administration → Script Variables.

Result: The Script Variables window appears.

2 Select the script variable that you want to delete.

3 Choose File → Delete.

4 Click Yes to confirm that you want to delete the variable.

Chapter 5

Creating and administering scripts

In this chapter

Overview

Introduction

Section A, “Creating scripts,” explains how to

- create a new script
- use the Scripts Editor to edit an existing script
- add script elements to scripts
- copy sections between scripts, including the sample scripts which are provided with the Symposium Call Center Server Client software

Section B, “Importing and exporting scripts,” gives procedures to import and export scripts, including the sample scripts that are provided with the Symposium Call Center Server Client software.

Section C, “Administering scripts,” explains how to administer scripts. It gives procedures to

- validate scripts and resolve validation errors
- activate and deactivate scripts
- save changes to scripts
- rename scripts
- delete scripts

Section A: Creating scripts

In this section

Creating new scripts

When to use

Follow the procedure in this section to add a new script to the system.

Before you begin

Before you begin to create scripts, Nortel Networks strongly recommends that you read see Chapter 3, “Scripting rules and conventions,” carefully. This chapter lists the system resources that must be set up before you can use your scripts. It also outlines the script planning process.

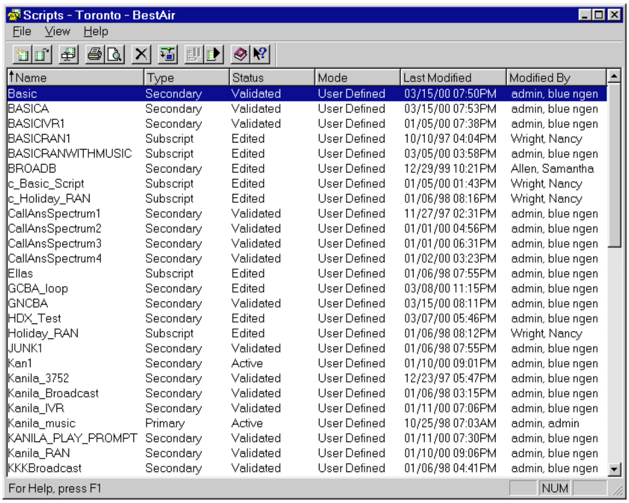
Script limits

Single scripts cannot exceed 30 000 characters. If you reach 30 000 characters in a single script, an error message appears.

To add new scripts

1 From the SMI window, choose Call Flow Administration → Scripts.

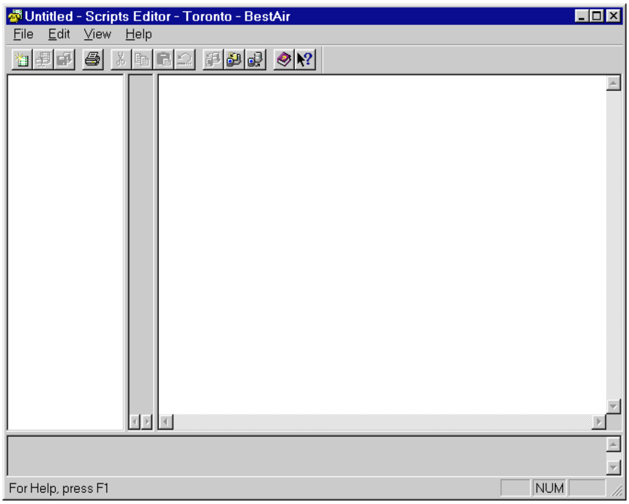
Result: The Script Manager appears.



Name	Type	Status	Mode	Last Modified	Modified By
Basic	Secondary	Validated	User Defined	03/15/00 07:50PM	admin, blue ngen
BASICA	Secondary	Validated	User Defined	03/15/00 07:53PM	admin, blue ngen
BASICIVR1	Secondary	Validated	User Defined	01/05/00 07:38PM	admin, blue ngen
BASICRAN1	Subscript	Edited	User Defined	10/10/97 04:04PM	Wright, Nancy
BASICRANWITHMUSIC	Subscript	Edited	User Defined	03/05/00 03:58PM	admin, blue ngen
BROADB	Secondary	Edited	User Defined	12/29/99 10:21PM	Allen, Samantha
c_Basic_Script	Subscript	Edited	User Defined	01/05/00 01:43PM	Wright, Nancy
c_Holiday_RAN	Subscript	Edited	User Defined	01/06/98 08:16PM	Wright, Nancy
CallAnsSpectrum1	Secondary	Validated	User Defined	11/27/97 02:31PM	admin, blue ngen
CallAnsSpectrum2	Secondary	Validated	User Defined	01/01/00 04:56PM	admin, blue ngen
CallAnsSpectrum3	Secondary	Validated	User Defined	01/01/00 06:31PM	admin, blue ngen
CallAnsSpectrum4	Secondary	Validated	User Defined	01/02/00 03:23PM	admin, blue ngen
Ellas	Subscript	Edited	User Defined	01/06/98 07:55PM	admin, blue ngen
GCBA_loop	Secondary	Edited	User Defined	03/08/00 11:15PM	admin, blue ngen
GNCBA	Secondary	Validated	User Defined	03/15/00 08:11PM	admin, blue ngen
HDX_Test	Secondary	Edited	User Defined	03/07/00 05:46PM	admin, blue ngen
Holiday_RAN	Subscript	Edited	User Defined	01/06/98 08:12PM	Wright, Nancy
JUNK1	Secondary	Validated	User Defined	01/06/98 07:55PM	admin, blue ngen
Kan1	Secondary	Active	User Defined	01/10/00 09:01PM	admin, blue ngen
Kanila_3752	Secondary	Validated	User Defined	12/23/97 05:47PM	admin, blue ngen
Kanila_Broadcast	Secondary	Validated	User Defined	01/06/98 03:15PM	admin, blue ngen
Kanila_IVR	Secondary	Validated	User Defined	01/11/00 07:06PM	admin, blue ngen
Kanila_music	Primary	Active	User Defined	10/25/98 07:03AM	admin, admin
KANILA_PLAY_PROMPT	Secondary	Validated	User Defined	01/11/00 07:30PM	admin, blue ngen
Kanila_RAN	Secondary	Validated	User Defined	01/10/00 09:06PM	admin, blue ngen
KKKBroadcast	Secondary	Validated	User Defined	01/06/98 04:41PM	admin, blue ngen

2 Choose File → New.

Result: The Script Editor appears.



3 Enter the text of the new script.

You can add commands, variables, intrinsics, and so on manually, or you can use the Script Command Reference panel. For more information, see “Adding script elements to scripts” on page 93.

4 Choose File → Save.

5 Click OK.

6 For Name, type the name of the new script.

Note: Script names are not case-sensitive; however, the first character in each script name must not be a numeral. The name you assign to a new script must be unique. You cannot enter the name of an existing script. If the script is referenced from the Master script, the name given to the script becomes the application name and it appears on Real-Time displays and historical reports. You must also consider customer requirements when viewing the applications. Does the customer want all relevant applications for each department to appear together? If so, then the applications must be in alphabetical order.

7 Click OK.

Adding script elements to scripts

Introduction

Follow the procedure in this section to insert script elements from the Script Command Reference panel into a script. The Script Command Reference panel provides a list of all the available script elements. You can select an element from the list and insert it into your script.

Note: You can type commands into your scripts manually, but if you use the Script Command Reference panel, all parameters that you have to replace are automatically inserted as well. This can help to reduce errors in your scripts.

Script elements

You can insert all of the following types of script elements from the Script Command Reference panel:

Commands

Basic commands include basic call processing elements general elements. For more information about these commands, see Chapter 6, “Basic script commands.”

Advanced

Advanced commands include basic call processing elements and host connectivity elements. For more information about these commands, see Chapter 7, “Advanced script commands.”

ATTENTION

Advanced commands are keycoded options. If you insert an advanced command into your script without purchasing these options, the script does not validate.

Intrinsics

Intrinsics include skillset, time, traffic, and call intrinsics. For more information about intrinsics, see Chapter 9, “Intrinsics.”

Variables

Variables include global variables and call variables. For more information, see Chapter 4, “Working with script variables.”

Events

Events are unsolicited events or failed responses that you can instruct the Event Handler to check for. For more information, see “Event Handler” on page 175.

Operators

Operators include logical, mathematical, and relational operators. For more information, see Chapter 10, “Script expressions.”

To add elements to scripts

1. From the SMI window, choose Call Flow Administration → Scripts.

   Result: The Script Manager appears.

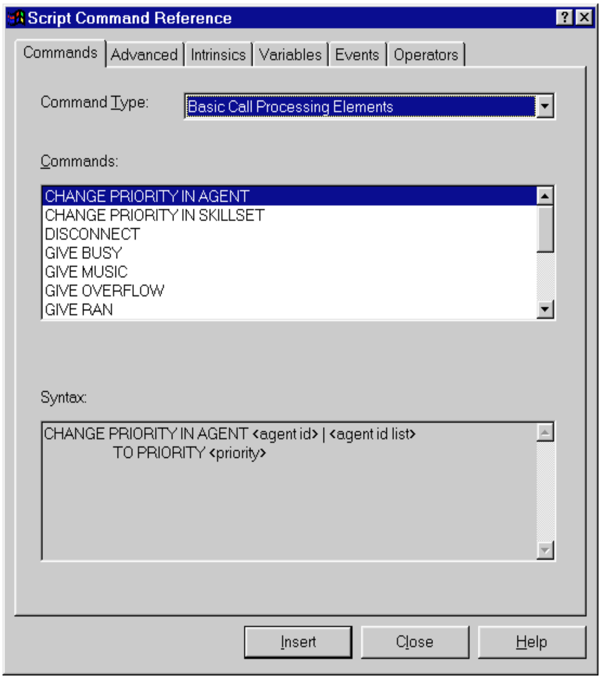
2. Double-click the script to which you want to add an element.

   Result: The Scripts Editor appears. The left section of the window shows the selected script and its subscripts in the tree view. The middle section of the window shows the line numbers of the currently selected script. The right section shows the text of the currently selected script. The bottom section of the window shows the errors that occurred during validation of the selected script.

   Tip: Open the Master script to see a view of all referenced scripts.

3. Position the cursor at the point where you want to insert the element.

4 Choose View → Script Commands.



Command Type: Basic Call Processing Elements

Commands:

CHANGE PRIORITY IN AGENT
CHANGE PRIORITY IN SKILLSET
DISCONNECT
GIVE BUSY
GIVE MUSIC
GIVE OVERFLOW
GIVE RAN

Syntax:

```
CHANGE PRIORITY IN AGENT <agent id> | <agent id list>
          TO PRIORITY <priority>
```

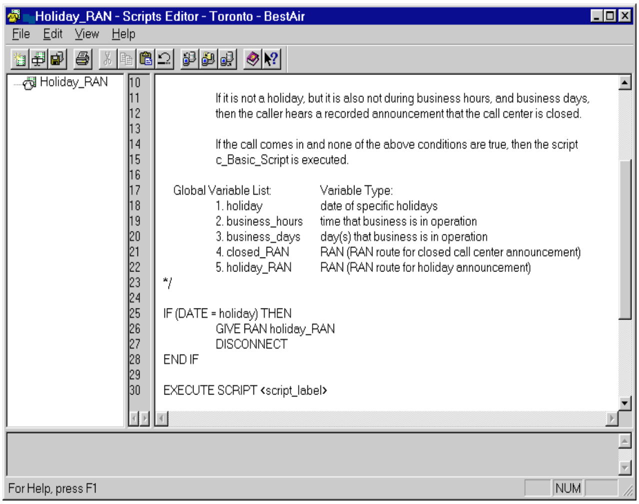
5 Click the property page of the element type that you want to insert. In this example, the Execute Script command has been inserted, which is on the Commands page.

6 Select the Command type from the drop-down list. For the Execute Script command example, select General Elements.

7 Select the element that you want to insert in the script.

8 Click Insert.

Result: The element appears in the script.



```
            If it is not a holiday, but it is also not during business hours, and business days,
            then the caller hears a recorded announcement that the call center is closed.

            If the call comes in and none of the above conditions are true, then the script
            c_Basic_Script is executed.

  Global Variable List:         Variable Type:
            1. holiday          date of specific holidays
            2. business_hours   time that business is in operation
            3. business_days    day(s) that business is in operation
            4. closed_RAN       RAN (RAN route for closed call center announcement)
            5. holiday_RAN      RAN (RAN route for holiday announcement)
*/

IF (DATE = holiday) THEN
            GIVE RAN holiday_RAN
            DISCONNECT
END IF

EXECUTE SCRIPT <script_label>
```

9 Edit the script for the new command. For example, to edit the script for the new Execute Script command, replace the parameter <script_label> with the name of the script you want to execute.

10 Repeat step 5 to step 9 for each element that you want to insert.

11 When you are finished, click Close.

Copying text into scripts

Introduction

You do not have to retype a section of script text that already exists. Instead, you can copy the text from one script and paste it into another. Similarly, you can copy text from a text document into your script.

Tip: Nortel Networks recommends that you do not make changes directly to scripts in the Scripts Editor. Instead, copy the text of the script, paste it into Notepad, and make your changes in Notepad. This prevents loss of data should an error occur on the server before you can save your changes.

Copying from sample scripts

You can copy parts of the sample scripts that are provided with the Symposium Call Center Server client software into your own script. To do so, import the script that you want to use. For instructions, see “Importing scripts into Symposium Call Center Server” on page 106. Once you have imported the script, follow the procedure on page 99 to copy a script, or sections of a script, into your own script.

Example

Nancy Wright writes scripts for BestAir Airlines. To create a new script, she does not need to type the entire script manually. Instead, she copies the text of a sample script, c_Basic. This is the text of c_Basic:

```
IF (DATE = holidays_gv)

OR (DAY OF WEEK = weekends_gv)

OR (TIME OF DAY = closed_hours_gv) THEN

   GIVE RAN closed_ran

   DISCONNECT

END IF
```

```
IF OUT OF SERVICE skillset_sk THEN
   GIVE RAN dayclosed_ran
   DISCONNECT
END IF

QUEUE TO SKILLSET skillset_sk
WAIT 2   /* Allow time in case an agent is available */
GIVE RAN first_ran

SECTION WaitLoop
    GIVE MUSIC music_route
    WAIT treatment_timer_gv
    IF NOT QUEUED THEN
        IF OUT OF SERVICE skillset_sk THEN
             GIVE RAN dayclosed_ran
             DISCONNECT
   ELSE
       QUEUE TO SKILLSET skillset_sk
       WAIT 2   /* Allow time in case an agent is available
*/
       END IF
    END IF
    GIVE RAN second_ran
EXECUTE WaitLoop
```

Nancy imports this script, and then copies the text into her own script. Finally she replaces the variables and parameters with BestAir’s system information.

To copy sections between scripts

1 From the SMI window, choose Call Flow Administration → Scripts.

Result: The Script Manager appears.

2 Double-click the script from which you want to copy.

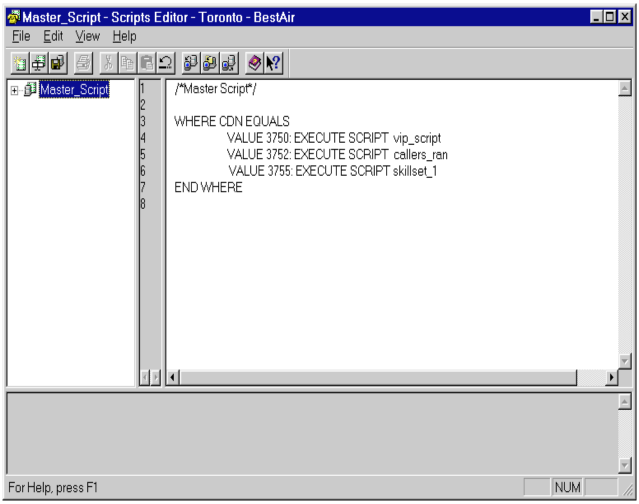
Result: The Script Editor appears.

3 Select the section of the script you want to copy.

4 If you want to permanently remove the section from the script, choose Edit → Cut. Otherwise, choose Edit → Copy.

5 Choose File → Close to close the script.

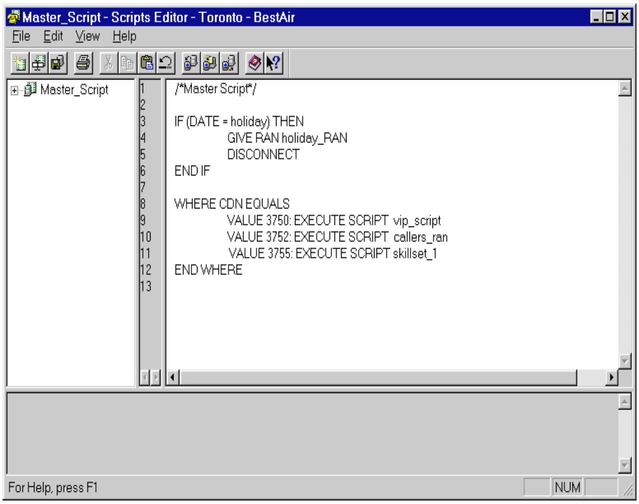
6 In the Script Manager, double-click the script into which you want to copy the text.



```
/*Master Script*/

WHERE CDN EQUALS
          VALUE 3750: EXECUTE SCRIPT vip_script
          VALUE 3752: EXECUTE SCRIPT callers_ran
          VALUE 3755: EXECUTE SCRIPT skillset_1
END WHERE
```

7 Place the cursor where you want to insert the copied text.

8 Choose Edit → Paste.



```
/*Master Script*/

IF (DATE = holiday) THEN
          GIVE RAN holiday_RAN
          DISCONNECT
END IF

WHERE CDN EQUALS
          VALUE 3750: EXECUTE SCRIPT vip_script
          VALUE 3752: EXECUTE SCRIPT callers_ran
          VALUE 3755: EXECUTE SCRIPT skillset_1
END WHERE
```

Result: The text is pasted into the script.

9 Choose File → Save to save your changes.

To copy text from other applications

1 Open the document that contains the text that you want to copy.

2 Select the text that you want to copy.

3 If you want to permanently remove the section from the document, choose Edit → Cut. Otherwise, choose Edit → Copy.

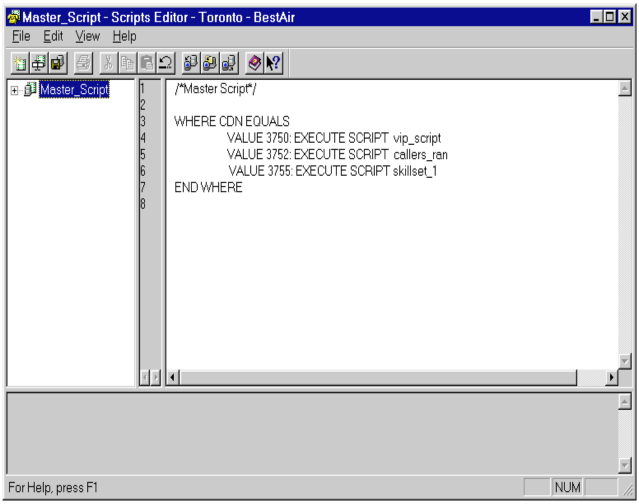
4 Close or minimize the application.

5 From the SMI window, choose Call Flow Administration → Scripts.

Result: The Script Manager appears.

6 Double-click the script into which you want to copy the text.

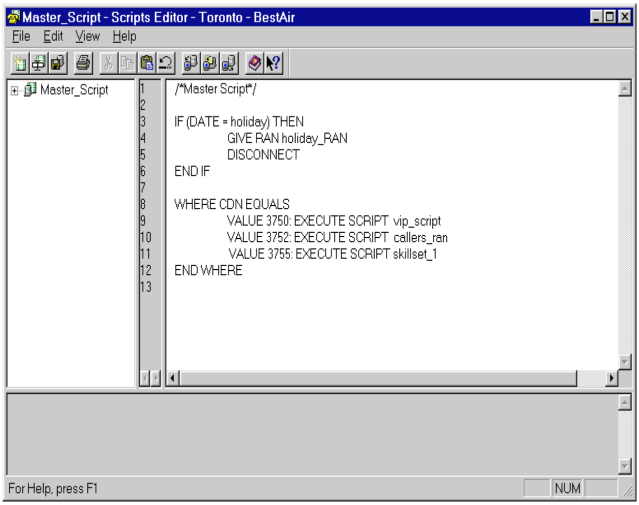
Result: The Script Editor appears.



```
/*Master Script*/

WHERE CDN EQUALS
          VALUE 3750: EXECUTE SCRIPT vip_script
          VALUE 3752: EXECUTE SCRIPT callers_ran
          VALUE 3755: EXECUTE SCRIPT skillset_1
END WHERE
```

7 Place the cursor where you want to insert the copied text.

8 Choose Edit → Paste.



```
/*Master Script*/

IF (DATE = holiday) THEN
          GIVE RAN holiday_RAN
          DISCONNECT
END IF

WHERE CDN EQUALS
          VALUE 3750: EXECUTE SCRIPT vip_script
          VALUE 3752: EXECUTE SCRIPT callers_ran
          VALUE 3755: EXECUTE SCRIPT skillset_1
END WHERE
```

Result: The text is pasted into the script.

9 Choose File → Save to save your changes.

Saving changes to scripts

Saving changes to an activated script

You can make changes to a script while it is activated. You can then choose to either activate the script with the changes immediately, or save the script without putting the changes into service right away. There are two ways to save a script that is currently activated:

- If you want to put the script into service immediately after making any changes to it, activate the script.
- If you do not want to put the script into service immediately after making changes to it, save the script using a different name.

Note: When a change to a script is activated (under the existing name of the script), calls in progress are processed using the old version of the script, and new calls are processed using the new version of the script.

If the activation fails (due to a compilation error), the change to the script is not saved. New calls still use the old version of the script.

To save changes to scripts

1 If the script that you want to save changes is not already open, from the SMI window, choose Call Flow Administration → Scripts.

Result: The Script Manager appears. If the script to which you want to save changes is already open, skip to step 3.

2 Double-click the script name in the Script Manager.

Result: The Script Editor appears.

3 Edit the script as necessary.

4 Choose one of the following actions:

a. To save the script using the current name, choose File → Save, then go to step 7.

b. If the script is active and you want to use the new version of the script immediately, choose File → Activate.

- **c.** To save the script using a new name, choose File → Save As.

5 Click OK to confirm that you want to save the script.

6 For Name, type the new name of the script.

7 Click OK.

Section B: Importing and exporting scripts

In this section

Importing scripts into Symposium Call Center Server

Import command

Use the Import command to copy an existing script from your local hard drive, a network drive, or a floppy disk into the current script. This command adds the text of the imported script to any text in the current script.

Note: You cannot import a script that was created in another application, such as Notepad. However, if you create a text document that you want to use as a script, you can copy the text into a script in the Script Editor. For more information, see "To copy text from other applications" on page 100.

Importing sample scripts

To use the sample scripts that come with the Symposium Call Center Server client software, you must first import them. To do so, follow the procedure on this page.

To import scripts

1. From the SMI window, choose Call Flow Administration → Scripts.

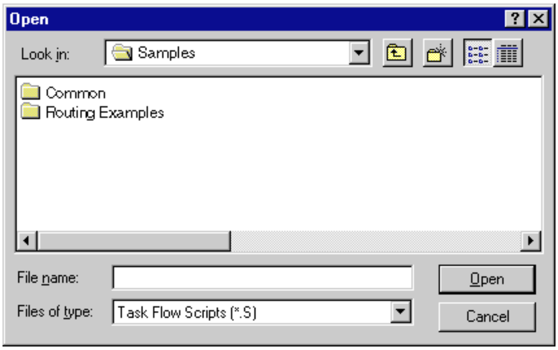
   Result: The Script Manager appears.

2. Double-click the script into which you want to import another script. (Choose File → New if you want to import the script into a new script.)

   Result: The Script Editor appears.

3 Choose File → Import.

Result: A dialog box appears prompting you for the location of the script you want to import.



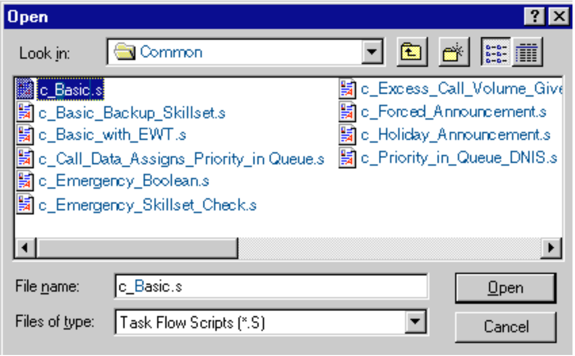
Look in: Samples

Files of type: Task Flow Scripts (*.S)

4 Navigate to the file you that want to import. Sample scripts are in the C:\Nortel\client\en\scripts\samples directory. The samples directory contains two subdirectories. For this example, choose the common directory.

5 Select the file that you want to import.

Result: The name of the file appears in the File name box.


Open
?
c_Basic.s
c_Basic_Backup_Skillset.s
c_Basic_with_EWT.s
c_Call_Data_Assigns_Priority_in Queue.s
c_Emergency_Boolean.s
c_Emergency_Skillset_Check.s
c_Excess_Call_Volume_Give
c_Forced_Announcement.s
c_Holiday_Announcement.s
c_Priority_in_Queue_DNIS.s
Open
Cancel

Look in: Common

File name: c_Basic.s

Files of type: Task Flow Scripts (*.S)

6 Click Open to import the script.

Result: The text of the imported file appends to any text that was in the Scripts Editor.

7 Modify the imported script as necessary.

Note: The script that you import might contain references to variables. Variables are not imported with the script. You must define the variables on your system. For more information, see “Creating script variables” on page 73.

Exporting scripts to a remote location

Export command

The Export command outputs the script you currently have open into a file on your local hard drive, or to a disk, with the file extension .s.

ATTENTION

If there is already a script in this location with the same name as the exported script, the exported script overwrites it.

To export to a new file

1. From the SMI window, choose Call Flow Administration → Scripts.

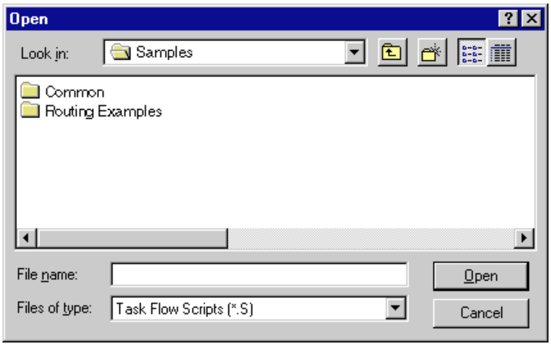
   Result: The Script Manager appears.

2. Double-click the file that you want to export.

   Result: The Script Editor appears.

3. Choose File → Export.

   Result: A dialog box appears prompting you for the location where you want to export the script.

4. For File name, type the name of the new script file.

5 Click Save.

Result: A copy of the script is placed in the location you specified.

To export to an existing file

1 From the SMI window, choose Call Flow Administration → Scripts.

Result: The Script Manager appears.

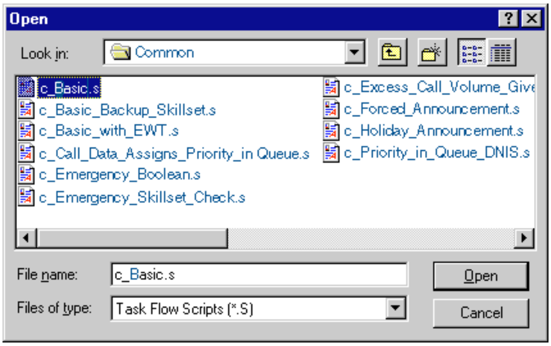
2 Double-click the file that you want to export.

Result: The Script Editor appears.

3 Choose File → Export.

Result: A dialog box appears prompting you for the location where you want to export the script.

4 Navigate to the file you want to overwrite.



Look in: Common

c_Basic.s
c_Basic_Backup_Skillset.s
c_Basic_with_EWT.s
c_Call_Data_Assigns_Priority_in Queue.s
c_Emergency_Boolean.s
c_Emergency_Skillset_Check.s
c_Excess_Call_Volume_Give
c_Forced_Announcement.s
c_Holiday_Announcement.s
c_Priority_in_Queue_DNIS.s

File name: c_Basic.s

Files of type: Task Flow Scripts (*.S)

5 Click Save.

6 Click OK to confirm that you want to replace the existing file.

Section C: Administering scripts

In this section

Validating scripts

Introduction

Follow the procedure in this section to validate a script. If you want to save a script without validating it (for example, you plan to continue working on it later), see “Saving changes to scripts” on page 102.

What is script validation?

Before a script is put into service or “activated,” it must be checked to ensure that the syntax and semantics are correct. This process is called validation. If the script does not contain any errors, validation results in an executable version of the script. If the script contains errors, validation results in a list of those errors and the corresponding lines of the script where the errors occurred.

If you create a new script or modify an existing non-active script, you must validate the script manually before it can be activated. If you are working with an active script, you must save any changes by activating the new version of the script. The activation process validates the script automatically.

Note: The system attempts to activate all referenced scripts when a script is activated. However, the referenced scripts are *not* validated automatically. You must validate each referenced script separately before you activate the script.

To validate scripts

1 From the SMI window, choose Call Flow Administration → Scripts.

Result: The Script Manager appears.

2 Double-click the script that you want to validate.

Result: The Script Editor appears.

3 Choose Edit → Validate.

4 Click OK to begin the validation process.

5 If you have not saved this script, the server prompts you to save it now. Enter a name for the script and click OK.

Result: The results of the validation process appear in the lower pane of the Script Manager window.

6 If the script does not contain any errors, a message telling you that the operation was successfully completed appears. You can now activate the script. For more information, see “Activating and deactivating scripts” on page 117.

If the script contains errors, you must correct them before you can activate it. For more information, see “Resolving validation errors” on page 114.

Resolving validation errors

Introduction

Follow the procedure in this section to resolve errors that can result when you validate a script. The Scripts Editor lists the errors with the number of the line on which the errors occurred. Once a script is free of errors, you can activate it.

Note: You can save a script without validating it. However, you cannot activate it if it has not validated successfully.

To resolve validation errors

1. From the SMI window, choose Call Flow Administration → Scripts.

   Result: The Script Manager appears.

2. Double-click the script that you want to validate.

   Result: The Script Editor appears.

3. Choose Edit → Validate.
4. Click OK to begin the validation process.
5. If you have not saved this script, the server prompts you to save it now. Enter a name for the script and click OK.
6. Review any errors that appear in the bottom section of the screen. For information about the errors and what they mean, see "List of validation errors" on page 347.
7. Make the required corrections to the lines of the script that appear in the error messages.
8. Choose Edit → Validate.
9. Continue to correct the errors and revalidate the script until no further errors appear.

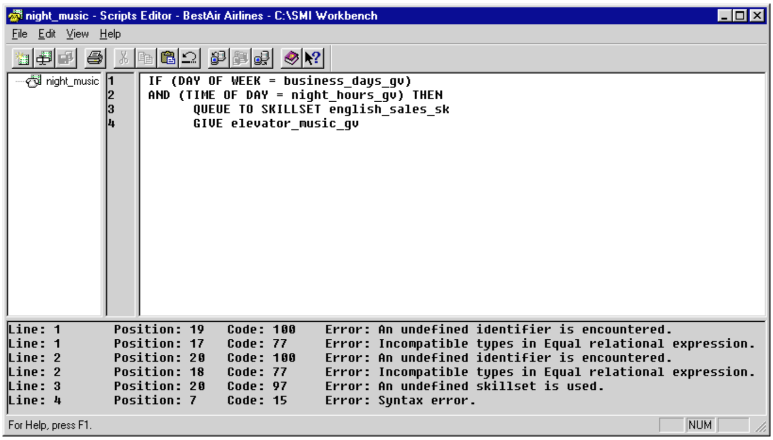
To interpret script error messages

If you try to validate the following section of script text

```
IF (DAY OF WEEK = business_days_gv)
```

```
AND(TIME OF DAY = night_hours_gv) THEN
    QUEUE TO SKILLSET english_sales_sk
    GIVE elevator_music_gv
```

the following errors appear:


night_music - Scripts Editor - BestAir Airlines - C:\SMI Workbench
File Edit View Help
night_music
For Help, press F1.
NUM

```
IF (DAY OF WEEK = business_days_gv)
AND (TIME OF DAY = night_hours_gv) THEN
      QUEUE TO SKILLSET english_sales_sk
      GIVE elevator_music_gv
```

```
Line: 1        Position: 19   Code: 100    Error: An undefined identifier is encountered.
Line: 1        Position: 17   Code: 77     Error: Incompatible types in Equal relational expression.
Line: 2        Position: 20   Code: 100    Error: An undefined identifier is encountered.
Line: 2        Position: 18   Code: 77     Error: Incompatible types in Equal relational expression.
Line: 3        Position: 20   Code: 97     Error: An undefined skillset is used.
Line: 4        Position: 7    Code: 15     Error: Syntax error.
```

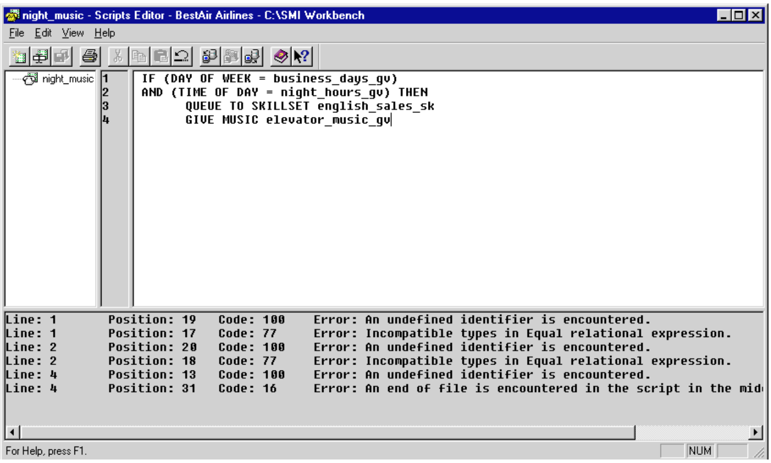
To correct these errors, do the following tasks:

- Define the skillset “english_sales_sk” in the Skillset window.
- Complete the Give command by adding the word “Music.”

This is how the script looks now:

```
IF (DAY OF WEEK = business_days_gv)
AND(TIME OF DAY = night_hours_gv) THEN
    QUEUE TO SKILLSET english_sales_sk
    GIVE MUSIC elevator_music_gv
```

If you try to validate the script, the following errors appear:


night_music - Scripts Editor - BestAir Airlines - C:\SMI Workbench
File Edit View Help
night_music
For Help, press F1.
NUM

```
IF (DAY OF WEEK = business_days_gv)
AND (TIME OF DAY = night_hours_gv) THEN
      QUEUE TO SKILLSET english_sales_sk
      GIVE MUSIC elevator_music_gv
```

```
Line: 1        Position: 19    Code: 100    Error: An undefined identifier is encountered.
Line: 1        Position: 17    Code: 77     Error: Incompatible types in Equal relational expression.
Line: 2        Position: 20    Code: 100    Error: An undefined identifier is encountered.
Line: 2        Position: 18    Code: 77     Error: Incompatible types in Equal relational expression.
Line: 4        Position: 13    Code: 100    Error: An undefined identifier is encountered.
Line: 4        Position: 31    Code: 16     Error: An end of file is encountered in the script in the mid
```

To correct these errors, do the following actions:

- Define the variables “business_days_gv,” “night_hours_gv,” and “elevator_music_gv” in the Script Variables window.
- Complete the If-Then-End If command by adding “End If” to the script.

This is how the script looks now:

```
IF (DAY OF WEEK = business_days_gv)
AND (TIME OF DAY = night_hours_gv) THEN
   QUEUE TO SKILLSET english_sales_sk
   WAIT 2
   GIVE MUSIC elevator_music_gv
END IF
```

If you validate the script again, the request is successful—there are no more errors.

For a list of validation errors and what they mean, see “List of validation errors” on page 347.

Activating and deactivating scripts

Limit for activated scripts

There is no limit on the number of scripts you can create on your Symposium Call Center Server. However, the maximum number of scripts that you can activate at one time is 1000.

What is script activation?

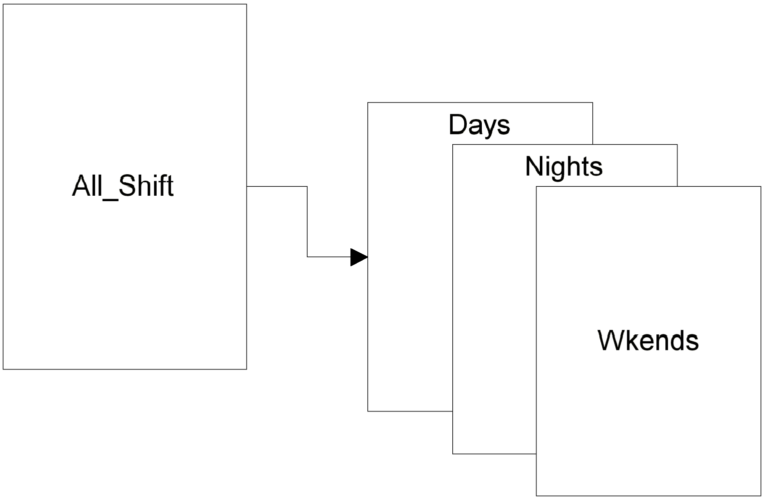
An activated script is a script that is processing calls or is in an active state ready to process calls.

If you create a new script or modify an existing non-active script, you must validate the script manually before it can be activated. If you are working with an active script, you must save any changes by activating the new version of the script. The activation process validates the script automatically.

Note: The system attempts to activate all referenced scripts when a script is activated. However, the referenced scripts are *not* validated automatically. You must validate each referenced script separately before you activate the script.

Example

One of BestAir's scripts, the All_Shift script, references the Days, Nights, and Wkends scripts. The All_Shift script is the primary script, and the others are secondary scripts. Chris Harris, BestAir's system administrator, must validate the secondary scripts before she can activate the primary script.



What is a deactivated script?

A deactivated script is a script that does not process any new calls. If any calls are already in the system using the script when you deactivate it, the script remains active for those calls until they are completed.

You can deactivate a script only if it is not referenced by any other active scripts.

Note: You cannot deactivate the Master script. However, you can change the contents and activate the new version.

To activate scripts

1. From the SMI window, choose Call Flow Administration → Scripts.

   Result: The Script Manager appears.

2. Choose one of the following options:

   a. Select the script that you want to activate.

   b. To activate the script in this Scripts Editor, double-click the name of the script.

3 Choose File → Activate.

Result: If the script has not been validated before this point, the validation process begins. Otherwise, the script is activated immediately.

To deactivate scripts

1 From the SMI window, choose Call Flow Administration → Scripts.

Result: The Script Manager appears.

2 Choose one of the following options:

a. Select the script that you want to deactivate.

b. To deactivate the script in this Scripts Editor, double-click the name of the script.

3 Choose File → Deactivate.

Notes:

1. You cannot deactivate a script if it is referenced by another active script. You must first either deactivate the referencing script or remove the reference.
2. To deactivate scripts that have circular dependencies, see "Deactivating scripts with circular dependencies" on page 120.

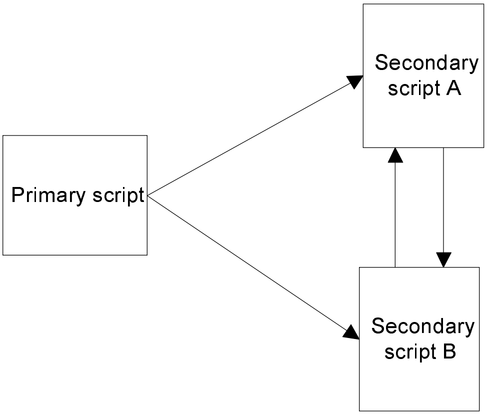
Deactivating scripts with circular dependencies

What are circular dependencies?

When two scripts reference each other (directly or indirectly through other scripts), they have a circular dependency. Symposium Call Center Server allows circular dependencies (with the exception that a script cannot reference itself directly), but they are not recommended. Since you cannot deactivate a script if it is referenced by another active script, you have to break the dependency first.

Example

The following illustration shows how a circular dependency is created when two scripts reference each other.



For example, if you want to deactivate secondary script A, you must first delete the reference to secondary script A from the primary script and from secondary script B. Then secondary script A can be deactivated.

To deactivate circular-dependent scripts

1 From the SMI window, choose Call Flow Administration → Scripts.

Result: The Script Manager appears.

2 Select a script that references the script you want to deactivate. In the example shown on page 120, this would be either the primary script or secondary script B.

3 Double-click the script name.

Result: The Scripts Editor appears.

4 Remove the Execute Script command that causes the circular dependency.

5 Validate and activate this new version of the script.

6 Repeat step 2 to step 5 for each script that references the script you want to deactivate.

When all of the referencing scripts are changed, the circular dependency is broken.

7 You can now deactivate the script that is no longer referenced by the script in step 1.

When calls are still active in the system, you might have to wait until active calls are complete before the system allows the deactivation of the referenced scripts.

Renaming scripts

Introduction

You can rename a script in either the Script Manager or the Scripts Editor. Symposium Call Center Server does not create a copy of the existing script with the old name. Ensure that you meet the following requirements when you rename a script:

- Deactivate the script. For more information, see “To deactivate scripts” on page 119.
- Give the script a unique name. You cannot enter the name of an existing script. Spaces are not allowed.

To rename scripts

1 From the SMI window, choose Call Flow Administration → Scripts.

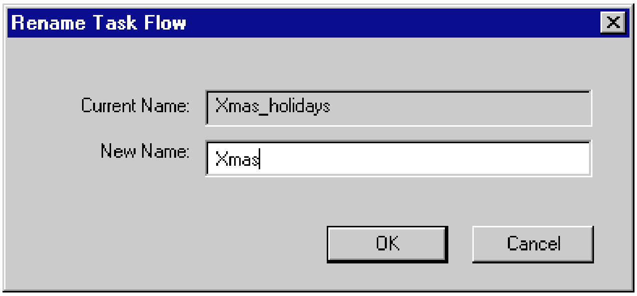
Result: The Script Manager appears.

2 Choose one of the following options:

a. To rename the script in the Script Manager, select the script you want to rename.

Choose File → Rename.

Result: The Rename Task Flow window appears prompting you for the new name of the script.



Current Name: Xmas_holidays

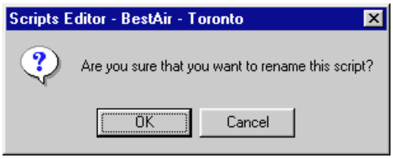
New Name: Xmas

Enter a new name for the script, and then click OK.

b. To rename the script in the Scripts Editor, double-click the script you want to rename.

Choose File → Rename.

Result: A dialog box appears prompting you to confirm the renaming of the script.



Are you sure that you want to rename this script?

Click OK.

Result: The Rename Task Flow window appears prompting you for the new name of the script.

Enter a new name for the script, and then click OK.

Note: The new name you assign to the script must be unique. You cannot enter the name of an existing script.

Deleting scripts

When to use

Follow the procedure in this section to remove a script from the system.

Before you begin

Ensure that the script is deactivated before it is removed from the system. For more information, see “Activating and deactivating scripts” on page 117.

To delete scripts

1. From the SMI window, choose Call Flow Administration → Scripts.

   Result: The Script Manager appears.

2. Select the script that you want to delete.
3. Choose File → Delete.
4. Click OK to confirm the deletion of the script.

   Note: When you delete a script, you do not delete any of its referenced variables. If no other scripts use the variables, delete them. For more information, see “Deleting script variables” on page 85.

Chapter 6

Basic script commands

In this chapter

Overview

Introduction

This chapter describes the basic script commands to which all Symposium Call Center Server systems have access. It shows how to write the commands in your scripts (script syntax), along with the parameters and optional segments that make up the commands.

Section A, “Basic general commands,” describes the general commands that you can use in your scripts. These include commands such as If-Then-Else-End If, Execute, and Quit.

Section B, “Basic call processing commands,” describes the basic call processing commands that you can use in your scripts. These include commands such as Disconnect, Give Ringback, Queue To Agent, and Route Call.

Section A: Basic general commands

In this section

Assign To

Introduction

Use the Assign To command in your scripts to assign values to call variables. For more information about call variables, see Chapter 4, “Working with script variables.”

ATTENTION

The Assigned command has been changed to the Assign To command. The Assigned command does continue to execute, however, Nortel Networks recommends that you use the Assign To command in your scripts.

Syntax change

The syntax of this command has changed from previous product releases. You can still use the old syntax, Assigned, in Symposium Call Center Server Release 1.5 scripts.

Script syntax

ASSIGN **<value>** TO **<variable>**

Parameters

Enter information for the following parameters:

- **<variable>** The name of the call variable to which you want to assign a value during the execution of the script. You can only use call variables of type Item.
- **<value>** The value that you want to assign to the call variable. Replace this parameter with an item or an expression.

Example 1

The following command assigns a value of 10 to the variable int_var.

```
ASSIGN 10 TO int_var_cv
```

Example 2

The following example assigns the value of the total number of active calls, divided by the call rate, plus 10 to the variable int_var.

```
ASSIGN (TOTAL ACTIVE CALLS / CALL RATE) + 10 TO int_var_cv
```

Example 3

The following example allows the section “Loop” to repeat three times.

```
ASSIGN 0 TO counter_cv
SECTION Loop
   WAIT 30
   GIVE RAN all_agents_busy_ran_gv
   IF (counter_cv = 3) THEN
       EXECUTE SCRIPT Finished
   END IF
   ASSIGN counter_cv + 1 TO counter_cv
   EXECUTE Loop
```

Execute

Introduction

Use the Execute command to branch to a section in the same script.

After the execution of the specific section, the execution of the script continues to the end of the script or until it encounters a Quit command. It does not return to the section of the script from which it was executed.

Script syntax

EXECUTE **<section_label>**

Parameter

Enter information for the following parameter:

- **<section_label>** The name of the section that you want to execute.

Example

In the following example, if the number of calls queued to the sales skillset exceeds 20, the section named “Estimate_wait_section” is executed. Otherwise, the caller hears music until the call is answered.

```
IF (QUEUED CALL COUNT sales_sk > 20) THEN

   EXECUTE Estimate_Wait_Section

ELSE

   GIVE MUSIC pop_music_gv

   QUIT

END IF

SECTION Estimate_Wait_Section

   ...
```

Execute Script

Introduction

Use the Execute Script command to branch from one script to another.

Note: The call branches to the referenced script, and does not return at the end of the referenced script.

Script syntax

EXECUTE SCRIPT **<Script_Name>**

Parameter

Enter information for the following parameter:

- **<Script_Name>** The name of the script that you want to execute.

Example

In the following example, the Execute Script command is used to run different scripts depending on the day and time that a call comes in to the Call Center.

```
IF (DAY OF WEEK = weekend_gv) THEN

   EXECUTE SCRIPT Script_B

END IF

IF (DAY OF WEEK = business_days_gv)

AND (TIME OF DAY = business_hours_gv) THEN

   EXECUTE SCRIPT Script_C

END IF
```

If-Then-End If

Introduction

Use the If-Then-End If command in conditional situations. If the specified condition is met, the Then statements are executed. Otherwise, the script skips to the next command.

An If-Then-End If command can have multiple statements between Then and End If.

An If command can appear within the Then or Else clause of another If command, and each If must have exactly one matching End If command.

Script syntax

IF **<logical_condition>** THEN **<statements>** END IF

Parameters

Enter information for the following parameters:

- **<logical_condition>** The condition for which you want the script to test.
- **<statements>** The action that you want the script to take if the condition is met.

Example

In the following example, callers whose CLID is included in the vip_list variable receive special treatment, including a recorded announcement, and are queued to the preferred customer skillset with high priority.

```
IF (CLID = vip_list_gv) THEN

   GIVE RAN special_callers_ran_gv

   QUEUE TO SKILLSET preferred_cust_sk WITH PRIORITY 1

   WAIT 2
   GIVE MUSIC classical_music_gv
END IF
```

If-Then-Else-End If

Introduction

Use the If-Then-Else-End If command in conditional situations. If the specified condition is met, then the statements are executed. Otherwise, the script executes the Else statements.

An If-Then-Else-End If command can have multiple statements between Then and Else, as well as multiple statements between Else and End If.

An If command can appear within the Then or Else clause of another If command, and each If must have exactly one matching End If command.

Script syntax

IF **<logical_condition>** THEN **<statement1>** ELSE **<statement2>** END IF

Parameters

Enter information for the following parameters:

- **<logical_condition>** The condition for which you want the script to test.
- **<statement1>** The action that you want the script to take if the condition is met.
- **<statement2>** The action that you want the script to take if the condition is not met.

Example 1

The following script example checks the CLID of the caller. If the CLID is 905-863-3123, then the call is queued to the customer service skillset with a priority of 1. This caller also hears a special announcement. Otherwise, the call is queued with a priority of 3 and the caller hears a different announcement.

```
IF (CLID = 9058633123) THEN
   QUEUE TO SKILLSET customer_service_sk WITH PRIORITY 1
```

```
    WAIT 2
    GIVE RAN special_ran_gv /* Special Greeting */
ELSE
    QUEUE TO SKILLSET customer_service_sk WITH PRIORITY 3
    WAIT 2
    GIVE RAN alternate_ran_gv
END IF
GIVE MUSIC classical_music_gv
```

Example 2

The following example shows how you can use an If-Then-End If command within an If-Then-Else-End If command. You must have an End If command for each If command.

Note: Each End If is associated with the nearest If command.

```
IF (CLID = vip_list_gv) THEN
    EXECUTE Vips
ELSE
    IF (CLID = special_list_gv) THEN
        EXECUTE Special
    END IF
END IF
```

Quit

Introduction

This command is necessary only when the execution of a script must end while there are still commands left to execute in the script.

Note: In general, the execution of a script stops automatically when there are no more commands left to execute. Therefore, the Quit command is not needed at the end of most scripts.

Script syntax

```
QUIT
```

When to use

Use the Quit command to terminate further script execution.

This command does not disconnect the call. For example, if the call is already queued to a skillset, the caller remains queued and continues to receive whatever tones, silence, or music were specified for the call earlier in the script. Quit only terminates the script execution, not call processing.

Restriction

The Quit command cannot be the first command in a script.

Example

In the following example, the script checks whether the customer service skillset is out of service. If it is out of service, then the script jumps to the closed section. Otherwise, the script continues to queue the call to customer service with a priority of 1. After two seconds, the caller hears a recorded announcement, and then music. The Quit command after the Give Music command prevents the script from continuing on to the closed section.

```
IF OUT OF SERVICE customer_service_sk THEN
   EXECUTE Closed
END IF
QUEUE TO SKILLSET customer_service_sk WITH PRIORITY 1
WAIT 2
GIVE RAN sevice_ran_gv
GIVE MUSIC classical_music_gv
QUIT
SECTION Closed
   GIVE RAN closed_ran_gv
   DISCONNECT
```

Section

Introduction

Use the Section command to define a section of commands. The execution of the script can jump to any section in the same script by using the Execute command.

Script syntax

SECTION **<section_name>**

Parameter

Enter information for the following parameter:

<section_name> The name of the section.

Example

The following example uses a Section command in conjunction with an Execute command to create a simple loop. The call can be requeued to remove it from the loop.

```
SECTION Wait_Loop
   WAIT wait_delay_gv
   IF NOT QUEUED THEN
       EXECUTE Requeue_Call
   END IF
   GIVE RAN please_wait_ran_gv
   EXECUTE Wait_Loop
SECTION Requeue_Call
   /* script continues here */
```

Wait

Introduction

Use the Wait command to suspend a script for a period of time before executing the next command. You must specify a length of time, otherwise the next command is executed immediately. Nortel Networks recommends that you use a Wait command with at least two seconds after you queue a call to a skillset or an agent. This allows time for the call to be answered by the agent before the next treatment begins.

Since the system checks suspended calls every two seconds, the value specified with the Wait command can be extended by up to an additional two seconds. The Wait command terminates at the first system check following the time specified with the Wait command.

For example, if you specify a Wait value of five seconds, the actual Wait time can be between six and eight seconds.

Script syntax

WAIT **<time_in_seconds>**

Parameter

Enter information for the following parameter:

- **<time_in_seconds>** The amount of time, in seconds, that you want the script to pause.

  Replace this parameter with a variable of type “seconds” or with a numerical constant.

Restriction

The Wait command cannot be the first command in a script.

Example

In the following example, calls are queued to the sales skillset. After a two-second delay, callers hear a recorded announcement asking them to wait.

```
QUEUE TO SKILLSET sales_sk

WAIT 2

GIVE RAN please_wait_ran_gv
```

Section B: Basic call processing commands

In this section

Change Priority In Agent

Introduction

Use the Change Priority In Agent command to change the priority of a call that is queued to a specified agent.

You can use the Change Priority In Agent command only to change the priority of a call in an agent queue when the call was queued by the Queue To Agent command. You cannot use it to change the priority of a call that was queued using the Queue To Skillset command.

The Change Priority In Agent command applies only to local agents.

Script syntax

CHANGE PRIORITY IN AGENT [**<agent_ID>** | **<agent_ID_list>**] TO PRIORITY **<priority>**

Parameters

Enter information for the following parameters:

- **<agent_ID>** or **<agent_id_list>** The ID of the agent to whom the call is queued, or a list of IDs of agents to whom the call is queued.
- **<priority>** The new priority with which you want the call queued to the specified agent.

Restriction

The command cannot be the first command in a script.

Example

In the following example, a call is queued to an agent variable named “agent_1” (representing an agent named Joe Smith) with a priority of 5 when it first arrives. After 20 seconds, if the call is still unanswered, the priority is changed to 2, and after 60 seconds, the priority of the call is increased to 1.

Note: When you replace a specific agent with an agent variable, you do not have to modify your scripts if the agent leaves the call center and is replaced by another agent.

```
ASSIGN 5 TO priority_cv

QUEUE TO AGENT agent_1 WITH PRIORITY priority_cv

WAIT 5

SECTION Check_Age

   GIVE RAN agent_busy

   WAIT 20

   IF (AGE OF CALL > 120) THEN

       EXECUTE Too_Long

   ELSE

       IF (AGE OF CALL > 60) THEN

             IF (priority_cv <> 1) THEN

                   ASSIGN 1 TO priority_cv

                   CHANGE PRIORITY IN AGENT agent_1 TO

                   PRIORITY priority_cv

             END IF

       ELSE

             IF (AGE OF CALL > 20) THEN

                   IF (priority_cv <> 2) THEN

                         ASSIGN 2 TO priority_cv

                         CHANGE PRIORITY IN AGENT agent_1

                         TO PRIORITY priority_cv
                   END IF
             END IF
       END IF
   END IF
EXECUTE Check_Age
```

Change Priority In Skillset

Introduction

Use the Change Priority In Skillset command to change the priority of a call in a skillset to which it is queued. The priority of the call is changed within all of the skillsets that are listed in the command. This applies only to local skillsets.

Script syntax

CHANGE PRIORITY IN SKILLSET [**<skillset> | <skillset_list>**] TO PRIORITY **<priority>**

Parameters

Enter information for the following parameters:

- **<skillset>** or **<skillset_list>** The skillset, or list of skillsets, to which the call is queued. You can also replace this parameter with an intrinsic returning a skillset or a skillset list.
- **<priority>** The new priority with which you want the call queued to the specified skillset.

Restriction

The Change Priority In Skillset command cannot be the first command in a script.

Example

In the following example, the call has been queued to the sales skillset. This section of the script tests the call’s current position in the skillset queue. If it is more than 5, the priority is raised to priority 1 (unless it is already priority 1).

```
QUEUE TO SKILLSET sales_sk

WAIT 2
```

```
IF (POSITION IN QUEUE sales_sk > 5)
AND (PRIORITY IN QUEUE sales_sk <> 1) THEN
   CHANGE PRIORITY IN SKILLSET sales_sk TO PRIORITY 1
END IF
```

Disconnect

Introduction

Use the Disconnect command to disconnect a call. The call can be disconnected at any time that it is in the CDN queue, except when it is being presented to, or has been answered by, an agent.

Script syntax

DISCONNECT

Restriction

Do not insert any commands after the Disconnect command.

Example

In the following example, calls coming in to the call center between August 10th and August 23rd (represented by a variable named “august_holidays_gv”) receive a recorded announcement stating that the call center is closed. Then the call is disconnected.

```
IF (DATE = august_holidays_gv) THEN
   GIVE RAN gone_fishing_ran_gv
   DISCONNECT
END IF
```

Give Busy

Introduction

Use the Give Busy command to provide a busy tone to a call before it is disconnected by the switch. A call can receive a busy tone any time that it is in the CDN queue, except when it is being presented to, or has been answered by, an agent.

First treatment

This treatment must be the first treatment that the call receives, otherwise, for some trunk types, the call is not disconnected automatically by the switch after giving the busy tone. This can cause trunks to be left connected until the caller disconnects the call.

Because some statements give an automatic ringback tone, it is important to analyze the script very carefully to ensure that these commands are not executed before the Give Busy command. This ensures that the busy tone is always the first treatment given to the call. For more information about commands that automatically generate a Ringback treatment, see “Default treatments” on page 49.

Script syntax

GIVE BUSY

Example

In the following example, if the number of calls queued to the sales skillset is greater than three times the number of agents logged on to the sales skillset, then the caller hears a busy tone. Otherwise, the call is queued to the sales skillset.

```
IF (QUEUED CALL COUNT sales_sk) > (LOGGED AGENT COUNT
sales_sk * 3)THEN
   GIVE BUSY
```

```
END IF
QUEUE TO SKILLSET sales_sk
```

Give Music

Introduction

Use the Give Music command to play music from a specified music route. You can use the Give Music command to play music to a call when it is waiting in a queue.

Note: If you use a Give Music command in a script, you must include a parameter specified in the Audio table. Table AUDIO defines audio routes available for broadcast for ACD groups. For more information about the Audio table, see the *Symposium and DMS Switch Guide*.

Script syntax

GIVE MUSIC **<music_route>**

Parameter

Enter information for the following parameter:

- **<music_route>** The music route that contains the music you want played to the caller.

  Replace this parameter with a variable of type “music,” or a numeric constant.

Restriction

Do not insert the Give Busy or Give Overflow command immediately after the Give Music command.

With the Wait command

Use a Wait command immediately after the Give Music command to control the duration of the music. If a call is given music followed by a Give RAN command, you must insert another Give Music command after the Give RAN. Otherwise, music does not automatically resume.

Music stops when another treatment or action with an unspecified time length is given (such as Give Ringback or Route Call). Music is suspended during presentation to an agent, and resumes if the call returns to the queue (if the agent does not answer).

Example

In the following example, a call is queued to the service skillset, and after a two-second delay, the caller hears music until an agent becomes available to take the call. In this example, the music route is represented by a variable (pop_music).

```
QUEUE TO service_sk WITH PRIORITY 3

WAIT 2

GIVE MUSIC pop_music_gv

WAIT 30
```

Give Overflow

Introduction

Use the Give Overflow command to provide an overflow tone to a call before it is disconnected by the switch. The call can be given an overflow tone at any time that it is in the CDN queue, except when it is being presented to, or has been answered by, an agent.

Note: An overflow tone is a fast busy tone.

First treatment

This treatment must be the first treatment that the call receives, otherwise, for some trunk types, the call is not disconnected automatically by the switch after giving the busy tone. This can cause trunks to be left connected until the caller disconnects the call.

Because some statements give an automatic ringback tone, it is important to analyze the script very carefully to ensure that these commands are not executed before the Give Overflow command. This ensures that the busy tone is always the first treatment given to the call. For more information about commands that automatically generate a Ringback treatment, see “Default treatments” on page 49.

Script syntax

```
GIVE OVERFLOW
```

Example

In the following example, if the number of calls queued to the sales skillset is greater than the number of agents logged on to the sales skillset plus two, then the caller receives an overflow treatment. Otherwise, the call is queued to the sales skillset.

```
IF (QUEUED CALL COUNT sales_sk) > (LOGGED AGENT COUNT
sales_sk + 2)THEN
   GIVE OVERFLOW
END IF
QUEUE TO SKILLSET sales_sk
```

Give RAN

Introduction

Use the Give RAN command to provide a recorded announcement (RAN) to a call through the specified RAN trunk.

The RAN is interrupted if an agent becomes available to take the call. Otherwise, the RAN is completed and the next command in the script is executed.

Note: If you use a Give Ran command in a script, you must include a parameter specified in the Audio table. Table AUDIO defines audio routes available for broadcast for ACD groups. For more information about the Audio table, see the *Symposium and DMS Switch Guide.*

Script syntax

GIVE RAN **<ran_route>**

Parameter

Enter information for the following parameter:

- **<ran_route>** The RAN route that contains the recorded announcement (RAN) that you want played to the caller.

  Replace this parameter with a variable of type “RAN,” or with the route number.

Restriction

Do not insert the Give Busy or Give Overflow command immediately after the Give RAN command.

Automatic ringback

If Give RAN is the first treatment in the script, and there is a delay before the RAN is available, then the caller hears ringback until the announcement plays.

Example

In the following example, calls entering the call center between 5:00 p.m. and 8:00 a.m. hear a recorded announcement telling them that the call center is closed.

```
IF (TIME OF DAY = closed_hours_gv) THEN
   GIVE RAN closed_ran_gv
   DISCONNECT
END IF
```

Give Ringback

Introduction

Use the Give Ringback command to provide a ringback tone to a call (that is, to play a ringing sound to the caller).

You can use this command with the Wait command if you want to control the duration of the ringback tone. Use this command before a call is queued against one or more destinations (that is, to skillsets or agents). If a call is queued by the script without a specific treatment being given first, then ringback is automatically applied by Symposium Call Center Server (while the call is queued). For more information about which commands give an automatic ringback tone, see “Default treatments” on page 49.

Note: When the call is presented to an agent, the ringback tone is automatically provided by the switch.

Script syntax

GIVE RINGBACK

Restriction

Do not insert the Give Busy or Give Overflow command immediately after the Give Ringback command.

Example

In the following example, the caller hears a ringback tone if the call is being transferred.

```
IF TRANSFERRED THEN

   GIVE RINGBACK

   WAIT 6

   /* Allows time for the transferring party to complete
```

```
     the transfer */
  END IF
```

Give Silence

Introduction

Use the Give Silence command to provide silence to a call. Give Silence is generally used to turn off either music or ringback. If a call is given music followed by a RAN, the music automatically resumes after the RAN or treatment is completed. Therefore, if you want silence after a RAN, use the Give Silence command.

This command cannot be the first treatment given to a call. Therefore, it is not allowed as the first command in a script.

ATTENTION

Give Silence could be the first treatment for a call even if it is not the first command in the script. This can happen, for example, if an If-Then-Else or Where-Equals command leads to a Give Silence. If this is the case, you receive an error, and the call is rejected and routed to the default ACD-DN of the CDN as described in “First command rule” on page 41.

Script syntax

GIVE SILENCE

Example

In the following example, calls are queued to the service skillset. The caller hears music for 30 seconds, and then a recorded announcement stating that the agents are busy. The Give Silence command prevents the music from resuming after the recorded announcement.

```
QUEUE TO SKILLSET service_sk

GIVE MUSIC pop_music_gv

WAIT 30

GIVE SILENCE
```

```
        GIVE RAN agents_busy_ran_gv
```

Queue To Agent

Introduction

Use the Queue To Agent command to deliver a call to a n agent. This command provides agent-based routing capability.

Script syntax: By agent ID

QUEUE TO AGENT **[<agent_id> | <agent_id_list>]** {WITH PRIORITY **<priority>}**

Script syntax: Using longest idle agent

QUEUE TO AGENT LONGEST IDLE AGENT [**<skillset> | <skillset_list>]** {WITH PRIORITY **<priority>}**

Note: Only use Longest Idle Agent if you are sure that there is always an idle agent available. If there are no idle agents available, Symposium Call Center Server does not execute the Queue To Agent command.

Optional

The With Priority segment is optional in this command.

Parameters

Enter information for the following parameters:

- **<agent_id> or <agent_id_list>** The ID of the agent, or the list of agent IDs, to which you want the call queued.
- **<priority>** The priority with which you want the call queued to the specified agent.

- **<skillset> or <skillset_list>** The skillset, or list of skillsets, from which the longest idle agent is taken.

Restriction

- Do not insert the Give Busy or Give Overflow command immediately after the Queue To Agent command.
- Use a Wait command with at least two seconds after a Queue To Agent command.

Longest Idle Agent intrinsic

If you include the Longest Idle Agent intrinsic, the agent who has been idle the longest from all of the skillsets listed is presented with the call. The criteria used to determine which agent is idle the longest is defined for all skillsets. The definition is either the amount of idle time since the last call taken by the agent, or the total amount of idle time since the agent logged on.

With Priority option

You can specify the priority with which the call should be queued. You can assign a priority of 1 to 6, with 1 being the highest priority and 6 being the lowest. A call is assigned a priority of 6 by default. Calls with higher priorities are presented to the agents before calls with lower priorities.

Example 1

In the following example, a variable named “french_callers_gv” contains a list of CLIDs that identify some of BestAir’s French-speaking clients. When a call from a French-speaking client enters the call center, it is first queued to an agent (represented by the agent variable “agent_fr_5”). If the call is not answered, it is then queued to an agent (represented by the agent variable “agent_fr_8”) with a priority of 1. If the call is not answered within 4 seconds, it is queued to a skillset named “backup_french_sk” with a priority of 5.

```
IF (CLID = french_callers_gv) THEN

   QUEUE TO AGENT agent_fr_5, agent_fr_8 WITH PRIORITY 1

   WAIT 4
   QUEUE TO SKILLSET backup_french_sk WITH PRIORITY 5
   WAIT 2
END IF
```

Example 2

In the following example, if the number of idle agents in both the sales skillset and the service skillset is greater than zero, the call is queued to the agent who has been idle for the longest amount of time in either the sales skillset or the service skillset.

```
IF (IDLE AGENT COUNT sales_sk > 0)
AND (IDLE AGENT COUNT service_sk > 0) THEN
   QUEUE TO AGENT LONGEST IDLE AGENT sales_sk, service_sk
   WAIT 2
END IF
```

Queue To Skillset

Introduction

Use the Queue To Skillset command to queue calls locally at the Symposium Call Center Server site. The calls can be queued according to how many agents are logged on to the skillsets or by the length of time the agents have been idle at the skillsets.

Script syntax

QUEUE TO SKILLSET [**<skillset> | <list_of_skillsets>] {**WITH PRIORITY **<priority>}**

Script syntax: Most logged agents

QUEUE TO SKILLSET MOST LOGGED AGENTS [**<skillset> | <list_of_skillsets>] {**WITH PRIORITY **<priority>}**

Script syntax: By longest idle agent

QUEUE TO SKILLSET [**<skillset> | <list_of_skillsets>]** BY LONGEST IDLE AGENT {WITH PRIORITY **<priority>}**

Optional

The With Priority segment is optional in this command.

Parameters

Enter information for the following parameters:

- **<skillset>** or **<list_of_skillsets>** The skillset, or list of skillsets, to which you want the call queued.

- **<priority>** The priority with which you want the call queued to the specified skillset.

Restriction

- Do not insert the Give Busy or Give Overflow command immediately after the Queue To Skillset command.
- Use a Wait command with at least two seconds after a Queue To Skillset command.

Skillset lists

You can queue calls to up to 20 local skillsets simultaneously. You can queue to more than one skillset either by specifying all of the skillsets to which you want calls queued in a single Queue To Skillset command, or by using a separate Queue To Skillset command for each skillset.

The Queue To Skillset command evaluates each skillset specified in a random order. For each skillset, if an agent is available, the call is presented to that agent. If no agents are available, the call is queued to that skillset.

Using Wait

Always include a Wait command with at least two seconds immediately after a Queue to Skillset command. This allows time for the call to be queued before the next command.

Most Logged Agents intrinsic

If you use the Most Logged Agents intrinsic, the skillset to which the call is queued is selected based on the number of agents that are logged on to the skillsets. The Queue To Skillset command finds the skillset among all of the skillsets listed that has the most agents logged on to it at the time. Therefore, the call is queued only to that one skillset, not to all of the skillsets in the skillset list.

By Longest Idle Agent option

If you use the By Longest Idle Agent option, and one or more agents are idle in the listed skillsets, the agent is selected based on the length of time he or she is idle.

Note: The agent priorities in each skillset are still considered, so only the front of each idle agent queue (for all the listed skillsets) is inspected for the longest idle agent. The agent priorities within each skillset are not overridden using this option.

The criteria used to determine which agent has been idle the longest is defined for all skillsets as one of the following:

- the amount of idle time since the last call the agent took
- the amount of idle time the agent has spent since logging on

If there are no idle agents, then the call is queued to all skillsets listed in the command.

If this option is not specified, then the agent at the front of the idle agent queue is presented with the call.

Note: You cannot use both the By Longest Idle Agent option and the Most Logged Agents intrinsic within the same Queue To Skillset command.

With Priority option

You can specify the priority with which the call should be queued. Assign a priority of 1 to 6, with 1 being the highest priority and 6 being the lowest. A call is assigned a priority of 6 by default. Calls with higher priorities are presented to the agents before the calls with lower priorities.

Example 1

In the following example, the script checks whether the number of idle agents in the sales skillset is greater than 0. If so, then the call is queued to the sales skillset.

```
IF (IDLE AGENT COUNT sales_sk > 0) THEN
```

```
   QUEUE TO SKILLSET sales_sk
   WAIT 2 /*Allow time for the call to be queued*/
END IF
```

Example 2

In the following example, the skillset with the most logged on agents (either the service skillset or the support skillset) is assigned to the variable named “skillset_cv.” The call is then queued to skillset_cv.

```
ASSIGN MOST LOGGED AGENTS service_sk, support_sk TO
skillset_cv
QUEUE TO SKILLSET skillset_cv
WAIT 2
```

Example 3

In the following example, the script checks whether the number of idle agents in the service skillset is greater than the number of idle agents in the support skillset. If so, then the call is queued to the service skillset. Otherwise, it is queued to the support skillset.

```
IF (IDLE AGENT COUNT service_sk > IDLE AGENT COUNT
support_sk) THEN
   QUEUE TO SKILLSET service_sk
ELSE
   QUEUE TO SKILLSET support_sk
END IF
WAIT 2 /*Allow time for the call to be queued*/
```

Remove From Agent

Introduction

Use the Remove From Agent command to remove a call from the specified agent. This command applies to local agents only. Use this command if a call has to be answered within a specific length of time. If the call is still queued after this amount of time has passed, the call can be retrieved and requeued.

You can use the Remove From Agent command only to remove a call from an agent queue that was queued using the Queue To Agent command. This command cannot be used to remove a call that was queued to an agent using to a Queue To Skillset command.

Script syntax

REMOVE FROM AGENT **[<agent_ID> | <agent_ID_list>]**

Parameter

Enter information for the following parameter:

- **<agent_ID>** or **<agent_ID_list>** The ID of the agent from which you want the call removed. This parameter can be replaced with a single agent ID or a list of agent IDs.

Example

A call is queued to agent variable “agent_2” with a priority of 2 when it first arrives. If, after 60 seconds, the call has not been answered, it is removed from the agent and is queued to the sales skillset.

```
QUEUE TO agent_2 WITH PRIORITY 2

WAIT 2

SECTION Check_Age

   WAIT 20
```

```
    GIVE RAN agent_busy_ran_gv
    IF (AGE OF CALL > 60) THEN
        REMOVE FROM agent_2
        WAIT 2 /* avoid race condition */
        QUEUE TO SKILLSET sales_sk WITH PRIORITY 1
        WAIT 2
        EXECUTE WaitLoop
    END IF
EXECUTE Check_Age
```

Remove From Skillset

Introduction

Use the Remove From Skillset command to remove a queued call from the skillset or skillsets after queuing. This command applies only to local skillsets.

Script syntax

REMOVE FROM SKILLSET [**<skillset> | <list_of_skillsets>]**

Parameter

Enter information for the following parameter:

- **<skillset>** or **<list_of_skillsets>** The skillset, or list of skillsets, from which you want the call removed.

Example

A call is queued to the sales skillset. If, after 60 seconds, the call has not been answered, the script performs a test to determine which skillset has the most idle agents. If the sales skillset has more idle agents, then the call is removed from the sales skillset, and is queued to the service skillset.

```
QUEUE TO SKILLSET sales_sk

WAIT 2

GIVE RAN agents_busy_ran_gv

SECTION Check_Age

   WAIT 20

   IF (AGE OF CALL > 60) THEN

       IF (IDLE AGENT COUNT service_sk >= 1) THEN

             REMOVE FROM SKILLSET sales_sk

             QUEUE TO SKILLSET service_sk
             WAIT 2
             EXECUTE Continue_Requeued
       END IF
   END IF
GIVE RAN agents_busy_ran_gv
EXECUTE Check_Age
SECTION Continue_Requeued
   ...
```

Route Call

Introduction

Use the Route Call command to deliver a call to a destination specified by the parameter.

Script syntax

ROUTE CALL [**<dn> |** DEFAULT DN]

Parameter

Enter information for the following parameter:

- **<dn>** The directory number (DN) to which you want the call routed.

  You can replace this parameter with any dialable number, internal or external, such as an ACD-DN or a personal DN.

Restriction

Do not insert any commands after the Route Call command.

Default DN

If you specify the default DN, the call is sent to the default ACD-DN configured on the switch for the CDN from which the call entered the system.

Example 1

In the following example, if it is a holiday or a weekend, or outside of business hours, then the call is routed to another site in the call center:

```
IF (DATE = holiday_gv)

OR (DAY OF WEEK = weekend_gv)
OR (TIME OF DAY = after_hours_gv) THEN
   ROUTE CALL other_site
END IF
```

Chapter 7

Advanced script commands

In this chapter

Overview

Introduction

This chapter describes the advanced script commands that you can use if you have purchased the Nortel Networks Symposium Call Center Server 200 application software package. It shows how to write the commands in your scripts (script syntax), along with the parameters and optional segments that make up the commands.

Event Handler

Introduction

Use the Event Handler to give the script instructions to manage certain conditions that can occur during a call, such as the call being abandoned or a RAN response fail.

About the Event Handler

The Event Handler can manage two events: Call Abandon and RAN Response Fail.

If, during normal script execution, an unsolicited (unexpected) event or a failed treatment response occurs, the Event Handler can be used to execute statements to handle that event. These statements are placed in the Event Handler at the beginning of the script, and are not executed unless the specific event occurs. If the event occurs, and this event is defined in the Event Handler of this script, the script execution jumps to the Event Handler and executes the statement for that event.

In the case of unsolicited events, the script execution does not return to the body of the script after completing the event processing within the Event Handler. In the case of failed treatment responses, the script execution normally returns to the next statement in the main body of the script after the treatment command that failed, unless you use the Execute Script or Execute command. Execute Script branches to another script, and Execute branches to a section within the same script.

Notes:

1. The Event Handler is optional, and only applies to the script in which it resides. Also, if it is present, there is no need to list all events. Only use the events required for your system.
2. The script continues to monitor the call after it has been answered by an agent. If the agent puts the call on hold, for example, the Event Handler can play music to the caller.

First command

The Event Handler, if included in a script, must be the first command in the script.

Script syntax

```
EVENT HANDLER

    EVENT <event_a>: <statements>

    EVENT <event_b>: <statements>

END HANDLER
```

Parameters

Enter information for the following parameters:

- **<event_a>, <event_b>** The events that you want the Event Handler to address.
- **<statements>** The action that you want the script to take if the event occurs.

Tip: You can use the Event Handler to override the default music route. For example, you might not want to play advertisements to callers who are waiting for an agent to become available. However, you can use the Event Handler to play advertisements to callers who are put on hold after their call has been answered by an agent.

Failed responses

Use the following failed responses to replace the event parameters in the Event Handler:

- RAN Response Fail

Once the failed response is addressed, the execution of the script leaves the Event Handler and returns to its original place in the script. It executes the command immediately after the treatment command that failed.

Alternatively, you can execute another script when a failed response occurs. Also, you can execute another section within the same script rather than returning to the original place in the script.

Note: If you use the Event Handler to play an announcement or message in the event of a response failure, use as general a message as possible. The Event Handler cannot distinguish which RAN failed, so the message you play to callers in the event of a failure should not be intended to replace a specific message. If you need to determine which commands failed, it is possible to use a call variable. Set the variable to a specific value before each treatment command, and then test this value in the Event Handler to determine which commands failed.

Restrictions for unsolicited events

Use the following commands only with the Call Abandon event:

- IF-THEN-END-IF
- IF-THEN-ELSE-END-IF
- QUIT
- WAIT
- LOG
- SEND INFO

Restrictions for failed responses

You can use all commands with the RAN Response Fail event, except for the Section command.

Note: Once the failed response is addressed, the execution of the script leaves the Event Handler and returns to its original place in the script. The command immediately after the failed treatment command is executed.

Event Handler and secondary scripts

The Event Handler applies only to the script in which it appears. If you want Event Handler conditions to apply to the call after it has been sent to a referenced script, you must repeat the Event Handler command at the beginning of each script. Alternatively, you can change the Event Handler conditions in each script, or you can choose to not include it in referenced scripts.

Example

RAN Response Fail, in the following example, is defined in the Event Handler. If the script receives a Ran Response Fail, then the event is logged in the Script Log file. The call then leaves the Event Handler and returns to the script.

```
EVENT HANDLER

       EVENT RAN RESPONSE FAIL: LOG “first RAN failed”

END HANDLER

QUEUE TO SKILLSET skillset_sk

WAIT 3

GIVE RAN please_wait_ran_gv
```

How Event Handler avoids infinite loops

To avoid infinite loops in scripts, the Event Handler does not repeat treatments that prompt the Event Handler response. Consider the following example:

```
EVENT HANDLER

   EVENT RAN RESPONSE FAIL:GIVE RAN error_ran_gv

END HANDLER

GIVE RAN error_ran_gv

QUEUE TO SKILLSET customer_service_sk
```

If `error_ran_gv` is an invalid RAN route, then the first command in the script, GIVE RAN `error_ran_gv`, starts the Event Handler. However, the same invalid route is called for in the Event Handler. To avoid an infinite loop, Symposium Call Center Server does not repeat the treatment again. Instead, Symposium Call Center Server continues to process the script. In this example, the call would be queued to the skillset customer_service_sk.

Log

Introduction

Use the Log command to log a message to the Event Browser. For more information about the Event Browser, refer to the *Administrator's Guide*. This command is typically used for script testing only, as it can have an adverse effect on the system capacity if executed for every call in a high-traffic call center.

Note: By default, messages are logged to the Event Browser with the severity level of information.

ATTENTION

Do not use the Log command for normal call processing. Older events in the log file are overwritten by new events being logged. Events logged by this command can reduce the amount of historical data in the log file.

Nortel Networks recommends that you use the Log command only in the following situations:

- in test scripts to which only a few calls are made
- to identify errors. For example, you could define a boolean in your script that, by default, is set to False. To troubleshoot, set the boolean to True. The following is an example of the Log command:

```
IF debug THEN
    LOG "message"
END IF
```

Script syntax

LOG **<"msg_character_string">**

Parameter

Enter information for the following parameter:

- **<“msg_character_string”>** The text of the message entry as you want it to appear in the Script Log file.

Note: You must use quotation marks around the text of the message entry, or use a variable of type “string” without quotation marks.

Limitations

The log command is shown as an event in the Event Browser for up to ten times. After this, the event is no longer visible due to the “throttling” feature that prevents reoccurring events from filling up the Event Browser log. The throttling feature is reset every 30 minutes, after which the event is visible again. For more information about the event throttling utility, refer to the *Software Installation and Upgrade Guide.*

Restriction

Do not insert the Give Busy or Give Overflow command immediately after the Log command.

Example

In the following example, if the script receives a RAN response fail, then the event is logged in the Event Browser log.

```
EVENT HANDLER
       EVENT RAN RESPONSE FAIL: LOG “RAN failed in Master”
END HANDLER
```

Where-Equals

Introduction

Use the Where-Equals command to test for a condition that can have more than one expected result. You can use this command, for example, to replace several If-Then-Else commands that all check for the same condition. The parameter value_expr (the condition for which you are testing) is evaluated first, then different statements are executed, depending on the value of the parameter. The result might be to advance to another command, execute another script, or terminate the execution of the script. Use the Default clause, which is optional, to specify a command to use in the event that none of the values are true.

After finishing the statements in the applicable Value clause, the script leaves the Where-Equals command and continues executing the next command after the End Where command (unless the statements transfer the call to another script).

Script syntax

```
WHERE <value_expr> EQUALS

    VALUE <value_1>: <statements>

    VALUE <value_2>: < statements>

    VALUE <value_3>: <statements>

    DEFAULT: <statements>

END WHERE
```

Parameters

Enter information for the following parameters:

<value_expr> The expression that you want the script to evaluate. You can replace this parameter with an item, an intrinsic (such as CDN, DNIS, or CLID), or a formula. If you use a variable within this parameter, it must be of type "item."

<value_1>, <value_2>, <value_3> The values for the expression that you want the script to address. Replace these parameters with an item, a list, or a range.

<statements> The action that you want the script to take if the expression is equal to the specified value.

Limitations

Note: The number of value clauses is unlimited, however, Nortel Networks recommends that you do not use more than 50 value clauses, and that you break value clauses into groups wherever possible. Otherwise, script execution problems can occur.

Example

The following example demonstrates how to break value clauses into groups:

```
IF (NPA = west_coast_gv) THEN
   WHERE CLID EQUALS
       VALUE wc_dist1: EXECUTE SCRIPT WCoast1_Sales
       VALUE wc_dist2: EXECUTE SCRIPT WCoast2_Sales
       VALUE wc_dist3: EXECUTE SCRIPT WCoast3_Sales
       ...
   END WHERE
ELSE
   IF (NPA = east_coast_gv) THEN
       WHERE CLID EQUALS
             VALUE ec_dist1: EXECUTE SCRIPT ECoast1_Sales
```

```
                VALUE ec_dist2: EXECUTE SCRIPT ECoast2_Sales
                VALUE ec_dist3: EXECUTE SCRIPT ECoast3_Sales
                ...
            END WHERE
        ELSE
            ROUTE CALL DEFAULT DN
        END IF
    END IF
```

Using variables

If you use a variable name for the <value_expr> parameter in Where-Equals, the class of the variable must be Item. However, if you use a variable name for the <value _1/2/3> parameter in Value statements, the class of the variable can be either Item or Set. Nortel Networks recommends that you replace constant ranges of values (or comma separated lists) with Set class variables, as they are easier to modify. For example, to represent winter months, you can use the following value statement:

```
WHERE MONTH OF YEAR EQUALS
   VALUE Jan .. Mar: GIVE RAN winter_months_gv
   ...
```

However, Nortel Networks recommends that you use a Set class variable instead:

```
WHERE MONTH OF YEAR EQUALS
   VALUE winter_gv: GIVE RAN winter_months_gv
   ...
```

where winter_gv is a variable of MONTH OF YEAR type and Set class that contains the value of January `..` March.

Each VALUE statement can have multiple variables and constants in a comma-separated list. For example,

```
   WHERE CDN EQUALS
```

```
        VALUE 2001, sales_cdn_gv, 5700 : GIVE RAN vip_ran_gv
        ...
```

where sales_cdn_gv is a variable of CDN type and Set class that contains multiple CDN values (ranges and lists).

Example 1

In the following example, callers hear different recorded announcements, depending on the month in which they call the call center.

```
WHERE MONTH OF YEAR EQUALS
   VALUE JANUARY: GIVE RAN winter_specials_ran_gv
   VALUE APRIL: GIVE RAN spring_specials_ran_gv
   VALUE JULY: GIVE RAN summer_specials_ran_gv
   VALUE SEPTEMBER: GIVE RAN fall_specials_ran_gv
   DEFAULT: GIVE RAN default_ran_gv
END WHERE
```

Example 2

In the following example, the script uses the Where-Equals statement to direct calls to the appropriate script, depending on the CDN of the call.

```
WHERE CDN EQUALS
   VALUE 5553500: EXECUTE SCRIPT Automotive
   VALUE 5553600: EXECUTE SCRIPT Electronics
   VALUE 5553700: EXECUTE SCRIPT Sporting_Goods
   VALUE 5553800: EXECUTE SCRIPT Kitchen_Ware
   DEFAULT:  GIVE RAN lost_ran_gv
END WHERE
```

Chapter 8

Host data exchange commands

In this chapter

Overview

Introduction

Symposium Call Center Server provides a set of Application Program Interface (API) functions that are accessible by a third-party application. Scripts can send data as well as query and receive responses from a third-party application using host connectivity commands.

Types of third-party applications that you can use include IVR systems or screen-pop applications.

Note: When using host data exchange commands, it is important to take traffic engineering into consideration. Every call accessing a script containing host data exchange commands is affected by

- application response time

  A slow third-party application response time can slow down all call processing in Symposium Call Center Server.
- application design

  Databases might need special indexing and optimization to handle a high number of requests at the same time. For example, if the application can only handle one request at a time, and each request takes 4 seconds to process, a maximum of 900 calls per hour can be handled by the script.
- LAN interference

  This can include backup and restore procedures, and large file transfers.

Call ID

Every call that enters the Symposium Call Center Server has a call ID associated with it. The call ID is sent automatically with every Send Info or Send Request command. You, or your company’s application developer, can use the call ID to show information about the call (such as skillset or caller entered data) in a screen pop on the agents’ desktops. For more information, refer to the *Host Data Exchange API Programmer’s Guide*.

Example

The following example shows how you can use host connectivity in your scripts. This example uses the call data intrinsic. The call center that uses this script has to have a third-party IVR system.

Before the call reaches this script, an IVR session asks the caller to provide an account number and a numerical password. The account number is stored as a call intrinsic called Call Data 1. The password is stored as Call Data 2.

This script first checks the DNIS of the call. If the caller dials one of the 1-800 numbers listed, then the appropriate section of the script executes, and the call is queued to a skillset.

If the caller does not dial one of the 1-800 numbers, the section of the script called Call_Enter_Data_Section executes. The following events take place:

- A call variable named account_number_cv is assigned the value of the call intrinsic Call Data 1.
- A call variable named password_cv is assigned the value of the call intrinsic Call Data 2.
- The script sends the account number and password to the third-party application.
- The third-party application returns information to the script about the last agent that the caller spoke to.
- If that agent was John Kim (john_kim is a variable), then the call is queued to him.
- If that agent was anyone else, then the call is queued to the default skillset.

```
WHERE DNIS EQUALS

   VALUE 8005554103: EXECUTE Business_Travel_Section

   VALUE 8005554203: EXECUTE Vacations_Section

   DEFAULT: EXECUTE Call_Enter_Data_Section

END WHERE

SECTION Business_Travel_Section

   QUEUE TO SKILLSET business_travel_sk

   WAIT 2
   GIVE MUSIC pop_music_gv
   QUIT
SECTION Vacations_Section
   QUEUE TO SKILLSET vacations_sk
   WAIT 2
   GIVE MUSIC pop_music_gv
   QUIT
SECTION Call_Enter_Data_Section
   ASSIGN TO CALL DATA 1 TO account_number_cv
   ASSIGN TO CALL DATA 2 TO password_cv
   SEND REQUEST app_ID account_number_cv, password_cv
   GET RESPONSE app_ID last_agent_talked_to_cv
   WHERE last_agent_talked_to_cv EQUALS
       VALUE 10: ASSIGN john_kim TO pref_agent_cv
       VALUE 11: ASSIGN sue_sim TO pref_agent_cv
       VALUE 12: ASSIGN lou_jones TO pref_agent_cv
       DEFAULT: QUEUE TO SKILLSET default_sk
                WAIT 2
                EXECUTE Message_Section
   END WHERE
   QUEUE TO AGENT pref_agent_cv
   WAIT 2
   SECTION Message_Section
       GIVE RAN agent_busy_ran_gv
       GIVE MUSIC soft_music_gv
   SECTION Wait_Loop
```

Variable types

You can use the following types of variables with host connectivity commands:

- integer
- string
- DN
- CLID
- DNIS

Use a call variable to store collected information. You can use a call or global variable for the application ID.

Agent ID and skillset variables

You cannot use agent ID or skillset variables with host connectivity commands. If you need to send or receive skillset or agent ID information, use another data type (for example, integer, string, or DN), and then use the Where Equals command to translate the information in the script.

Example

The following section of script sends a caller's DNIS number to a third-party application and then receives the preferred skillset for that customer. The first line of the script assigns an integer-type variable named "skillset_Integer" a default value of zero. Then the script sends the customer's DNIS number to the third-party application. The third-party application uses this information to retrieve the preferred skillset for the customer. The script receives the preferred skillset information, which is represented by an integer. The Where Equals command determines the integer value and assigns a skillset value to a skillset variable.

For example, if the third-party application sends the value 2 as the preferred skillset for the customer, the script assigns the variable "skillset_var" to the skillset "general_sales."

```
ASSIGN 0 TO skillset_Integer_cv

SEND REQUEST app_ID DNIS

GET RESPONSE app_ID skillset_Integer_cv
```

```
WHERE skillset_Integer_cv EQUALS
   VALUE 1: ASSIGN customer_svc TO skillset_cv
   VALUE 2: ASSIGN general_sales TO skillset_cv
   VALUE 3: ASSIGN vip_sales TO skillset_cv
   DEFAULT: ASSIGN default_skillset TO skillset_cv
END WHERE
QUEUE TO SKILLSET skillset_cv
WAIT 2
```

Intrinsic types

You can use the following types of intrinsics with host connectivity commands:

- CDN
- CLID
- Dialed DN
- DNIS
- CALL DATA

Using an optional third-party IVR system

Introduction

You can use a third-party Integrated Voice Response (IVR) system to collect information from callers. This system plays voice prompts to callers and collects their responses. After the calls receive IVR treatment, the Symposium Call Center Server resumes control of them. The IVR system delivers the responses for the calls to the server.

Scripts can then use the information entered by the caller to route the call. You can also use the Symposium Call Center Server's host connectivity commands to export the information to a third-party screen pop application.

Example 1

Karen Johnson calls BestAir Airlines to inquire about European vacation packages. Before her call is answered by an agent, she receives IVR treatment. She hears a prompt such as

"For information about business travel, press 1. For information about vacations, press 2."

Karen presses 2. This information is stored in a call data intrinsic, and is sent to Symposium Call Center Server where it might be used in a script to route and treat the call appropriately.

Call handling with IVR

If you are using an third-party IVR system, you can configure your system to handle calls as follows:

1. The incoming call enters the switch and is queued to a DN that is connected to the front-end IVR system.
2. When an IVR voice channel (configured on the switch as a phoneset) becomes available, the call is routed to the IVR system.

3. The caller receives IVR treatment, which typically involves playing voice prompts and collecting responses. The caller’s responses are collected into the Call Data intrinsic.
4. When the IVR session is finished, the IVR system transfers the call to a CDN that is monitored and controlled by the Symposium Call Center Server. It also routes the caller-entered data to the server over the CLAN.
5. Call processing continues as configured in the script for that CDN.

Scripting recommendation

Several seconds can pass between the start of the call being transferred from the IVR system to the CDN. The switch expects the call to receive some treatment from the Symposium Call Center Server within 4 seconds; otherwise, it gives the call default treatment. Nortel Networks recommends that you write your script to give the call an immediate treatment.

Example

The following section of a script shows how you can check whether a call was transferred from the IVR system. The script uses the intrinsic CLID and a variable (mail_ports) to check whether the call was transferred from the front-end IVR system. If it was, then the call is immediately given ringback, the treatment that prevents it from defaulting if the transfer takes longer than 4 seconds.

```
IF CONSULTED AND (CLID = mail_ports) THEN
   GIVE RINGBACK /* First treatment 8?
   WAIT 20 /* Wait for the transfer to complete */
   QUIT /* Should not reach here */
END IF
```

Send Info

Introduction

Use the Send Info command to send data to and start a third-party application, such as a screen-pop application. You must use the Send Request and Get Response commands to request and receive information.

Script syntax

SEND INFO **<application_ID> [<variable> | <list_of_variables>] [<call_intrinsic> | <list_of_call_intrinsics>]**

Note: There is no comma between the application ID and the first variable, but commas are mandatory between the variables and intrinsics (additional spaces are optional).

Parameters

Enter information for the following parameters:

- **<application_ID>** The application ID. Replace this parameter with an Integer type variable.
- **<variable>** or **<list_of_variables>** The variable, or list of variables, that contains the data that you want to send to the third-party application. List of variables are single variable values separated by commas. For a list of variable types that you can use with this command, see “Variable types” on page 191.
- **<call_intrinsic> or <list_of_call_intrinsics>** The call intrinsic, or list of call intrinsics, that you want to send to the third-party application. For a list of intrinsics that you can use with this command, see “Intrinsic types” on page 192.

Restriction

Do not insert the Give Busy or Give Overflow command immediately after the Send Info command.

Example

In the following example, any calls entering CDN number 5553750 initiate a request to a third-party application (such as a database) for the callers account balance. Once the account balance is returned, the call is queued to the skillset “customer_service_sk”. If a call enters any other CDN, the CDN number is send to the third-party application and the call is queued to skillset “general_information_sk”.

```
WHERE CDN EQUALS
   VALUE 5553750:
       SEND REQUEST app_ID account_balance_cv, CDN, CLID
       GET RESPONSE app_ID account_balance_cv
       QUEUE TO SKILLSET customer_service_sk
       WAIT 2
   DEFAULT:
       SEND INFO app_ID CDN
       QUEUE TO SKILLSET general_information_sk
       WAIT 2
END WHERE
GIVE RAN agents_busy_ran_gv
GIVE MUSIC soft_music_gv
```

Send Request

Introduction

Use the Send Request command to ask for specific data from a third-party application. Call and global variables are used in this command to identify the data that you are requesting from the host. These variables are read-only. That is, the host application cannot change their value.

Note: The Send Request command must always be followed by a Get Response command. Only comments can separate these commands in the script.

Script syntax

SEND REQUEST **<application_ID> [<variable> | <list_of_variables>] [<list_of_call_intrinsics>]**

Note: There is no comma between the application ID and the first variable, but commas are mandatory between the variables and intrinsics (additional spaces are optional).

Parameters

Enter information for the following parameters:

- **<application_ID>** The application ID. Replace this parameter with an Integer type variable.
- **<variable>** or **<list_of_variables>** The variable, or list of variables, that contains the data that you are requesting from the third-party application. For a list of variable types that you can use with this command, see “Variable types” on page 191.
- **<call_intrinsic> or <list_of_call_intrinsics>** The call intrinsic, or list of call intrinsics, that you want to send to the third-party application. For a list of intrinsics that you can use with this command, see “Intrinsic types” on page 192.

Restriction

Do not insert the Give Busy or Give Overflow command immediately after the Send Request command.

Example

The following section of script sends a caller’s CLID to a third-party application and then receives the preferred banker for that customer. The first line of the script assigns a default value of zero to the variable named “personal_banker_cv.” Then the Send Request command is used to send the customer’s CLID to the third-party application. The Get Response command is used to retrieve the preferred banker for the customer from the third-party application. The Where Equals command is used to replace the default value of zero (assigned initially to the variable “personal_banker_cv”) with the value retrieved from the third-party application.

If the value retrieved from the third-party application is returned within two seconds, the call is presented to the appropriate agent. Otherwise, the call is queued to the sales skillset.

Note: When you replace a specific agent with an agent variable, you do not have to modify any scripts if the agent leaves the call center and is replaced by another agent.

```
ASSIGN 0 TO personal_banker_cv

SEND REQUEST app_ID CLID

GET RESPONSE app_ID TIMER 2 personal_banker_cv

WHERE personal_banker_cv EQUALS

   VALUE 1000: ASSIGN 1000 TO banker_choice_cv

   VALUE 1111: ASSIGN 1111 TO banker_choice_cv

   VALUE 1112: ASSIGN 1112 TO banker_choice_cv

   DEFAULT:    QUEUE TO SKILLSET sales_sk

               EXECUTE Delay_Message

END WHERE

QUEUE TO AGENT banker_choice_cv
```

```
SECTION Delay_Message
   WAIT 2
   GIVE RAN agent_busy_ran_gv
   GIVE MUSIC soft_music_gv
```

Get Response

Introduction

Use the Get Response command to obtain the response from the Send Request command previously sent to a third-party application. Send Request must be the previous command in the script. Script validation fails if the command is not preceded by a Send Request command.

You can specify one or more call variables in which to store the data in the response message.

Note: All variables to be returned from a third-party application using the Get Response command must have their initial value set to a default value. The value returned replaces the default value.

Script syntax

GET RESPONSE **<application_ID>** {TIMER **<timer>} [<variable> | <list_of_variables>]**

Note: There is no comma between the application ID and the first variable, but commas are mandatory between the variables (additional spaces are optional).

Optional

The Timer segment is optional.

Parameters

Enter information for the following parameters:

- **<application_ID>** The application ID. Replace this parameter with an integer type variable.
- **<timer>** The maximum amount of time, in seconds, to wait for a response.
- **<variable>** or **<list_of_variables>** The call variable, or list of variables, in which you want to store the data received from the third-party application.

For a list of variable types that you can use with this command, see "Variable types" on page 191.

Note: Do not use global variables in the variable list—only call variables are allowed. However, you can use either global or call variables for the application_ID and the timer.

Timer option

You can specify the maximum amount of time to wait for a response from the host. If a response is not received in the amount of time specified, the command fails and is aborted.

Note: If you do not specify a timer, the default of ten seconds is used.

The only way to detect that the timer has expired in the script is to initialize the call variables with specific values (using the Assign To command), and test them after the Get Response to see if they have changed.

Example

The following section of script sends a caller's CLID to a third-party application, and then receives the preferred skillset for that customer. The first line of the script assigns a default value of zero to the variable named "pref_skillset_cv." Then the Send Request command is used to send the customer's CLID to the third-party application. The Get Response command is used to retrieve the preferred skillset for the customer from the third-party application. The Where Equals command is used to replace the default value of zero (assigned initially to the variable "pref_skillset_cv") with the value retrieved from the third-party application.

If the value retrieved from the third-party application is returned within ten seconds, the call is queued to the appropriate skillset.

```
ASSIGN 0 TO pref_skillset_cv

SEND REQUEST app_ID CLID

GET RESPONSE app_ID TIMER 10 pref_skillset_cv

WHERE pref_skillset_cv EQUALS

   VALUE 10: ASSIGN sales_sk TO skillset_cv
   VALUE 11: ASSIGN service_sk TO skillset_cv
   VALUE 12: ASSIGN support_sk TO skillset_cv
   DEFAULT:  ASSIGN service_sk TO skillset_cv
END WHERE
QUEUE TO SKILLSET skillset_cv
WAIT 2
GIVE RAN agent_busy_ran_gv
GIVE MUSIC soft_music_gv
```

Chapter 9

Intrinsics

In this chapter

Overview

Introduction

Intrinsics contain system-wide information about skillsets, agents, time, traffic, and call type. You can use intrinsics in your scripts to access system information. The script then uses this information in formulas and decision-making statements.

Symposium Call Center Server creates and maintains intrinsics automatically. Intrinsics are available only to query data about the system within scripts, not to modify data. Any script can use information from intrinsics throughout the system.

Types of intrinsics

There are four types of intrinsics described in this chapter:

Section A, “Skillset intrinsics,” describes the intrinsics that are based on information about skillsets or agents.

Section B, “Time intrinsics,” describes the intrinsics that are based on system time information including the time of day, day of week, and day of year.

Section C, “Traffic intrinsics,” describes the intrinsics that are based on the system traffic level information.

Section D, “Call intrinsics,” describes the intrinsics that are specific to each call (for example, the CLID).

Return value

The data that an intrinsic gathers from the system and inserts into the script is referred to as the “return value” of the intrinsic. For example, the following section of a script instructs the system to queue calls to the support skillset if the number of agents in the service skillset (Logged Agent Count) is less than five.

```
IF (LOGGED AGENT COUNT service_sk < 5) THEN
    QUEUE TO SKILLSET support_sk
END IF
```

If, at 2:00 p.m., three agents from the service skillset are logged on, then the return value for the intrinsic is 3. Therefore, incoming calls are queued to the support skillset. Suppose later, at 3:30 p.m., nine agents from the service skillset are logged on. Now, the return value for the intrinsic is 9, and calls are not queued to the support skillset.

Intrinsics returns no value

If an intrinsic cannot return a valid value, it might return no value at all. For example, in the following statement:

```
QUEUE TO AGENT LONGEST IDLE AGENT service_sk,
general_sales_sk
```

If all of the agents in the service_sk and general_sales_sk skillsets are busy on active calls or in Not Ready mode, then the intrinsic Longest Idle Agent does not return any value. As a result, Symposium Call Center Server cannot queue the call to an agent.

To prevent the call from being queued to the default skillset, check to ensure that the call is queued before the end of the script.

Example

In the following example, the script attempts to queue the call to the agent who has been idle the longest in either the service_sk or the general_sales_sk skillset. If, after 2 seconds, the call is not queued, it is queued to a third skillset named backup_skillset_sk.

```
QUEUE TO AGENT LONGEST IDLE AGENT service_sk,
general_sales_sk

WAIT 2

IF NOT QUEUED THEN
    QUEUE TO SKILLSET backup_skillset_sk
    WAIT 2
```

```
        END IF
```

Examples of intrinsics use

Introduction

This section provides examples of how you can use intrinsics in your scripts.

Decision making within a script based on skillset intrinsics

This script checks if the number of idle agents in the service skillset is greater than the number of idle agents in the support skillset. If the service skillset has more idle agents, the call is queued to the service skillset. Otherwise, the call is queued to the support skillset.

```
IF (IDLE AGENT COUNT service_sk > IDLE AGENT COUNT
support_sk) THEN

   QUEUE TO SKILLSET service_sk

   WAIT 2

ELSE

   QUEUE TO SKILLSET support_sk

   WAIT 2

END IF
```

Decision making within a script based on time intrinsics

This script first checks if the time of day is between 5:00 p.m. and 8:00 a.m., and that the day of the week is not Saturday or Sunday. If all of these conditions are true, then the Night_section statement is executed.

```
IF (TIME OF DAY = dinner_hour_gv)

AND(DAY OF WEEK < > SATURDAY, SUNDAY)THEN

   EXECUTE Night_Section

END IF
```

Decision making within a script based on traffic intrinsics

This script checks to see if the number of calls in the system is greater than 50. If it is greater, the caller hears a busy tone.

```
IF (TOTAL ACTIVE CALLS > 50) THEN
   GIVE BUSY
END IF
```

Decision making within a script based on call intrinsics

This script checks to see if the CLID of the call is 416-555-1212. If it is, the call is queued to the skillset gold_skills_sk. Otherwise, the call is queued to the skillset general_skills_sk.

```
IF (CLID = 4165551212) THEN
   QUEUE TO SKILLSET gold_skills_sk
   WAIT 2
ELSE
   QUEUE TO SKILLSET general_skills_sk
   WAIT 2
END IF
```

Section A: Skillset intrinsics

In this section

Overview of skillset intrinsics

Introduction

Skillset intrinsic elements are based on information about skillsets. The returned value from the intrinsic can then be used in queuing commands, conditional commands, and so on. Skillset intrinsics return skillsets, integer values, and agent IDs.

Using lists of skillsets and agents

You can enter either a single skillset or agent ID, or a list of skillsets or agent IDs, for many intrinsics. If you use lists, be sure to follow these rules:

- List entries must be separated with commas.
- Lists of skillsets cannot contain more than 20 skillsets.
- Lists of agent IDs cannot contain more than 20 agent IDs.
- A call can be queued to a total of 20 agents and skillsets. That is, you cannot queue a call to both 20 skillsets and 20 agents.

Default values for errors

If the intrinsic value cannot be calculated due to an error condition (for example, the skillset does not exist), a default value is returned so that script execution can continue. The following table shows the default value for skillset return types:

Return type	Default value
Integer	0
Boolean (True or False)	False
Skillset	Skillset ID = 0

Answered Call Count

Introduction

The Answered Call Count intrinsic is the total number of incoming calls that have been answered during the ten-minute interval (see “How data is collected” on this page).

If you specify a list of skillsets, then the sum of answered call counts for the skillsets is returned.

Script syntax

ANSWERED CALL COUNT [**<skillset>** | **<skillset_list>]**

Parameter

Enter information for the following parameter:

- **<skillset>** or **<skillset_list>** The skillset, or list of skillsets, that you want to include in the return value.

Return type

This intrinsic returns an integer to the script.

How data is collected

Data is collected from the system at ten-minute intervals for this intrinsic. This ensures that burst-rate call traffic is reflected in the result. It also ensures that steady call traffic is reflected, since steady traffic is the same for longer time periods as it is for the given ten-minute interval.

Example

In the following example, if the number of answered calls for the service skillset is less than that of the support skillset, then incoming calls are queued to the service skillset. Otherwise, calls are queued to the support skillset. If the call is not answered immediately, the caller hears a message indicating that all agents are busy, followed by music.

```
IF (ANSWERED CALL COUNT service_sk < ANSWERED CALL COUNT
support_sk) THEN
   QUEUE TO SKILLSET service_sk
ELSE
   QUEUE TO SKILLSET support_sk
END IF
WAIT 2
GIVE RAN agents_busy_ran_gv
GIVE MUSIC pop_music_gv
SECTION WaitLoop
```

Average Speed Answer

Introduction

The Average Speed Answer is the calculated average speed of call answering in the given priority coming into the system. This data is based on the same calculations as the real-time supervisor data displays using the real-time moving window time frame of ten minutes.

If you specify a list of skillsets, then the minimum Average Speed Answer for the list of skillsets is returned. If you omit the With Priority segment, then the return value includes calls of all priorities.

Script syntax

AVERAGE SPEED ANSWER [**<skillset>** | **<skillset_list>]** {WITH CALL PRIORITY **<priority>}**

Optional

The With Call Priority segment is optional.

Parameters

Enter information for the following parameters:

- **<skillset>** or **<skillset_list>** The skillset, or list of skillsets, that you want to include in the return value.
- **<priority>** The priority of the calls that you want to track.

Return type

This intrinsic returns a value, in seconds, to the script.

How data is collected

Data is collected from the system at ten-minute intervals for this intrinsic. This ensures that burst-rate call traffic is reflected in the results. It also ensures that steady call traffic is reflected, since steady traffic is the same for longer time periods as it is for the given ten-minute interval.

Example

In the following example, if calls are answered more quickly, on average, by the service skillset than by the support skillset, then incoming calls are queued to the service skillset. Otherwise, calls are queued to the support skillset. If the call is not answered immediately, the caller hears a message indicating that all agents are busy, followed by music.

```
IF (AVERAGE SPEED ANSWER service_sk < AVERAGE
SPEED ANSWER support_sk) THEN

   QUEUE TO SKILLSET service_sk

ELSE

   QUEUE TO SKILLSET support_sk

END IF

WAIT 2

GIVE RAN agents_busy_ran_gv

GIVE MUSIC pop_music_gv

SECTION WaitLoop
```

Expected Wait Time

Introduction

The Expected Wait Time is the predicted expected wait time of the current call in the given skillset at the moment the intrinsic is executed. This value can change over time, depending on call traffic. This intrinsic is calculated by the call processing executor using real-time data. If the call is not yet in the skillset indicated, an average expected wait time for the skillset is returned.

If you specify a skillset list, then the returned value is the minimum Expected Wait Time of all the skillsets.

The purpose of the Expected Wait Time intrinsic is to estimate, based on historical information, how long it can take for the current call to be answered by a particular skillset.

Tip: Use this intrinsic to play the expected wait time to callers only if the wait is unusually long for your call center. For example, if the normal wait time for your call center is two minutes, but a burst of traffic has increased the wait time to five minutes, use this intrinsic to warn callers of the long wait. Give the caller options at this point (for example, the choice of leaving a message or continuing to wait).

Script syntax

EXPECTED WAIT TIME **[<skillset> | <skillset_list>]**

Parameter

Enter information for the following parameter:

- **<skillset>** or **<skillset_list>** The skillset, or list of skillsets, that you want to include in the return value.

Return type

This intrinsic returns a value, in seconds, to the script.

Conditions that increase the wait time

The Expected Wait Time can become longer as a caller waits. This means that there is no guarantee that the caller does not wait longer than the announced wait time. This is due to any of the following conditions:

- Calls with different priorities are queued to any given skillset. Thus, new incoming calls of higher priority are inserted in the queue in front of calls of lower priority. A new, lower-priority caller might initially hear that his or her wait time is 40 seconds and, a minute later, can be advised that his or her wait time is now five minutes.
- A burst of traffic can come into the system (including a call that is of higher priority) after the first wait time is given, which increases the caller's wait time.
- Agents servicing the queue could log off (to have lunch, for example) and substantially change the expected wait time for the calls that came into the queue before they logged off. If the number of agents in the skillset queue is not used in the algorithm, this situation takes several sample periods to filter through to an adjusted expected wait time.

Nortel Networks recommends that you use only the Expected Wait Time intrinsic in single-priority systems to avoid these conditions.

ATTENTION

Because the expected wait time can increase while a caller waits, play the expected wait time to callers once.

Formulas

Default Expected Wait Time when not queued

Symposium Call Center Server provides a default system calculation for the Expected Wait Time (EWT) intrinsic. The Expected Wait Time is based on the Average Speed of Answer (ASA) of calls in the skillset and the current call load for the skillset. The call load factor is included because the ASA is calculated at a given answered call count and increases or decreases in proportion to the current queued call count.

EWT (skillset) = ASA * Call Load

where

Call Load = Queued Call Count (skillset) / Answered Call Count (skillset)

Default Expected Wait Time when queued

If the call is queued, the calculation is further refined to take into account the call's current position in the queue. This is to better estimate the time remaining in queue for the call. For this calculation, only queued call statistics are used.

EWT (call, skillset) = ASA Waiting Calls * Queued Call Load * Position in Queue Factor

where

ASA Waiting Calls = Total Call Time Delay (skillset) / Queued Answered Call Count (skillset)

Queued Call Load = Queued Call Count (skillset) / Queued Answered Call Count (skillset)

Position in Queue Factor = Position in Queue (call, skillset) /Queued Call Count (skillset)

Customized formulas

If you do not want to use the default formula to estimate Expected Wait Time, you can use intrinsics to build a customized formula. For example, create the following customized formula:

EWT = (Queued Call Count * Talk Time)/Logged Agent Count

with Talk Time being an estimated constant for the call center.

Example

In the following example, the call is first queued to the sales skillset. If the Expected Wait Time for the call is greater than 240 seconds, the caller hears a recorded announcement informing him or her that a long wait is expected. If the Expected Wait Time is greater than 60 seconds but less than 240 seconds, the recorded announcement indicates an average wait time. An Expected Wait Time less than 60 seconds uses a short wait announcement:

```
QUEUE TO SKILLSET sales_sk
WAIT 4
ASSIGN EXPECTED WAIT TIME sales_sk TO exp_wait_cv
IF (exp_wait_cv > 240) THEN
   GIVE RAN long_wait_ran_gv
ELSE
   IF (exp_wait_cv > 60) THEN
       GIVE RAN average_wait_ran_gv
   ELSE
       GIVE RAN short_wait_ran_gv
   END IF
END IF
```

Idle Agent

Introduction

Use this intrinsic to check whether the specified agent is currently idle. If the agent is idle, the value returned is True. Otherwise, the value returned is False.

Script syntax

IDLE AGENT **<agent_ID>**

Parameter

Enter information for the following parameter:

- **<agent_ID>** The login ID of the agent that you want to track.

Return type

This intrinsic returns a True or False value to the script.

Example

In the following example, if the agent represented by the agent variable “agent_4” is idle and available, the call goes to that agent. Otherwise, the call is queued to the sales skillset:

```
IF IDLE AGENT agent_4 THEN
   QUEUE TO AGENT agent_4
ELSE
   QUEUE TO SKILLSET sales_sk
   WAIT 2
END IF
```

Idle Agent Count

Introduction

The Idle Agent Count is the current number of idle agents in the given skillset list.

If you specify a list of skillsets, then the return value is the maximum Idle Agent Count of all the skillsets in the list.

Script syntax

IDLE AGENT COUNT **[<skillset> | <skillset_list>]**

Parameter

Enter information for the following parameter:

- **<skillset>** or **<skillset_list>** The skillset, or list of skillsets, that you want to include in the return value.

Return type

This intrinsic returns a numerical value to the script.

Example

The following sample script first queues the call to the sales skillset. If the call is not answered immediately, the script checks if there are more than two idle agents in the service skillset. If so, the call is queued to the service skillset. If the call is not answered immediately, the caller hears a message indicating that all agents are busy, followed by music:

```
QUEUE TO SKILLSET sales_sk

WAIT 2

IF (IDLE AGENT COUNT service_sk > 2) THEN
```

```
      QUEUE TO SKILLSET service_sk
      WAIT 2
   END IF
   GIVE RAN agents_busy_ran_gv
   GIVE MUSIC soft_music_gv
   SECTION WaitLoop
```

Logged Agent Count

Introduction

The Logged Agent Count is the number of currently logged-in agents in the skillset list.

If you specify a list of skillsets, then the return value is the maximum of the Logged Agent Count for the skillsets. This is not the actual number of logged agents in all of the skillsets.

Use this intrinsic in loops to ensure that there are agents logged in to a skillset while the call is waiting to be answered.

Note: Do not use the Logged Agent Count intrinsic to test whether a skillset is in service. Although there are agents logged into a skillset, it does not mean that the skillset is in service. (You can put a skillset into out of service or transition mode from the Skillset Properties window.) Instead, use the Out Of Service intrinsic to test whether a skillset is in service or not.

Script syntax

LOGGED AGENT COUNT **[<skillset> | <skillset_list>]**

Parameter

Enter information for the following parameter:

- **<skillset>** or **<skillset_list>** The skillset, or list of skillsets, that you want to include in the return value.

Return type

This intrinsic returns a numerical value to the script.

Example

The following example shows how to test if there are enough agents within a skillset to handle the current call volume. If there are not enough agents, the caller receives a busy tone.

```
IF (QUEUED CALL COUNT sales_sk) > (2 * LOGGED AGENT COUNT
sales_sk) THEN
   GIVE BUSY
END IF
QUEUE TO SKILLSET sales_sk
WAIT 2
```

Logged Out Agent

Introduction

The Logged Out Agent intrinsic tests whether agents are logged off.

Script syntax

LOGGED OUT AGENT **<agent ID | agent ID list>**

Parameter

Enter information for the following parameter:

- **<agent ID | agent ID list>** The ID, or list of IDs, of the agents you want to include in the return value.

Return type

This intrinsic returns a True or False value to the script. The value is True if all of the agents you are querying are logged off. If you use a list of agents for the parameter and one of the agents is logged on, then the return value is False.

Example

In the following example, the Logged Out Agent intrinsic is used to test if the preferred agent is available. If not, the call is queued to a backup skillset instead:

```
IF LOGGED OUT AGENT agent_4 THEN

   QUEUE TO SKILLSET backup_sk

   WAIT 2

ELSE

   QUEUE TO AGENT agent_4

   WAIT 2

END IF
```

Longest Idle Agent

Introduction

The Longest Idle Agent is the ID of the agent in the skillset list who has been idle the longest.

The longest idle agent is determined by either the idle time of the agent since the last call was disconnected, or the total idle time of the agent since logging on. The algorithm that is used is a system parameter set by the user in the Global Parameters screen.

If you specify a list of skillsets, then the return value is the Longest Idle Agent for all of the skillsets.

Note: If you queue the call to only one skillset, you do not need to use this intrinsic. The server automatically queues the call to the agent who has been idle the longest.

Script syntax

LONGEST IDLE AGENT **[<skillset> | <skillset_list>]**

Parameter

Enter information for the following parameter:

- **<skillset>** or **<skillset_list>** The skillset, or list of skillsets, that you want to include in the return value.

By longest idle agent option

If you use the By Longest Idle Agent option, and one or more agents are idle in the listed skillsets, the agent is selected based on the length of time he or she is idle.

Note: The agent priorities in each skillset are still considered, so only the front of each idle agent queue (for all the listed skillsets) is inspected for the longest idle agent. The agent priorities within each skillset are not overridden using this option.

Return type

This intrinsic returns an agent ID to the script.

Example

In the following example, the call is queued to the skillset with the longest idle agent. If the call is not answered immediately, the caller hears a message indicating that all agents are busy, followed by music:

```
QUEUE TO SKILLSET sales_sk, service_sk BY LONGEST IDLE
AGENT

WAIT 2

GIVE RAN agents_busy_ran_gv

GIVE MUSIC pop_music_gv

SECTION WaitLoop
```

Most Logged Agents

Introduction

The Most Logged Agents intrinsic is the skillset with the most logged on agents at the time the intrinsic is executed.

Script syntax

MOST LOGGED AGENTS **<skillset_list>**

Parameter

Enter information for the following parameter:

- **<skillset_list>** The list of skillsets that you want to include in the return value.

Return type

This intrinsic returns a skillset to the script.

Example

In the following example, the value of the skillset with the most logged on agents is assigned to a call variable named “skillset_cv.” The call is then queued to the skillset represented by this variable. If the call is not answered immediately, the caller hears a message indicating that all agents are busy, followed by music. After this, the Section named “WaitLoop” is executed. Every 30 seconds, this section checks whether the call has been answered and if there are agents available in the required skillset to answer the call:

```
/* Always assign the skillset to a skillset call variable
(skillset_cv) so that you can check where the call was
queued to */

ASSIGN MOST LOGGED AGENTS service_sk, support_sk TO
skillset_cv
```

```
QUEUE TO SKILLSET skillset_cv
WAIT 2
GIVE RAN agents_busy_ran_gv
GIVE MUSIC classical_music_gv
SECTION WaitLoop
   WAIT 30
   IF NOT QUEUED THEN
       IF OUT OF SERVICE skillset_cv THEN
             GIVE RAN sorry_closed_ran_gv
             DISCONNECT
       ELSE
             QUEUE TO SKILLSET skillset_cv
             WAIT 2
       END IF
   END IF
   GIVE RAN agents_still_busy_ran_gv
   EXECUTE WaitLoop
```

Oldest Call

Introduction

The Oldest Call is the age of the call (with the given call priority, if you choose to include it) that has waited longest in the given skillset queue at the time that the intrinsic is executed.

If you specify a list of skillsets, then the return value is the maximum value of the Oldest Call values for all of the skillsets.

Note: See “Age Of Call” on page 256 to compare.

Script syntax

OLDEST CALL **[<skillset> | <skillset_list>]** {WITH CALL PRIORITY **<priority>}**

Optional

The With Call Priority segment is optional.

Parameters

Enter information for the following parameters:

- **<skillset>** or **<skillset_list>** The skillset, or list of skillsets, that you want to include in the return value.
- **<priority>** The priority of the calls that you want to track.

Return type

This intrinsic returns a value, in seconds, to the script.

Example

In the following example, if the oldest call queued to the sales skillset is more than five minutes old, then the current call is queued to a backup skillset. This is to prevent further calls from being queued to the sales skillset. If the oldest call queued to the sales skillset is less than five minutes, the script continues to queue the current call to that skillset:

```
IF (OLDEST CALL sales_sk > 300) THEN

   QUEUE TO SKILLSET backup_sk

   WAIT 2

ELSE

   QUEUE TO SKILLSET sales_sk

   WAIT 2

END IF
```

Out of Service

Introduction

Use the Out of Service intrinsic to test whether skillsets are out of service. If you specify a list of skillsets, then the Out Of Service intrinsic returns a True value if all skillsets are out of service. Otherwise, the intrinsic returns a False value.

A skillset is out of service when

- it is placed into Out of Service mode in the Skillset Properties window (including transition mode)
- all agents have logged off the skillset
- all agents are in Standby mode in this skillset

Script syntax

OUT OF SERVICE **[<skillset> | <skillset_list>]**

Parameter

Enter information for the following parameter:

- **<skillset>** or **<skillset_list>** The skillset, or list of skillsets, that you want to include in the return value.

Return type

This intrinsic returns a True or False value to the script.

Example

In the following example, if the sales skillset is out of service, then incoming calls receive night call treatment. Otherwise, they are queued to the sales skillset. If the call is not answered immediately, the caller hears a message indicating that all agents are busy, followed by music:

```
IF OUT OF SERVICE sales_sk THEN
```

```
      EXECUTE Night_Section
   END IF
   QUEUE TO SKILLSET sales_sk
   WAIT 2
   GIVE RAN agents_busy_ran_gv
   GIVE MUSIC pop_music_gv
   SECTION WaitLoop
```

Position In Queue

Introduction

The Position In Queue is the position of the call in the skillset queue at the time the intrinsic is executed.

Note: The call must be queued to the skillset before this intrinsic is used.

If you specify a list of skillsets, then the return value is the minimum value of the Position In Queue values for all of the skillsets.

The position is calculated using the number of calls queued in front of the current call. This number includes all calls with priorities equal to or higher than the priority of the current call. For example, if the current call is priority 2, then all priority 2 and priority 1 calls are included in the calculation.

Notes:

1. Nortel Networks recommends that this intrinsic only be used for call centers that do not assign priority levels.
2. Nortel Networks recommends that you do not use this intrinsic to play a caller's position in queue to him or her. Even in single-priority systems, a caller's position in queue can become higher rather than lower.

Script syntax

POSITION IN QUEUE **[<skillset> | <skillset_list>]**

Parameters

Enter information for the following parameter:

- **<skillset>** or **<skillset_list>** The skillset, or list of skillsets, that you want to include in the return value.

Return type

This intrinsic returns a numerical value to the script.

Example

In the following example, the call has already been queued to the service skillset. This section of the script tests the call's current position in the skillset queue. If it is more than 5, the priority is raised to priority 1 (unless it is already priority 1).

```
IF (POSITION IN QUEUE service_sk > 5)
AND (PRIORITY IN QUEUE service_sk < > 1) THEN
   CHANGE PRIORITY IN SKILLSET service_sk TO PRIORITY 1
END IF
```

Priority In Queue

Introduction

The Priority In Queue is the priority of the current call in the queue. This value can range from 0–6, with 1 being the highest and 6 being the lowest priority. A value of 0 indicates the call is not currently in the queue.

Note: The call must be queued to the skillset before this intrinsic is used.

Script syntax

PRIORITY IN QUEUE **<skillset>**

Parameters

Enter information for the following parameter:

- **<skillset>** The skillset that you want to include in the return value.

Return type

This intrinsic returns an integer to the script.

Example

In the following example, the script first checks if the call is queued in the sales skillset. If this is true and there are agents available to answer the call in the service skillset, the call is queued to the service skillset. Otherwise, the caller receives a message indicating that the call center is closed:

```
IF (PRIORITY IN QUEUE sales_sk = 0) THEN

   IF NOT OUT OF SERVICE service_sk THEN

      QUEUE TO SKILLSET service_sk

      WAIT 2

   ELSE
```

```
            GIVE RAN sorry_closed_ran_gv
            DISCONNECT
        END IF
    END IF
```

Queued Call Count

Introduction

The Queued Call Count is the number of calls outstanding against the skillsets in the given call priority at the time the intrinsic is executed. If you specify a list of skillsets, then the return value is the maximum value of the Queued Call Count value for all skillsets.

Note: This count includes only calls that have not yet been serviced and does not include calls that are currently being presented to an agent.

Script syntax

QUEUED CALL COUNT **[<skillset> | <skillset_list>]** {WITH CALL PRIORITY **<priority>}**

Optional

The With Call Priority segment is optional.

Parameters

Enter information for the following parameters:

- **<skillset>** or **<skillset_list>** The skillset, or list of skillsets, that you want to include in the return value.
- **<priority>** The priority of the calls that you want to track.

Return type

This intrinsic returns a numerical value to the script.

Example

In the following example, if the number of calls queued to the sales skillset is greater than twice the number of agents logged on from the sales skillset, then the caller is given a busy tone. Otherwise, the call is queued to the sales skillset. If the call is not answered immediately, the caller hears a message indicating that all agents are busy, followed by music:

```
IF (QUEUED CALL COUNT sales_sk) > (2 * LOGGED
AGENT COUNT sales_sk) THEN

   GIVE BUSY

END IF

QUEUE TO SKILLSET sales_sk

WAIT 2

GIVE RAN agents_busy_ran_gv

GIVE MUSIC soft_music_gv

SECTION WaitLoop
```

Section B: Time intrinsics

In this section

Time of Day

Introduction

The value returned by this intrinsic is the current time of day. You can use this intrinsic to determine the exact time or whether the current time of day is in a specified range of time.

Script syntax

TIME OF DAY

Return type

This intrinsic returns a time to the script.

Valid range

This value can range from 00:00 to 23:59.

Note: Do not use >= or <= with ranges.

Format

hh:mm

Example 1: Time range

In the following example, calls coming in to the call center between 5:00 p.m. and 8:00 a.m. receive “night” call treatment. Otherwise, calls are queued to general skillset. If the call is not answered immediately, the caller hears a message indicating that all agents are busy, followed by music.

Note: You can use a variable to represent a specific period of time. For example, the variable named “closed_hours_gv” represents the hours from 5:00 p.m. to 8:00 a.m.

```
IF (TIME OF DAY = closed_hours_gv) THEN
   EXECUTE Night_Section
END IF
QUEUE TO SKILLSET general_sk
WAIT 2
GIVE RAN agents_busy_ran_gv
GIVE MUSIC soft_music_gv
SECTION WaitLoop
```

Example 2: Exact time

To be meaningful, time comparisons using an exact time should include a greater than operator (>) or a less than operator (<). For example, given the following statements:

```
IF (TIME OF DAY = 08:00)
IF (TIME OF DAY < 08:00)
```

the first expression is true for one minute; the second is true from midnight until 7:59 a.m.

Day of Week

Introduction

The value returned by this intrinsic is the current day of the week. You can use this intrinsic to determine whether the current day is a specific day or is in a list or range of days.

Script syntax

DAY OF WEEK

Return type

This intrinsic returns a day to the script.

Possible values

The possible values for this intrinsic are

- Monday
- Tuesday
- Wednesday
- Thursday
- Friday
- Saturday
- Sunday

You cannot use abbreviations for the days of the week.

Note: The days are ordered from Monday to Sunday, with Monday as the first day of the week and Sunday as the last. Therefore, the expression Monday < Sunday is true, and Friday < Tuesday is false. Ranges also “wrap around,” so the expression “Friday .. Tuesday” is valid, and is true on Friday, Saturday, Sunday, Monday, and Tuesday.

Example

In the following example, calls coming into the call center between 5:00 p.m. and 8:00 a.m. on Saturday and Sunday receive “closed” call treatment. Otherwise, calls are queued to general skillset. If the call is not answered immediately, the caller hears a message indicating that all agents are busy, followed by music.

Note: You can use a variable to represent a specific period of time. For example, the variable named “weekend_gv” represents the days from Saturday to Sunday, and a variable named “closed_hours_gv” represents the hours from 5:00 p.m. to 8:00 a.m.

```
IF (DAY OF WEEK = weekend_gv)

AND (TIME OF DAY = closed_hours_gv) THEN

   EXECUTE Closed_Section

END IF

QUEUE TO SKILLSET general_sk

WAIT 2

GIVE RAN agents_busy_ran_gv

GIVE MUSIC pop_music_gv

SECTION WaitLoop
```

Date

Introduction

The value returned by this intrinsic is the current date. You can use this intrinsic to determine whether the current date is an exact date or is in a range of dates.

Script syntax

DATE

Return type

This intrinsic returns a date to the script.

Valid range

The value can range from January 1 to December 31.

Format

The month can be specified either by using the first three characters of the month (for example, Apr) or by spelling out the entire month (for example, September). In addition, the month can appear before or after the day.

Example

The following example uses both exact dates and a range of dates. Calls coming in to the call center on May 31, July 4, and December 25 January to 1 receive holiday call treatment.

Note: You can use a variable to represent a specific period of time. For example, the variable named “christmas_holiday_gv” represents the days from December 25 to January 1.

```
IF (DATE = May 31, 4 July, christmas_holiday_gv) THEN
```

```
        EXECUTE Holiday_Section
    END IF
```

Day of Month

Introduction

The value returned by this intrinsic is the current day of the month. You can use this intrinsic to determine whether the current day is an exact day of the month (the 15th, for example) or is in a range of days (the 2nd through the 10th, for example).

Script syntax

DAY OF MONTH

Return type

This intrinsic returns a day (the day of the current month) to the script.

Valid range

The value can range from 1–31.

Format

Use a number or a list of numbers from 1–31.

Example 1: Range of days

BestAir Airlines has monthly specials on different flights. Callers calling BestAir between the 27th and the 31st of each month hear a recorded announcement that gives information about the upcoming month’s specials.

Note: You can use a variable to represent a specific period of time. For example, the variable named “monthly_special_gv” represents the days between the 27th and the 31st of each month.

```
IF (DAY OF MONTH = monthly_special_gv) THEN
```

```
   GIVE RAN specials_ran_gv
END IF
```

Example 2: Exact days

In the following example, on the 1st and 15th day of each month, a section of the script called “Pay_Day_Section” is executed:

```
IF (DAY OF MONTH = 1,15) THEN
   EXECUTE Pay_Day_Section
END IF
```

Month of Year

Introduction

The value returned by this intrinsic is the current month of the year. You can use this intrinsic to determine whether the current month is a specific month or is in a range of months.

Script syntax

MONTH OF YEAR

Return type

This intrinsic returns a month to the script.

Valid range

The value can range from January to December.

Note: The month can be specified either by using the first three characters of the month (for example, Apr) or by spelling out the entire month (for example, September).

Example 1: Range of months

In the following example, calls entering the call center between the months of December and February hear a recorded announcement describing winter sales.

Note: You can use a variable to represent a specific period of time. For example, the variable named “winter_months_specials_gv” represents the months of December to February.

```
IF (MONTH OF YEAR = winter_months_specials_gv) THEN
   GIVE RAN winter_specials_ran_gv
END IF
```

Example 2: Exact month

In the following example, callers hear a different recorded announcement depending on the month in which they call the call center. For example, in January, they hear a recorded announcement describing sales highlights for the month. If there is no holiday in a particular month, callers hear a default recorded announcement:

```
WHERE MONTH OF YEAR EQUALS
   VALUE January: GIVE RAN jan_highlights_ran_gv
   VALUE Oct: GIVE RAN oct_highlights_ran_gv
   VALUE November: GIVE RAN nov_highlights_ran_gv
   VALUE Dec: GIVE RAN dec_highlights_ran_gv
   DEFAULT: GIVE RAN default_ran_gv
END WHERE
```

Section C: Traffic intrinsics

In this section

Call Rate

Introduction

The value returned by this intrinsic is the number of calls that entered the system during the previous ten-minute period.

Script syntax

CALL RATE

Return type

This intrinsic returns a numerical value to the script.

Example

In the following example, when the call rate for the 10 minute period exceeds 50 calls, callers hear a recorded announcement that states that the call center is experiencing a high level of traffic:

```
IF (CALL RATE > 50) THEN

   GIVE RAN busy_ran_gv

   ...

END IF
```

Total Active Calls

Introduction

The value returned by this intrinsic is the total number of calls in the system at the moment the intrinsic is executed.

Active calls include all calls in the system that are currently queued, being presented to agents, or being answered by agents. This does not include abandoned, rejected, defaulted, or completed calls.

Script syntax

TOTAL ACTIVE CALLS

Return type

This intrinsic returns a numerical value to the script.

Example

In the following example, if the number of active calls exceeds 200, then callers receive “overflow” treatment. You can restrict the number of active calls to 200 to preserve shared phone lines used by other staff in your call center:

```
IF (TOTAL ACTIVE CALLS >= 200) THEN

   GIVE OVERFLOW

END IF
```

Section D: Call intrinsics

In this section

Age Of Call

Introduction

This intrinsic is the age of the call at the time the intrinsic is executed.

Script syntax

AGE OF CALL

Return type

This intrinsic returns a value, in seconds, to the script.

Example

The following example is a section of script that queues the call to the sales skillset, and then plays music. The section Check_Age checks the age of the call every 20 seconds, until the call is more than 2 minutes old. If the call is greater than two minutes old, it is queued to a backup skillset:

```
QUEUE TO SKILLSET sales_sk WITH PRIORITY 2

WAIT 2

GIVE MUSIC pop_music_gv

SECTION Check_Age

   WAIT 20

   IF (Age of Call > 120) THEN

       EXECUTE Requeue

   END IF

   EXECUTE Check_Age

SECTION Requeue

   QUEUE TO SKILLSET backup_sk WITH PRIORITY 1

   WAIT 2
```

Call Data

When to use

This intrinsic holds caller-entered data. You must have a third-party IVR system in place to use this intrinsic in your scripts.

Script syntax

CALL DATA **<data number> | <variable (1-10)>**

Parameters

Enter information for the following parameters:

- **<data number>** A number between 1 and 10. You can use up to ten Call Data intrinsics in your scripts.
- **<variable>** The name of the variable in which to store the caller entered data. The type for this variable must be digit. Lists of variables are single variables separated by commas.

Return type

This intrinsic returns a digit to the script.

Note: This intrinsic can only return digits to the script. It cannot return letters or symbols.

Example

In the following example, this portion of the script checks the value of the caller-entered data intrinsic, Call Data 1. The value of this intrinsic is determined by the information entered by the caller in an IVR session. If the caller pressed 1 on his or her keypad, then the call is queued to General Sales. If the caller pressed 2, then the call is queued to the Business Travel skillset, and so on. If the caller pressed any other number, or no number at all, the call is automatically queued to the skillset General Sales.

```
...
WHERE CALL DATA 1 EQUALS
   VALUE 1: QUEUE TO SKILLSET General_Sales
   VALUE 2: QUEUE TO SKILLSET Business_Travel
   VALUE 3: QUEUE TO SKILLSET Gold_Clients
   DEFAULT: QUEUE TO SKILLSET skillset_A
END WHERE
```

Call Forward

Introduction

This intrinsic indicates if the call has been forwarded (where the return value is True) or not (where the return value is False).

This intrinsic is useful, for example, if a caller dials an agent DN directly, but the agent is not available to take the call. If the call is forwarded to the call center, you can play an announcement that informs the caller that he or she has been forwarded.

Script syntax

CALL FORWARD

Return type

This intrinsic returns a True or False value to the script.

Example

In the following example, calls that have been forwarded are given a recorded announcement that tells the callers that they have been forwarded to the call center:

```
IF CALL FORWARD THEN

   GIVE RAN forward_to_call_center_ran_gv

END IF
```

Call Forward Busy

Introduction

This intrinsic indicates whether the call has been forwarded because the phoneset at the original destination was busy (where the return value is True) or not (where the return value is False).

Script syntax

CALL FORWARD BUSY

Return type

This intrinsic returns a True or False value to the script.

Example

In the following example, calls that have been forwarded due to a busy condition are given a recorded announcement that tells the callers that their calls have been forwarded:

```
IF CALL FORWARD BUSY THEN
   GIVE RAN busy_forwarded_ran_gv
END IF
```

Call Forward Do Not Disturb

Introduction

This intrinsic indicates whether the call has been forwarded due to a do not disturb condition at the destination (where the return value is True) or not (where the return value is False).

Script Syntax

CALL FORWARD DO NOT DISTURB

Return type

This intrinsic returns a True or False value to the script.

Example

In the following example, calls that have been forwarded due to a do not disturb condition at the destination are given a recorded announcement that tells the callers that their calls have been forwarded:

```
IF CALL FORWARD DO NOT DISTURB THEN
   GIVE RAN sleeping_ran_gv
END IF
```

Call Forward No Answer

Introduction

This intrinsic indicates whether the call has been forwarded because no one answered the phoneset at the original destination (where the return value is True) or not (where the return value is False).

Script syntax

CALL FORWARD NO ANSWER

Return type

This intrinsic returns a True or False value to the script.

Example

In the following example, calls that have been forwarded due to no answer at the destination are given a recorded announcement that tells the callers that their calls have been forwarded:

```
IF CALL FORWARD NO ANSWER THEN
   GIVE RAN forward_no_answer_ran_gv
END IF
```

CDN

Introduction

The value is the controlled directory number (CDN) of the current call.

Script syntax

CDN

Return type

A CDN is returned by this intrinsic.

Example

In the following example, the script executes either the GoldCard, Reg_Customers, or Promotions script, depending on the CDN of the call:

```
WHERE CDN EQUALS
   VALUE 4165551230: EXECUTE SCRIPT Gold_Customers
   VALUE 4165551231: EXECUTE SCRIPT Reg_Customers
   VALUE 4165551232: EXECUTE SCRIPT Promotions
END WHERE
```

CLID

Introduction

The value is the calling line identification (CLID) of the current call.

Script syntax

CLID

Return type

A CLID is returned by this intrinsic.

Example 1

In the following example, calls with CLIDs that are included in the vip_list_gv variable are sent to be handled in a special way:

```
IF (CLID = vip_list_gv) THEN
   EXECUTE Special_Handling
END IF
```

Wildcards and placeholders

The following types of free-format expression in CLID comparisons are allowed to provide flexible digit string evaluation:

- ***wildcard*** Use the @ symbol to represent a string of digits (zero or more).
- ***placeholder*** Use a question mark (?) to represent a single-digit position.

Wildcard and placeholder rules

The following rules apply to the use of wildcards and placeholders:

1. Wildcards and placeholders can be used only with the CLID intrinsic.
2. Only one wildcard is allowed in any one CLID string. For example, the string @345@ is not valid.
3. Use wildcards only at the beginning or the end of a string.
4. Wildcards and placeholders cannot be used in ranges of CLIDs. For example, 333@ .. 339@ is not allowed. However, you can use wildcards and placeholders in lists of CLIDs.
5. Wildcards and placeholders can be used only with equal (=) and not equal (< >) operators. Greater than and less than operations on wildcard expressions give ambiguous results. A validation error is generated for scripts that use this construct.
6. Only variables of the WILDCLID data type allow the wildcard and placeholder characters.
7. The statement

   CLID = @

   always returns a True value.
8. A string with placeholders only has a return value of True for CLIDs with the same number of digits as there are placeholders. For example, the statement

   CLID = ???

   is true for all CLID strings of three digits, and false for all other strings. This type of expression is useful if you want to screen out strings of a fixed length.
9. Both wildcards and placeholders can occur in a CLID digit string. All rules still apply to placement.

Example 2

The following example shows the use of a wildcard in a CLID digit string. Calls with a CLID of 3050–3059 are queued to the VIP skillset with a priority of 1 and hear a special recorded announcement:

```
IF (CLID = 305@) THEN
```

```
    QUEUE TO SKILLSET VIP_sk WITH PRIORITY 1
    WAIT 2
    GIVE RAN you_are_special_ran_gv
END IF
```

Consulted

When to use

The return value indicates if the call was transferred or conferenced (where the return value is true) or not (where the return value is false).

Script syntax

CONSULTED

Return type

This intrinsic returns a True or False value to the script.

Example

This script plays a recorded announcement to callers who have been transferred or conferenced.

```
IF CONSULTED THEN
   GIVE RAN 24 /* “Your call has been transferred” */
END IF
```

Dialed DN

Introduction

The value is the number that the caller originally dialed if the call is forwarded to the CDN using a switch feature. This applies only to forwarded calls.

Script syntax

DIALED DN

Return type

A DN is returned by this intrinsic.

Example

In the following example, calls are queued to the appropriate skillset depending on the dialed DN:

```
WHERE DIALED DN EQUALS
   VALUE 2512: QUEUE TO SKILLSET sales_sk WITH PRIORITY 1
   VALUE 2603: QUEUE TO SKILLSET sales_sk WITH PRIORITY 2
   VALUE 2776: QUEUE TO SKILLSET service_sk WITH PRIORITY 1
   DEFAULT: QUEUE TO SKILLSET service_sk WITH PRIORITY 2
END WHERE
```

Note: The Dialed DN intrinsic can handle the blind transfer of a call back into the call center by an agent. The dialed DN of the agent’s transfer call is preserved for the original call when the transfer is complete. This allows the caller to be redirected to a new treatment or skillset.

Direct Call

Introduction

The value indicates if the call is a direct call (where the return value is True) or not (where the return value is False). The Direct Call intrinsic should be used with phonesets that have the forwarding option.

Script syntax

DIRECT CALL

Return type

This intrinsic returns a True or False value to the script.

Example

In the following example, all calls that are not direct calls receive a recorded announcement:

```
IF NOT DIRECT CALL THEN

   GIVE RAN you_were_forwarded_ran_gv

END IF
```

DNIS

Introduction

The value is the first number that the caller dialed to enter the system.

This intrinsic is called the Dialed Number Identification Service (DNIS), and is specific to the trunk used. Not all trunks can be configured to support DNIS; therefore, the DNIS intrinsic is empty for any trunk that is not configured on the switch to provide DNIS.

Note: If you are using Virtual Network Services (VNS) trunks, the DNIS feature is not supported. This means that a DNIS number cannot be forwarded from one site to another in a VNS multisite call center. If you want to send DNIS numbers from one site to another, you must configure Meridian Customer Defined Networking (MCDN) between sites.

Script syntax

DNIS

Return type

A DNIS is returned by this intrinsic.

Example

In the following example, calls are queued to the appropriate skillset depending on the DNIS:

```
WHERE DNIS EQUALS

   VALUE 8005552512: QUEUE TO SKILLSET sales_sk WITH
PRIORITY 1

   VALUE 8005552603: QUEUE TO SKILLSET sales_sk WITH
PRIORITY 2

   VALUE 8005552776: QUEUE TO SKILLSET service_sk WITH
PRIORITY 1
```

```
     DEFAULT: QUEUE TO SKILLSET service_sk WITH PRIORITY 2
END WHERE
     DISCONNECT
```

Queued

Introduction

The value indicates whether the call is queued in any local skillsets or agent queues (where the return value is True) or not (where the return value is False).

Script syntax

QUEUED

Return type

This intrinsic returns a True or False value to the script.

Example

This script uses the Queued command to avoid an endless loop. It ensures that the initial Queue To Skillset command worked:

```
IF OUT OF SERVICE sales_sk THEN

   EXECUTE Help_Me_Now

END IF

QUEUE TO SKILLSET sales_sk WITH PRIORITY 3

WAIT 3

/* This section repeats a recorded announcement every 30
seconds. It also checks whether the call is queued before
playing the announcement. */

SECTION Play_2nd_RAN

   WAIT 30

   IF NOT QUEUED THEN

       EXECUTE Help_Me_Now

   END IF
```

```
          GIVE RAN agents_still_busy_ran_gv
          EXECUTE Play_2nd_RAN
       SECTION Help_Me_Now
          ...
```

Chapter 10

Script expressions

In this chapter

Overview

Introduction

This chapter describes the expressions that you can use in your scripts to test for different conditions. It describes Logical (for example, Not, And, Or), Mathematical (for example, multiplication, division, addition, and subtraction), and Comparison (for example, greater than or less than) expressions. This chapter also explains how to control the order in which operations are performed.

Logical expressions

NOT

The Not expression evaluates a conditional expression. It returns a True value to the script if the expression is false, and a False value if the expression is true.

AND

The And expression evaluates two or more comparative expressions. It returns a True value to the script if the two expressions are both true, and a False value if either expression is false.

OR

The Or expression evaluates two or more comparative expressions. It returns a True value to the script if either, or both, of the two expressions are true, and a False value if both expressions are false.

Example 1

BestAir’s Customer Service department is open Monday to Friday, from 8:00 a.m. to 6:00 p.m. After 6:00 p.m., and on Saturdays and Sundays, the department is closed.

```
IF (TIME OF DAY = business_hours_gv)

AND (DAY OF WEEK = weekdays_gv) THEN

   GIVE RAN open_ran_gv

ELSE

   GIVE RAN closed_ran_gv

   DISCONNECT

END IF
```

Example 1 table

The following table illustrates the conditions under which calls are given “open” call treatment in Example 1.

Expression	Is it a weekday?	Is it between 8:00 a.m. and 6:00 p.m.?	Give “open” service?
AND	yes	yes	yes
AND	yes	no	no
AND	no	no	no
AND	no	yes	no

Example 2

```
IF (TIME OF DAY > 18:00) OR (DAY OF WEEK = SATURDAY,
SUNDAY)THEN
   GIVE RAN closed_ran_gv
   DISCONNECT
ELSE
   GIVE RAN open_ran_gv
END IF
```

Example 2 table

The following table illustrates the conditions under which callers receive “closed” call treatment.

Expression	Is it after 6:00 p.m.?	Is it Saturday or Sunday?	Play “closed” recording?
OR	yes	yes	yes
OR	yes	no	yes
OR	no	no	no
OR	no	yes	yes

Example 3

On the 1st and 15th of every month, BestAir has a company-wide meeting that reduces the number of agents available to take sales calls from customers. Therefore, during these meetings, they play a recorded announcement that tells callers that there might be some delay in answering their call. However, they do not hold the meetings if the 1st or 15th falls on a weekend. Regular sales service is available on weekends. This example also shows how you can combine two expressions (in this case, And and Not).

```
IF (DAY OF MONTH = 1,15)

AND NOT(DAY OF WEEK = SATURDAY, SUNDAY) THEN

   GIVE RAN meeting_ran_gv

END IF
```

Example 3 table

The following table illustrates the conditions under which the recording is or is not played to a caller.

Expression	Is it the 1st or 15th day of the month?	Is it a Saturday or Sunday?	Give meeting recording?
AND NOT	yes	yes	no
AND NOT	yes	no	yes
AND NOT	no	no	no
AND NOT	no	yes	no

Mathematical expressions

Addition

Symbol

+

Description

The Addition expression (plus sign) adds two values of the same type. This expression can be used with numerical constants, variables, and expressions that return the data types Integer and Seconds.

Division

Symbol

/

Description

The Division expression (forward slash) divides the first value by the second value of the same type. This expression can be used with numerical constants, variables, and expressions that return the data types Integer and Seconds.

Note: If the result of the division is not an integer number, the value is truncated, not rounded. For example, 10.7 becomes 10.

Multiplication

Symbol

*

Description

The Multiplication expression (asterisk) multiplies two values of the same type. This expression can be used with numerical constants, variables, and expressions that return the data types Integer and Seconds.

Subtraction

Symbol

–

Description

The Subtraction expression (dash) subtracts the second value from the first value of the same type. It can be used with numerical constants, variables, and expressions that return the data types Integer and Seconds.

Relational expressions

Introduction

You can use comparison expressions to compare the 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 valid expression. 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 a no (False) answer.

Values that can only be True or False are referred to as Boolean values. The state of several situations at once (for example, it is later than 08:00, and there is at least one agent logged on to the skillset) can be evaluated by writing several comparison expressions and then joining them into a logical expression consisting of a series of true or false answers.

Limitations

Some comparison expressions can be used only with variables that return integers, seconds, and dates. For example, while you can have a numerical value greater than 312, you cannot have a skillset "greater than" Sales.

Equal to (=) and not equal to (< >) can be used for all types of data.

Greater Than (>), Less Than (<), Greater Than or Equal (> =), and Less Than or Equal (< =) can be used only with integers, seconds, and dates. These expressions cannot be used with skillsets, DNs, agent IDs, and so on.

Example 1: Incorrect

The following is an example of an invalid use of the Greater Than expression:

```
IF (LONGEST IDLE AGENT gold_card_sk > 1543) THEN
    ...
```

Example 2: Correct

The following is an example of a valid use of an expression:

```
IF (LONGEST IDLE AGENT gold_card_sk = 1543) THEN

    ...
```

Set variables and constants

If you are testing for a set variable or constant (for example, a list of values or a range of values), you can use only the = and < > expressions, and they must only be compared with an “item.” That is, you are testing whether the item has a value that is in the set.

Example

```
IF (CLID = 5130, 5139, 5200 .. 5300) THEN

    ...
```

This tests whether the CLID is any of these values. The left side of the comparison is an item (CLID intrinsic), and the right side is a set (in this example, it is a constant list, including a range, but it could also be a set variable).

Equal

Symbol

=

Description

The Equal expression compares two values of the same type to see if they are equal or if they are of the same set (list or range).

Not Equal

Symbol

< >

Description

The Not Equal expression compares two values of the same type to see if the first value is different from the second value, or if the value on the left is not in the set (list or range) of values on the right.

Greater Than

Symbol

>

Description

The Greater Than expression compares two values of the same type to see if the first value is greater than the second.

Less Than

Symbol

<

Description

The Less Than expression compares two values of the same type to see if the first value is less than the second value.

Greater Than or Equal

Symbol

> =

Description

The Greater Than or Equal expression compares two values of the same type to see if the first value is greater than or equal to the second value.

Less Than or Equal

Symbol

< =

Description

The Less Than or Equal expression compares two values of the same type to see if the first value is less than or equal to the second value.

Order of operations

Introduction

When evaluating conditional expressions, the expression with the highest precedence (importance) is evaluated first, then the one with the second highest precedence, and so on, down to the expression with the lowest precedence.

When expressions appear more than once, or when two expressions with equal importance appear in the same expression, they are evaluated from left to right.

Expressions in conditional expressions are evaluated in the following order:

- parentheses ( )
- multiplication * and division /
- addition + and subtraction –
- comparison expressions =, < >, >, <, > =, < =
- logical expressions Not, And, Or
- left to right

Note: The expression Not has precedence over the And and Or expressions.

If an expression contains parentheses, the partial expression within the parentheses is resolved first. Then multiplications are resolved, additions, subtractions, and so on.

Example

Compare the results of the following calculations that have the parentheses in different places:

2 * 3 + 2 * 3 = 12

2 * (3 + 2 * 3) = 18

(2 * 3 + 2) * 3 = 24

(2 * 3) + (2 * 3) = 12

Chapter 11

Applications

In this chapter

Overview

Introduction

This chapter describes the following topics:

- how to view the list of applications that are defined on the system
- how to view the thresholds assigned to an application
- how to change the threshold class assigned to an application

What is an application?

An application is a unique identifier of a Master or primary script. There are applications for the Master script and for every primary script that it references. Symposium Call Center Server assigns a numerical application ID to each script.

Application IDs are used to identify the script from which call information is collected for reporting purposes. Since a call can pass through many scripts, calls are tracked by the application ID of the primary script that they enter from the Master script.

Symposium Call Center Server collects information and reports for applications to give call center managers specific details about call types, callers, or conditions.

ATTENTION

The maximum number of applications you can configure for Symposium Call Center Server is 505 (including system scripts). If you try to activate more primary scripts than the maximum number of applications configured for your system, the activation request is denied. This prevents Symposium Call Center Server from tracking call information. You can resolve this by either reducing the number of primary scripts or by increasing the number of applications configured for your server.

Viewing and changing applications, thresholds, and classes

Introduction

Use the Applications window to view the list of applications defined on your system. From this window, you can also access the Application Properties page, where you can view the name and the thresholds that apply to an application and change the threshold class assigned to it.

What is a threshold class?

There are two types of threshold classes: display and pegging. A display threshold class specifies the fields that apply to an application that you use in a real-time display. First- and second-level thresholds are defined in the class. The field on your real-time display is highlighted if its value falls below the first-level threshold or above the second-level threshold.

Pegging thresholds indicate a level that, when surpassed, causes statistics to be logged.

Example

A Short Call pegging threshold is set at ten seconds. This means that if a caller hangs up within ten seconds of speaking to an agent, the call is logged as a short call by the system.

For example, a customer inquires about the price of a flight to New York. He speaks to a BestAir sales agent. The agent lists the price of $199.99. The customer thanks him and hangs up. This call lasted eight seconds, so it is pegged as a short call.

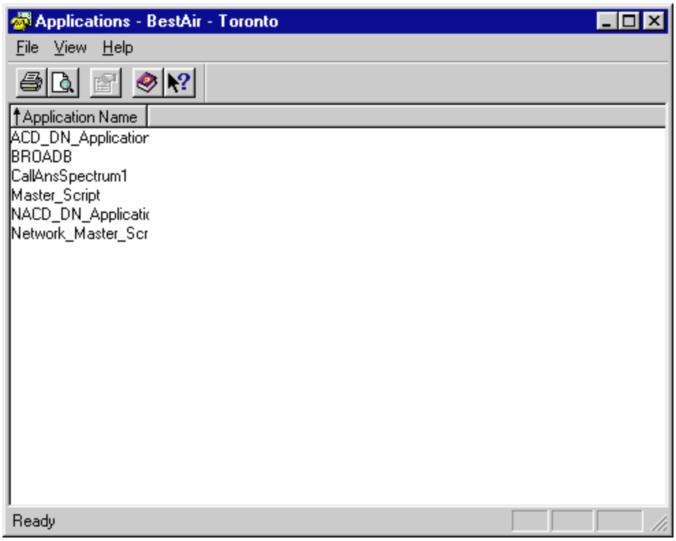
Where to change the thresholds defined in a threshold class

The threshold classes are changed in the Threshold Class window. You must have the proper privileges assigned to change a threshold class. For more information, refer to the *Administrator's Guide*.

To view applications

1 From the SMI window, choose Call Flow Administration → Applications.

Result: The Applications window appears. This window lists all of the applications defined on your system.

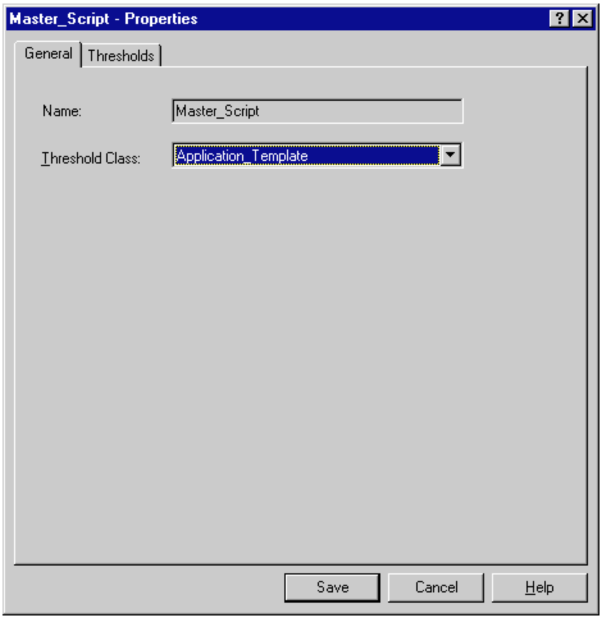


Application Name
ACD_DN_Applicatior
BROADB
CallAnsSpectrum1
Master_Script
NACD_DN_Applicatic
Network_Master_Scr

2 Select the application that you want to view.

3 Choose File → Properties.

Result: The Application Properties page appears displaying the application name and the threshold class.



Name: Master_Script

Threshold Class: Application_Template

To view application thresholds

1 From the SMI window, choose Call Flow Administration → Applications.

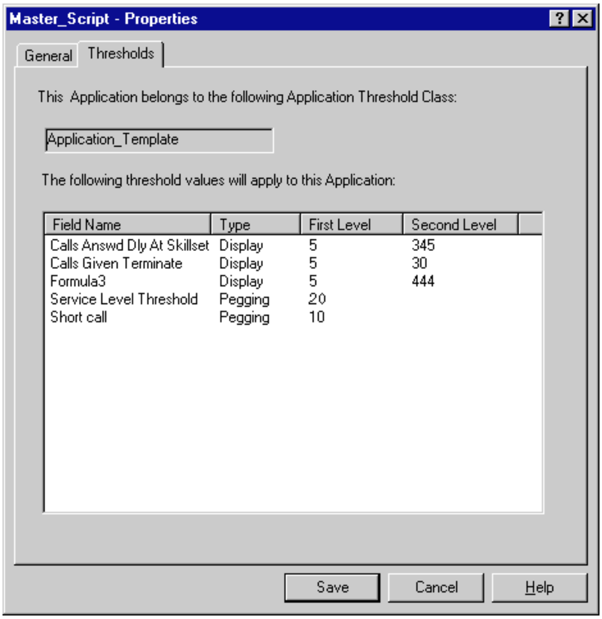
Result: The Applications window appears.

2 Select the application that you want to view.

3 Choose File → Properties.

Result: The Application Properties page appears.

4 Choose the Thresholds tab.



This Application belongs to the following Application Threshold Class:

Application_Template

The following threshold values will apply to this Application:

Field Name	Type	First Level	Second Level
Calls Answd Dly At Skillset	Display	5	345
Calls Given Terminate	Display	5	30
Formula3	Display	5	444
Service Level Threshold	Pegging	20	
Short call	Pegging	10	

5 Click Save to close the Application Properties Page.

To change threshold classes

1 From the SMI window, choose Call Flow Administration → Applications.

Result: The Applications window appears.

2 Select the application that you want to change.

3 Choose File → Properties.

4 The Applications Properties page appears.

5 From the Threshold Class list, select the new threshold class.

6 Click Save.

Chapter 12

Using sample scripts

In this chapter

Overview

Introduction

Nortel Networks provides sample scripts with Symposium Call Center Server to get you started on your server’s scripts. This chapter contains the following information.

Section A, “Getting started with sample scripts,” explains how to create the initial scripts that you need to get started, and how to edit the Master script.

Section B, “Sample scripts,” lists the sample scripts that are included with the Symposium Call Center Server Client software, and gives a brief explanation of what each script does.

You can also import sample scripts into the Scripts Editor, and then copy the sections that you need into your own scripts. For more information, see “Importing scripts into Symposium Call Center Server” on page 106, and “Copying text into scripts” on page 97.

Note: The variables used in the sample scripts are examples only. If you use a sample script that contains variables, you have to create and define the variables on your system. For more information, see “Creating script variables” on page 73.

Advanced commands

Some sample scripts use advanced commands. You must purchase the Symposium Call Center Server 200 application package to be able to use advanced commands. For a list of advanced commands you can use Symposium Call Center Server, see Chapter 7, “Advanced script commands.”

Section A: Getting started with sample scripts

In this section

Overview

Introduction

This section outlines how to begin using sample scripts.

Before you begin

Before you begin using sample scripts, Symposium Call Center Server must be installed and configured. In addition, all system resources such as RAN routes, music routes, voice ports, call treatments, and DNs must be set up. For more information about setting up these resources, refer to the *Switch Guide.*

All variables, agents, and skillsets must also be created. For more information about creating agents and skillsets, refer to the *Administrator's Guide.*

If you plan to use a third party IVR system to collect caller-entered data, you must configure the system before you can use the caller-entered data in your scripts. See the documentation that came with your IVR system for more information.

Finally, if you plan to use a third-party screen-pop application, you must set up Meridian Link Services, TAPI, and CTI.

Note: The voice ports and position IDs for the IVR system on the switch must be identical to the configuration on the Symposium Call Center Server.

Variables

The variables used in the sample scripts are examples only. If you use a sample script that contains variables, you have to create and define the variables on your system. For more information, see “Creating script variables” on page 73.

Steps to take

Once your system is configured and you have created all variables you need, complete the following procedures:

- “Creating the initial scripts” on page 298
- “Editing the Master script” on page 301

When you have completed these procedures, you can test the scripts by placing calls to your call center.

Creating the initial scripts

Introduction

This section explains how to create initial scripts from the sample scripts provided with Symposium Call Center Server. Create these scripts to test how calls are handled once they arrive at the Symposium Call Center Server.

The sample scripts directory contains a subdirectory named “routing examples.” This directory contains three types of scripts (Master, primary, and secondary) that must reference each other to function properly. You must create and activate one Master script, one primary script, and one secondary script to begin testing call routing in your call center.

To begin using the routing example sample scripts on your system, create the following scripts:

- c_Primary_One
- c_Common_Secondary

To create and validate initial scripts

1. From the SMI window, choose Call Flow Administration → Scripts.

   Result: The Script Manager appears.

2. Choose File → New.

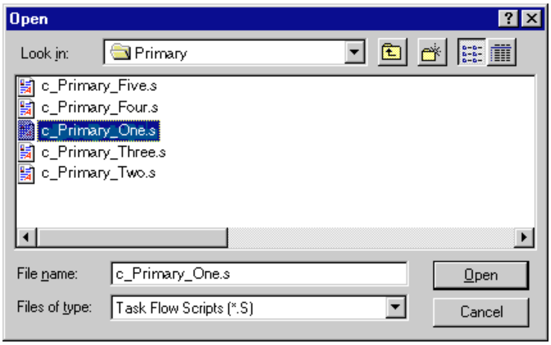
   Result: The Script Editor appears.

3. Choose File → Import.

   Result: A dialog box appears prompting you for the location of the script you want to import.

4. Navigate to the C:\Nortel\client\en\scripts\samples\routing examples\primary\ directory.

5 Select the primary sample script that you want to create. For example, select the script, c_Primary_One.s.



Look in: Primary

c_Primary_Five.s
c_Primary_Four.s
c_Primary_One.s
c_Primary_Three.s
c_Primary_Two.s

File name: c_Primary_One.s

Files of type: Task Flow Scripts (*.S)

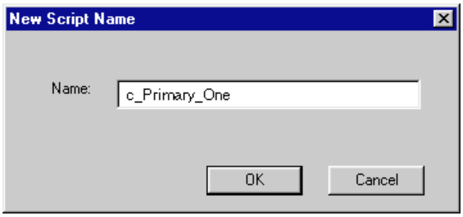
6 Click Open to import the script.

7 From the Script Editor, choose File → Save.

Result: A dialog box appears asking if you are sure you want to save the script to the server.

8 Click OK.

Result: The New Script Name window appears.



New Script Name

Name: c_Primary_One

9 Enter the same name as the sample script, for example, c_Primary_One.

Note: Do not enter the file extensions.

10 Click OK.

11 From the Script Editor, choose Edit → Validate.

Note: If the validation fails, it might be because of a missing variable or a variable of the wrong type. To find out what variables you need for each script, refer to the script that you want to use in Section B, “Sample scripts.”

12 Next, create a secondary script. From the Script Manager, choose File → New.

Result: The Script Editor appears.

13 Choose File → Import.

Result: A dialog box appears prompting you for the location of the script you want to import.

14 Navigate to the C:\Nortel\client\en\scripts\samples\routing examples\secondary\ directory.

15 Select the secondary script that you want to create. For example, select the script, c_Common_Secondary.s.

16 Click Open to import the script.

17 From the Script Editor, choose File → Save.

Result: A dialog box appears asking if you are sure you want to save the script to the server.

18 Click OK.

Result: The New Script Name window appears.

19 Enter the same name as the sample script, for example, c_Common_Secondary.

Note: Do not enter the file extensions.

20 Click OK.

21 From the Script Editor, choose Edit → Validate.

Note: If the validation fails, it might be because of a missing variable or a variable of the wrong type. To find out what variables you need for each script, refer to the script that you want to use in Section B, “Sample scripts.”

Editing the Master script

Introduction

This section explains how to import the contents of the sample script "c_Master_with_Exception_Checks" into a new script. The script you create serves as the Master script for Symposium Call Center Server.

To edit the Master script

1 From the SMI window, choose Call Flow Administration → Scripts.

Result: The Script Manager appears.

2 Double-click Master_Script.

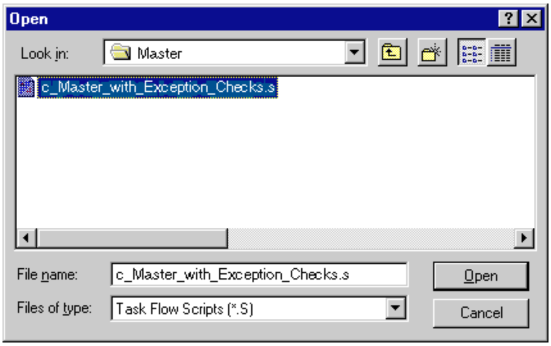
Result: The Script Editor appears.

3 Choose File → Import.

Result: A dialog box appears prompting you for the location of the script you want to import.

4 Navigate to the C:\Nortel\client\en\scripts\samples\routing examples\master\ directory.

5 Select c_Master_with_Exception_Checks.s.



Look in: Master

File name: c_Master_with_Exception_Checks.s

Files of type: Task Flow Scripts (*.S)

6 Click Open to import the script.

7 From the Script Editor, choose File → Activate.

Result: The script is automatically validated. If the validation fails, it might be because of a missing variable, a variable of the wrong type, or a dependent script that has not been validated. To find out what variables you need for each script, refer to the script that you want to use in Section B, “Sample scripts.”

The c_Primary_One and c_Common_Secondary scripts are automatically activated.

You can now test your scripts by placing a call to Symposium Call Center Server.

Section B: Sample scripts

In this section

Common scripts

Introduction

Symposium Call Center Server comes with ten common scripts that you can use in your call center. The sample scripts are located in the C:\Nortel\client\en\scripts\samples\common\ directory. The sample scripts serve as primary scripts. You do not need to reference any other scripts to test call routing in your call center when using these scripts.

Script name

c_Basic

Function

In the following script, a test is performed to determine if the call center is closed due to a holiday or a weekend, or if it is after business hours. If the call center is open and agents with the required skillset are available to answer the call, it is queued to the skillset, and the caller hears a recorded announcement and then music.

While the caller is waiting, the call is repeatedly checked to determine if it is still queued. If not, a test is performed to determine if agents are still logged on to the skillset. If all agents are logged off, the caller hears a recorded announcement and is disconnected; otherwise, the call is requeued to the skillset.

Script text

```
/*   Title:  c_Basic

Note: Replace skillset_sk with the desired skillset name.


   Global Variable List:     Variable Type:

   1. holidays_gv          - DATE (holiday dates)
   2. weekend_gv            - DAY  (contains Saturday and
                              Sunday)
   3. closed_hours_gv       - TIME (value for closed hours)
   4. first_ran             - RAN (RAN route for first
                              announcement)
   5. second_ran            - RAN (RAN route for second
                              announcement)
   6. closed_ran            - RAN (RAN route for night
                              announcement)
   7. dayclosed_ran         - RAN (RAN route for day closed
                              announcement)
   8. treatment_timer_gv    - SECONDS (delay time for wait)
   9. music_route           - MUSIC (route number for
                              music)
*/


IF (DATE = holidays_gv)
OR (DAY OF WEEK = weekends_gv)
OR (TIME OF DAY = closed_hours_gv) THEN
   GIVE RAN closed_ran
   DISCONNECT
END IF


IF OUT OF SERVICE skillset_sk THEN
   GIVE RAN dayclosed_ran
   DISCONNECT
END IF


QUEUE TO SKILLSET skillset_sk
WAIT 2   /* Allow time in case an agent is available */
```

```
GIVE RAN first_ran


SECTION WaitLoop
    GIVE MUSIC music_route
    WAIT treatment_timer_gv
    IF NOT QUEUED THEN
        IF OUT OF SERVICE skillset_sk THEN
             GIVE RAN dayclosed_ran
             DISCONNECT
       ELSE
             QUEUE TO SKILLSET skillset_sk
             WAIT 2   /* Allow time in case an agent is
                      available */
       END IF
    END IF
    GIVE RAN second_ran
EXECUTE WaitLoop
```

Script name

c_Basic_Backup_Skillset

Function

In the following script, a test is performed to determine if the call center is closed due to a holiday or a weekend, or if it is after business hours. If the call center is open and agents with the required skillset are available to answer the call, it is queued to the skillset, and the caller hears a recorded announcement and then music.

After ten seconds, a test is performed to determine if agents are still logged on to the skillset. While the caller is waiting, the call is repeatedly checked to determine if it is still queued. If not, a test is performed to determine if agents are logged on to the backup skillset. If all agents in the backup skillset are logged off, the caller hears a recorded announcement and is disconnected; otherwise, the call is requeued to the skillset.

Script text

```
/*   Title:  c_Basic_Backup_Skillset

Note: Replace skillset_sk and backup_skillset_sk with the
desired skillset names.


    Global Variable List:    Variable Type:

   1. holidays_gv          - DATE (holiday dates)

   2. weekend_gv           - DAY (contains Saturday and
                             Sunday)

   3. closed_hours_gv      - TIME (value for closed hours)

   4. first_ran            - RAN (audio route for first
                             announcement)

   5. second_ran           - RAN (audio route for second
                             announcement)

   6. closed_ran           - RAN (audio route for night
                             announcement)

   7. dayclosed_ran        - RAN (audio route for day
                             closed announcement)

   8. treatment_timer_gv   - SECONDS (delay time for wait)

   9. music_route          - MUSIC (audio route number for
                             music)

*/


IF (DATE = holidays_gv)

OR (DAY OF WEEK = weekend_gv)
OR (TIME OF DAY = closed_hours_gv) THEN
   GIVE RAN closed_ran
   DISCONNECT
END IF

IF OUT OF SERVICE skillset_sk THEN
   GIVE RAN dayclosed_ran
   DISCONNECT
END IF

QUEUE TO SKILLSET skillset_sk
WAIT 2   /* Allow time in case an agent is available */
GIVE RAN first_ran
GIVE MUSIC music_route
WAIT 10

IF NOT OUT OF SERVICE backup_skillset_sk THEN
   QUEUE TO SKILLSET backup_skillset_sk
   WAIT 2
END IF

SECTION WaitLoop
    WAIT treatment_timer_gv
    IF NOT QUEUED THEN
        IF OUT OF SERVICE skillset_sk THEN
             IF OUT OF SERVICE backup_skillset_sk THEN
                   GIVE RAN dayclosed_ran
                   DISCONNECT
```

```
                    ELSE

                          QUEUE TO SKILLSET backup_skillset_sk

                          WAIT 2

                    END IF

               ELSE

                    QUEUE TO SKILLSET skillset_sk

                    WAIT 2  /*Allow time in case an agent is
                              available*/

               END IF

           END IF

           GIVE RAN second_ran

           GIVE MUSIC music_route

       EXECUTE WaitLoop
```

Script name

c_Basic_with_EWT

Function

In the following script, a number of tests are performed to determine how best to route the call. The script firsts determines if the call center is closed due to a holiday or a weekend, or if it is after business hours. If the call center is open and agents with the required skillset are available to answer the call, it is queued to the skillset, and the caller hears a recorded announcement about the amount of time he or she can expect to wait before the call is answered.

While the caller is waiting, the call is repeatedly checked to determine if it is still queued. If not, a test is performed to determine if agents are still logged on to the skillset. If all agents are logged off, the caller hears a recorded announcement and is disconnected; otherwise, the call is requeued to the skillset.

Script text

```
/* Title:  c_Basic_with_EWT
```

```
Note: Replace skillset_sk with the desired skillset name.

    Global Variable List:   Variable Type:

   1. holidays_gv         - DATE (holiday dates)

   2. weekend_gv          - DAY (contains Saturday and
                           Sunday)

   3. closed_hours_gv     - TIME (value for closed hours)

   4. short_wait_ran      - RAN (audio route for first
                           announcement)

   5. medium_wait_ran     - RAN (audio route for first
                           announcement)

   6. long_wait_ran       - RAN (audio route for first
                           announcement)

   7. second_ran          - RAN (audio route for second
                           announcement)

   8. closed_ran          - RAN (audio route for night
                           announcement)

   9. dayclosed_ran       - RAN (audio route for day closed
                           announcement)

   10.treatment_timer_gv - TIME (in seconds)

   11.music_route         - MUSIC (audio route number for
                           music)


    Call Variable List:     Variable Type:

   1. expwait_cv          - INTEGER or Seconds

*/


IF (DATE = holidays_gv)

OR (DAY OF WEEK = weekend_gv)

OR (TIME OF DAY = closed_hours_gv) THEN

   GIVE RAN closed_ran
   DISCONNECT
END IF

IF OUT OF SERVICE skillset_sk THEN
   GIVE RAN dayclosed_ran
   DISCONNECT
END IF

QUEUE TO SKILLSET skillset_sk
WAIT 2   /* Allow time in case an agent is available */

ASSIGN EXPECTED WAIT TIME skillset_sk TO expwait_cv

IF (expwait_cv < 100) THEN
   GIVE RAN short_wait_ran
ELSE
   IF (expwait_cv < 200) THEN
       GIVE RAN medium_wait_ran
   ELSE
       GIVE RAN long_wait_ran
   END IF
END IF

SECTION WaitLoop
    GIVE MUSIC music_route
    WAIT treatment_timer_gv
    IF NOT QUEUED THEN
        IF OUT OF SERVICE skillset_sk THEN
```

```
                GIVE RAN dayclosed_ran
                DISCONNECT
            ELSE
                QUEUE TO SKILLSET skillset_sk
                WAIT 2   /* Allow time in case an agent is
                          available */
            END IF
        END IF
        GIVE RAN second_ran
EXECUTE WaitLoop
```

Script name

c_Call_Data_Assigns_Priority_in_Queue

Function

In the following script, a number of tests are performed to determine how best to route the call. The script firsts determines if the call center is closed due to a holiday or a weekend, or if it is after business hours. If the call center is closed and no agents with the required skillset are available to answer the call, the caller hears a recorded announcement and is disconnected.

If the call center is open and agents with the required skillset are available to answer the call, it is queued to the skillset with a priority level determined during an IVR session established with the caller.

While the caller is waiting, the call is repeatedly checked to determine if it is still queued. If not, a test is performed to determine if agents are still logged on to the skillset. If all agents are logged off, the caller hears a recorded announcement and is disconnected; otherwise, the call is requeued to the skillset.

Script text

```
/*  Title:  c_Call_Data_used_to_assign_Priority_in_Queue
```

```
   Note: Replace skillset_sk with the desired skillset
   name.


    Global Variable List:  Variable Type:

   1. holidays_gv         - DATE (holiday dates)

   2. weekend_gv          - DAY (contains Saturday and
                           Sunday)

   3. closed_hours_gv     - TIME (value for closed hours)

   4. CALL DATA 1         - CALL DATA (values passed from
                           IVR session)

   5. first_ran           - RAN (audio route for first
                           announcement)

   6. second_ran          - RAN (audio route for second
                           announcement)

   7. closed_ran          - RAN (audio route for night
                           announcement)

   8. dayclosed_ran       - RAN (audio route for day closed
                           announcement)

   9. treatment_timer_gv - SECONDS (delay time for wait)

   10.music_route         - MUSIC (audio route number for
                           music)


    Call Variable List:     Variable Type:

   1. sk_priority_cv      - INTEGER (priority number)

*/


IF (DATE = holidays_gv)

OR (DAY OF WEEK = weekend_gv)

OR (TIME OF DAY = closed_hours_gv) THEN

   GIVE RAN closed_ran

   DISCONNECT
```

```
END IF

IF OUT OF SERVICE skillset_sk THEN
   GIVE RAN dayclosed_ran
   DISCONNECT
END IF

WHERE CALL DATA 1 EQUALS
   VALUE <data>:ASSIGN 1 TO sk_priority_cv
   VALUE <data>:ASSIGN 2 TO sk_priority_cv
   VALUE <data>:ASSIGN 3 TO sk_priority_cv
   DEFAULT:  ASSIGN 4 TO sk_priority_cv
END WHERE

QUEUE TO SKILLSET skillset_sk WITH PRIORITY sk_priority_cv
WAIT 2   /* Allow time in case an agent is available */
GIVE RAN first_ran

SECTION WaitLoop
    GIVE MUSIC music_route
    WAIT treatment_timer_gv
    IF NOT QUEUED THEN
        IF OUT OF SERVICE skillset_sk THEN
             GIVE RAN dayclosed_ran
             DISCONNECT
        ELSE
             QUEUE TO SKILLSET skillset_sk WITH PRIORITY
             sk_priority_cv
```

```
                WAIT 2 /*Allow time in case an agent is
                       available */
           END IF
       END IF
       GIVE RAN second_ran
   EXECUTE WaitLoop
```

Script name

c_Emergency_Boolean

Function

In the following script, a test is performed to determine if there is an emergency in the call center. If so, callers receive a recorded announcement and are disconnected. If there is no emergency, and agents with the required skillset are available to answer the call, it is queued to that skillset.

While the caller is waiting, the call is repeatedly checked to determine if it is still queued, and the caller hears a series of recorded announcements. If the call is still in the queue after all of the recorded announcements have played, the caller receives a message informing him or her that all agents are busy, and the call remains in the queue.

Script text

```
/* Title:  c_Emergency_Boolean

Note: Replace skillset_sk with desired skillset name.


   Global Variable List:  Variable Type:

   1. emergency_gv      - BOOLEAN (true or false)

   2. emerg_ran         - RAN (audio route for
                         announcement)

   3. reps_busy_ran     - RAN (audio route for
                         announcement)
```

```
    4. tip1_ran          - RAN (audio route for
                           announcement)

    5. tip2_ran          - RAN (audio route for
                           announcement)

    6. still_busy_ran    - RAN (audio route for
                           announcement)

    7. day_closed_ran    - RAN (audio route for
                           announcement)

    8. music_soft        - MUSIC (audio route for music)

*/


IF emergency_gv  THEN

   GIVE RAN emerg_ran

   DISCONNECT

END IF


IF OUT OF SERVICE skillset_sk THEN

   EXECUTE DayClosed

END IF


QUEUE TO SKILLSET skillset_sk

WAIT 2

GIVE RAN reps_busy_ran



SECTION General

   GIVE MUSIC music_soft

   WAIT 45

   IF NOT QUEUED THEN

        IF OUT OF SERVICE skillset_sk THEN
```

```
          EXECUTE DayClosed
     ELSE
          QUEUE TO SKILLSET skillset_sk
          WAIT 2
     END IF
END IF
GIVE RAN tip1_msg_ran
GIVE MUSIC music_soft
WAIT 60
IF NOT QUEUED THEN
     IF OUT OF SERVICE skillset_cv THEN
          EXECUTE DayClosed
     ELSE
          QUEUE TO SKILLSET skillset_cv
          WAIT 2
     END IF
END IF
GIVE RAN still_busy_ran
GIVE MUSIC music_soft
WAIT 90
IF NOT QUEUED THEN
     IF OUT OF SERVICE skillset_cv THEN
          EXECUTE DayClosed
     ELSE
          QUEUE TO SKILLSET skillset_cv
          WAIT 2
     END IF
END IF
```

```
   GIVE RAN tip2_msg_ran
   GIVE MUSIC music_soft
   WAIT 60
   EXECUTE ClosedCheckLoop


SECTION ClosedCheckLoop

   IF NOT QUEUED THEN
        IF OUT OF SERVICE skillset_sk THEN
             EXECUTE DayClosed
        ELSE
             QUEUE TO SKILLSET skillset_sk
             WAIT 2
        END IF
   END IF
   GIVE RAN still_busy_ran
   GIVE MUSIC music_soft
   WAIT 180
   EXECUTE ClosedCheckLoop

SECTION DayClosed
   GIVE RAN day_closed_ran
   DISCONNECT
```

Script name

c_Emergency_Skillset_Check

Function

In the following script, a test is performed to determine if the call center is closed due to a holiday or a weekend, or if it is after business hours. If so, the caller hears a recorded announcement and is disconnected. If the call center is not closed, a test is performed to determine if there is an emergency in the call center. If so, the caller hears a recorded announcement and is disconnected.

If the call center is open, there is no emergency, and agents with the required skillset are available to answer the call, it is queued to the skillset.

While the caller is waiting, the call is repeatedly checked to determine if it is still queued. If not, a test is performed to determine if agents are still logged on to the skillset. If all agents are logged off, the caller hears a recorded announcement and is disconnected; otherwise, the call is requeued to the skillset. If the call is requeued, a test is repeatedly performed to determine if an emergency has been declared while the caller waits in the queue.

Script text

```
/*   Title:  c_Emergency_Skillset_Check

Note: Replace skillset_sk with the desired skillset name.


   Global Variable List:   Variable Type:

   1. holidays_gv        - DATE (holiday dates)

   2. weekend_gv         - DAY (contains Saturday and
                          Sunday)

   3. closed_hours_gv    - TIME (value for closed hours)

   4. emergency_ran      - RAN (audio route for emergency
                          announcement)

   5. first_ran          - RAN (audio route for first
                          announcement)

   6. second_ran         - RAN (audio route for second
                          announcement)

   7. closed_ran         - RAN (audio route for night
                          announcement)
   8. dayclosed_ran      - RAN (audio route for day closed
                           announcement)

   9. treatment_timer_gv - SECONDS (delay time for wait)

   10.music_route        - MUSIC (audio route number for
                           music)

*/

IF (DATE = holidays_gv)

OR (DAY OF WEEK = weekend_gv)

OR (TIME OF DAY = closed_hours_gv) THEN

   GIVE RAN closed_ran

   DISCONNECT

END IF


IF NOT OUT OF SERVICE emergency_sk THEN

   GIVE RAN emergency_ran

   DISCONNECT

END IF


IF OUT OF SERVICE skillset_sk THEN

   GIVE RAN dayclosed_ran

   DISCONNECT

END IF


QUEUE TO SKILLSET skillset_sk

WAIT 2   /* Allow time in case an agent is available */

GIVE RAN first_ran


SECTION WaitLoop

   GIVE MUSIC music_route
```

```
    WAIT treatment_timer_gv
    IF NOT OUT OF SERVICE emergency_sk THEN
        GIVE RAN emergency_ran
        DISCONNECT
     END IF
     IF NOT QUEUED THEN
         IF OUT OF SERVICE skillset_sk THEN
              GIVE RAN dayclosed_ran
              DISCONNECT
         ELSE
              QUEUE TO SKILLSET skillset_sk
              WAIT 2   /* Allow time in case an agent is
                        available */
        END IF
     END IF
     GIVE RAN second_ran
EXECUTE WaitLoop
```

Script name

c_Excess_Call_Volume_Give_Busy

Function

In the following script, a test is performed to determine if the call center’s capacity to handle calls has exceeded the specified threshold. If so, callers receive a busy treatment; otherwise, the call is queued to the specified skillset.

Script text

```
/* Title: c_Excess_Call_Volume_Give_Busy
```

```
  Global Variable List:         Variable Type:

    1. holidays_gv             - DATE (holiday dates)

    2. weekend_gv              - DAY (contains Saturday and
                                 Sunday)

    3. closed_hours_gv         - TIME (value for closed
                                 hours)

    4. skillset_threshold_gv - INTEGER (value for set
                                 threshold)

    5. first_ran               - RAN (audio route for first
                                 announcement)

    6. second_ran              - RAN (audio route for second
                                 announcement)

    7. closed_ran              - RAN (audio route for night
                                 announcement)

    8. dayclosed_ran           - RAN (audio route for day
                                 closed announcement)

    9. treatment_timer_gv      - SECONDS (delay time for wait)

    10. music_route            - MUSIC (audio route number
                                 for music)

*/

IF (DATE = holidays_gv)

OR (DAY OF WEEK = weekend_gv)

OR (TIME OF DAY = closed_hours_gv) THEN

   GIVE RAN closed_ran

   DISCONNECT

END IF


IF OUT OF SERVICE skillset_sk THEN

   GIVE RAN dayclosed_ran

   DISCONNECT

END IF
```

```
IF (QUEUED CALL COUNT skillset_sk >
skillset_threshold_gv) THEN
   GIVE BUSY
END IF

QUEUE TO SKILLSET skillset_sk
WAIT 2   /* Allow time in case an agent is available */
GIVE RAN first_ran

SECTION WaitLoop]
    GIVE RAN first_ran
    WAIT treatment_timer_gv
    IF NOT QUEUED THEN
        IF OUT OF SERVICE skillset_sk THEN
             GIVE RAN dayclosed_ran
             DISCONNECT
        ELSE
             QUEUE TO SKILLSET skillset_sk
             WAIT 2   /* Allow time in case an agent is
                       available */
        END IF
    END IF
    GIVE RAN second_ran
EXECUTE WaitLoop
```

Script name

c_Forced_Announcement

Function

In the following script, a test is performed to determine if agents with the required skillset are available to answer the call. If so, the caller hears an entire recorded announcement, and the call is then queued to the skillset.

While the caller is waiting, the call is repeatedly checked to determine if it is still queued. If not, a test is performed to determine if agents are still logged on to the skillset. If all agents are logged off, the call is routed to a mailbox and the caller is asked to leave a recorded announcement. If agents are available to answer the call, it is requeued to the skillset.

Script text

```
/* Title:  c_Forced_Announcement


    Global Variable List:    Variable Type:

   1. holidays_gv           - DATE (holiday dates)

   2. weekend_gv            - DAY (contains Saturday and
                              Sunday)

   3. closed_hours_gv       - TIME (value for closed hours)

   4. first_ran             - RAN (audio route for
                              announcement)

   5. second_ran            - RAN (audio route for
                              announcement)

   6. closed_ran            - RAN (audio route for
                              announcement)

   7. dayclosed_ran         - RAN (audio route for
                              announcement)

   8. music_route           - MUSIC (audio route number for
                              music)

   9. treatment_delay_gv    - SECONDS (delay time for wait)
*/


IF (DATE = holidays_gv)
```

```
OR (DAY OF WEEK = weekends_gv)
OR (TIME OF DAY = closed_hours_gv) THEN
   GIVE RAN closed_ran
   DISCONNECT
END IF

IF OUT OF SERVICE skillset_sk THEN
   GIVE RAN dayclosed_ran
   DISCONNECT
END IF

GIVE RAN first_ran
/*This forces the RAN to be heard before queuing a call*/

QUEUE TO SKILLSET skillset_sk
WAIT 2
GIVE MUSIC music_route

SECTION WaitLoop
   WAIT treatment_delay_gv
   IF NOT QUEUED THEN
       IF OUT OF SERVICE skillset_sk THEN
             ROUTE CALL mailbox_gv
       ELSE
             QUEUE TO SKILLSET skillset_sk
             WAIT 2
       END IF
   END IF
   GIVE RAN second_ran
EXECUTE WaitLoop
```

Script name

c_Holiday_Announcement

Function

In the following script, a test is performed to determine if the call center is closed due to a holiday. If so, the caller hears a special holiday announcement. If it is a weekend, or if it is after business hours, the caller hears a recorded announcement informing him or her that the call center is closed. If the call center is open, and agents with the required skillset are available to handle the call, it is queued to the skillset.

While the caller is waiting, the call is repeatedly checked to determine if it is still queued. If not, a test is performed to determine if agents are still logged on to the skillset. If all agents are logged off, the caller hears a recorded announcement and is disconnected. If agents are available to answer the call, it is requeued to the skillset.

Script text

```
/* Title: c_Holiday_Announcement
Note: Replace skillset_sk with the desired skillset name.

      Global Variable List:  Variable Type:
   1. holiday_gv             - DATE of specific holidays
   2. closed_hours_gv        - TIME that business is closed
   3. weekends_gv            - DAY(s) that business is
                              closed
   5. closed_ran             - RAN (audio route for night
                              announcement)
   6. holiday_ran            - RAN (audio route for holiday
                              announcement)
   7. first_ran             - RAN (audio route for first
                              announcement)

   8. second_ran            - RAN (audio route for second
                              announcement)

   9. dayclosed_ran         - RAN (audio route for day
                              closed announcement)

   10.treatment_timer_gv    - SECONDS (delay time for wait)

   11.music_route           - MUSIC (audio route for music)

*/


IF (DATE = holiday_gv) THEN

   GIVE RAN holiday_ran

   DISCONNECT

END IF


IF (DAY OF WEEK = weekends_gv)

OR (TIME OF DAY = closed_hours_gv) THEN

   GIVE RAN holiday_ran

   DISCONNECT

END IF


IF OUT OF SERVICE skillset_sk THEN

   GIVE RAN closed_ran

   DISCONNECT

END IF


QUEUE TO SKILLSET skillset_sk

WAIT 2   /* Allow time in case an agent is available */

GIVE RAN first_ran
```

```
SECTION WaitLoop
    GIVE MUSIC music_route
    WAIT treatment_timer_gv
    IF NOT QUEUED THEN
        IF OUT OF SERVICE skillset_sk THEN
             GIVE RAN dayclosed_ran
             DISCONNECT
        ELSE
             QUEUE TO SKILLSET skillset_sk
             WAIT 2   /* Allow time in case an agent is
                       available */
        END IF
    END IF
    GIVE RAN second_ran
EXECUTE WaitLoop
```

Script name

c_Priority_in_Queue_DNIS

Function

In the following script, a test is performed to determine if the call center is closed due to a holiday or a weekend, or if it is after business hours. If the call center is open and agents with the required skillset are available to answer the call, it assigned a priority level based on the call’s DNIS number, and then queued to the skillset.

While the caller is waiting, the call is repeatedly checked to determine if it is still queued. If not, a test is performed to determine if agents are still logged on to the skillset. If all agents are logged off, the caller hears a recorded announcement and is disconnected. If agents are available to answer the call, it is requeued to the skillset.

Script text

```
/*  Title:  c_Priority_in_Queue_DNIS

Note: Replace skillset_sk with the desired skillset name.


    Global Variable List:    Variable Type:

   1. holidays_gv          - DATE (holiday dates)

   2. weekend_gv           - DAY (contains Saturday and
                             Sunday)

   3. closed_hours_gv      - TIME (value for closed hours)

   4. dnis_1               - DNIS

   5. dnis_2               - DNIS

   6. dnis_3               - DNIS

   7. first_ran            - RAN (audio route for first
                             announcement)

   8. second_ran           - RAN (audio route for second
                             announcement)

   9. closed_ran           - RAN (audio route for night
                             announcement)

   10. dayclosed_ran       - RAN (audio route for day
                             closed announcement)

   11. treatment_timer_gv  - SECONDS (dalay time for wait)

   12. music_route         - MUSIC (audio route number for
                             music)


    Call Variable List:      Variable Type:

   1. sk_priority_cv       - INTEGER priority number

*/


IF (DATE = holidays_gv)

OR (DAY OF WEEK = weekends_gv)
OR (TIME OF DAY = closed_hours_gv) THEN
   GIVE RAN closed_ran
   DISCONNECT
END IF

IF OUT OF SERVICE skillset_sk THEN
   GIVE RAN dayclosed_ran
   DISCONNECT
END IF

WHERE DNIS EQUALS
   VALUE dnis_1:ASSIGN 1 TO sk_priority_cv
   VALUE dnis_2:ASSIGN 2 TO sk_priority_cv
   VALUE dnis_3:ASSIGN 3 TO sk_priority_cv
   DEFAULT:  ASSIGN 4 TO sk_priority_cv
END WHERE

QUEUE TO SKILLSET skillset_sk WITH PRIORITY sk_priority_cv
WAIT 2   /* Allow time in case an agent is available */
GIVE RAN first_ran

SECTION WaitLoop
    GIVE MUSIC music_route
    WAIT treatment_timer_gv
    IF NOT QUEUED THEN
        IF OUT OF SERVICE skillset_sk THEN
             GIVE RAN dayclosed_ran
             DISCONNECT
```

```
            ELSE

                QUEUE TO SKILLSET skillset_sk WITH PRIORITY
                sk_priority_cv

                WAIT 2   /* Allow time in case an agent is
                                available */

            END IF

        END IF

        GIVE RAN second_ran

    EXECUTE WaitLoop
```

Routing examples

Introduction

These sample scripts are located in the C:\Nortel\client\en\scripts\samples\routing examples\ directory. This directory contains three types of scripts (Master, primary, and secondary) that must reference each other to function properly. You must create and activate one Master script, one primary script, and one secondary script to begin testing call routing in your call center.

Master scripts

Master scripts are located in the C:\Nortel\client\en\scripts\samples\routing examples\master\ directory. If you intend to use a Master script, it must reference a primary script, which in turn must reference a secondary script.

Script name

c_Master_with_Exception_Checks

Function

In the following script, a test is performed to determine if the call center is closed due to a holiday, or if there is an emergency in the caller center. If the call center is open and there is no emergency, the script determines at which CDN the call arrived. If the call was transferred, the script determines if the transferring agent is still on the line. If so, the script waits for six seconds, allowing the agent to hang up. If the agent has hung up or the call was not transferred, a primary script is executed based on the CDN at which the call arrived.

Script text

```
/* Title:  c_Master_Script_with_Exception_Checks
```

```
Note: Replace <CDN> and <Script_Names> with site specific
values.



   Global Variable List:    Variable Type:

   1. holiday_gv          - DATE (holiday dates)

   2. spec_holiday_gv     - DATE (special holiday dates)

   3. spec_holidayhrs_gv  - TIME (special value for
                            holiday hours)

   4. holiday_ran         - RAN (audio route for
                            announcement)

   5. emergency_gv        - BOOLEAN (value is normally
                            false)

   6. emerg_ran           - RAN (audio route for
                            announcement)

*/


IF (DATE = holiday_gv)

OR (DATE = spec_holiday_gv AND TIME OF DAY =
spec_holidayhrs_gv) THEN

   GIVE RAN holiday_ran

   DISCONNECT

END IF


IF emergency_gv THEN

   GIVE RAN emerg_ran

   DISCONNECT

END IF
```

```
IF CONSULTED THEN
   GIVE RINGBACK
   WAIT 6 /* wait for blind transfer complete */
END IF

WHERE CDN EQUALS
   VALUE <CDN>: EXECUTE SCRIPT <Script_Name>
   VALUE <CDN>: EXECUTE SCRIPT <Script_Name>
   VALUE <CDN>: EXECUTE SCRIPT <Script_Name>
   VALUE <CDN>: EXECUTE SCRIPT <Script_Name>
   VALUE <CDN>: EXECUTE SCRIPT <Script_Name>
   VALUE <CDN>: EXECUTE SCRIPT <Script_Name>
   VALUE <CDN>: EXECUTE SCRIPT <Script_Name>
   VALUE <CDN>: EXECUTE SCRIPT <Script_Name>
   VALUE <CDN>: EXECUTE SCRIPT <Script_Name>
   VALUE <CDN>: EXECUTE SCRIPT <Script_Name>
   DEFAULT:    ROUTE CALL DEFAULT DN
END WHERE
```

Primary scripts

Primary scripts are located in the C:\Nortel\client\en\scripts\samples\routing examples\primary\ directory. If you intend to use a primary script, it must reference a secondary script.

Script name

c_Primary_One

Function

This primary script defines the values of call variables and then directs the call to a common secondary script.

Script text

```
/* Title: c_Primary_One


   Call Variable List:    Variable Type:

   1. ran_cv             - RAN (will hold assigned audio
                           announcement route value)

   2. skillset_cv        - SKILLSET (will hold assigned
                           skillset value)

   3. time_loop_cv       - SECONDS (will hold value for
                           delay time)

*/


ASSIGN 100 TO ran_cv

ASSIGN A_skillset_sk TO skillset_cv

ASSIGN 20 TO time_loop_cv


EXECUTE SCRIPT c_Common_Secondary
```

Script name

c_Primary_Two

Function

This primary script defines the values of call variables and then directs the call to a common secondary script.

Script text

```
/* Title: c_Primary_Two


   Call Variable List:    Variable Type:

   1. ran_cv              - RAN (will hold assigned audio
                            announcement route value)

   2. skillset_cv         - SKILLSET (will hold assigned
                            skillset value)

   3. time_loop_cv        - SECONDS (will hold value for
                            delay time)

*/


ASSIGN 101 TO ran_cv

ASSIGN B_skillset_sk TO skillset_cv

ASSIGN 40 TO time_loop_cv


EXECUTE SCRIPT c_Common_Secondary
```

Script name

c_Primary_Three

Function

This primary script defines the values of call variables and then directs the call to a common secondary script.

Script text

```
/* Title: c_Primary_Three


   Call Variable List:    Variable Type:
```

```
   1. ran_cv             - RAN (will hold assigned audio
                           announcement route value)

   2. skillset_cv        - SKILLSET (will hold assigned
                           skillset value)

   3. time_loop_cv       - SECONDS (will hold value for
                           delay time)


*/


ASSIGN 103 TO ran_cv

ASSIGN C_skillset_sk TO skillset_cv

ASSIGN 30 TO time_loop_cv


EXECUTE SCRIPT c_Common_Secondary
```

Script name

c_Primary_Four

Function

This primary script defines the values of call variables and then directs the call to a common secondary script.

Script text

```
/* Title: c_Primary_Four


   Call Variable List:    Variable Type:

   1. ran_cv             - RAN (will hold assigned audio
                           announcement route value)

   2. skillset_cv        - SKILLSET (will hold assigned
                           skillset value)
   3. time_loop_cv      - SECONDS (will hold value for
                          delay time)

*/


ASSIGN 104 TO ran_cv

ASSIGN D_skillset_sk TO skillset_cv

ASSIGN 50 TO time_loop_cv


EXECUTE SCRIPT c_Common_Secondary
```

Script name

c_Primary_Five

Function

This primary script defines the values of call variables and then directs the call to a common secondary script.

Script text

```
/* Title: c_Primary_Five


   Call Variable List:    Variable Type:

   1. ran_cv             - RAN (will hold assigned audio
                          announcement route value)

   2. skillset_cv        - SKILLSET (will hold assigned
                          skillset value)

   3. time_loop_cv       - SECONDS (will hold value for
                          delay time)

*/


ASSIGN 105 TO ran_cv
ASSIGN E_skillset_sk TO skillset_cv
ASSIGN 30 TO time_loop_cv


EXECUTE SCRIPT c_Common_Secondary
```

Secondary scripts

Secondary scripts are located in the C:\Nortel\client\en\scripts\samples\routing examples\secondary\ directory. If you intend to use a secondary script, it must be referenced by a primary script.

Script name

c_Common_Secondary

Function

In the following script a test is performed to determine if the call center is closed due to a holiday or a weekend, or if it is after business hours. If so, the caller hears a recorded announcement and is disconnected; otherwise, the caller hears a message informing him or her that the call might be recorded, and then the call is queued to the appropriate skillset.

While the caller is waiting, the call is repeatedly checked to determine if it is still queued. If not, a test is performed to determine if agents are still logged on to the skillset. If all agents are logged off, the caller hears a recorded announcement and is disconnected. If agents are available to answer the call, it is requeued to the skillset.

Script text

```
/* Title: c_Common_Secondary


   Global Variable List:    Variable Type:
   1. holiday_gv          - DATE (holiday dates)
```

```
    2. weekend_gv          - DAY (weekend days: Saturday,
                             Sunday)

    3. closed_hrs_gv       - TIME (special value for closed
                             hours)

    4. nite_ran            - RAN (RAN route for night
                             announcement)

    5. wg_theme_gv         - MUSIC (route for music)

    6. dayclosed_ran       - RAN (RAN route for day closed
                             announcement)

    7. agents_busy_ran     - RAN (RAN route for agents busy
                             announcement)

*/


IF (DATE = holiday_gv)

OR (DAY OF WEEK = weekend_gv)

OR (TIME OF DAY = closed_hrs_gv) THEN

   GIVE RAN nite_ran

   DISCONNECT

END IF


IF OUT OF SERVICE skillset_cv THEN

   GIVE RAN dayclosed_ran

   DISCONNECT

END IF


GIVE RAN ran_cv  /* session may be recorded message*/

QUEUE TO SKILLSET skillset_cv

WAIT 2


SECTION Wait_Loop
```

```
   GIVE MUSIC wg_theme_gv
   WAIT time_loop_cv
   IF NOT QUEUED THEN
       IF OUT OF SERVICE skillset_cv THEN
             GIVE RAN dayclosed_ran
             DISCONNECT
       ELSE
             QUEUE TO SKILLSET skillset_cv
             WAIT 2
       END IF
   END IF
   GIVE RAN agents_busy_ran
EXECUTE Wait_Loop
```

Appendix A

Troubleshooting

In this appendix

Script execution problems

Assigned command fails to change value of a variable

The Assigned command has been changed to the Assign To command. The Assigned command continues to execute, however, Nortel Networks recommends that you use the Assign To command in your scripts.

The syntax of the Assign To command is: ASSIGN **<value>** TO **<variable>**. You can only use call variables of type Item.

Script does not validate

If the script executes a series of commands, but then ignores any following commands, you might not have included both an opening and a closing marker with comments inserted in the script. Comments included in your scripts must begin with an opening marker (/*) and end with a closing marker (*/). If you do not include both an opening and a closing marker, the script does not validate.

Callers are caught in an endless loop

A common mistake that scriptwriters make when using loops to give recorded announcements (GIVE RAN) to callers waiting in queue is to forget to include a test to determine if the call has been answered. If queuing fails or if the call is disconnected for some reason, the caller continues to hear a recorded announcement. However, the call is never answered.

It is important to include a test such as using the Queued or the Out of Service intrinsics inside loops. The results of these tests allow you to provide default treatments to the call to ensure that the call is successfully routed. In the following example, a call is queued to the sales skillset, and then the caller is given a recorded announcement. Every 30 seconds, a loop is used to test whether the call is still in queue or if agents are logged on to the sales skillset:

```
QUEUE TO SKILLSET sales_sk

WAIT 10

SECTION LoopForever
```

```
        IF NOT QUEUED THEN
            IF OUT OF SERVICE sales_sk THEN
                  ROUTE CALL auto_att
            ELSE
                  QUEUE TO SKILLSET sales_sk
                  WAIT 2
            END IF
        END IF
        GIVE RAN please_wait_ran_gv
        WAIT 30
    EXECUTE LoopForever
```

Calls are being routed to a default ACD-DN due to a scripting failure

If one of the commands listed below is the first command in a script, Symposium Call Center Server does not take control of the call. The DMS/MSL-100 routes the call to the default ACD-DN configured for the CDN. (The call is not queued to the default skillset or RAN.) The Symposium Call Center Server logs an error to the alarm monitor and event browser.

- WAIT
- QUIT
- GIVE SILENCE
- REMOVE FROM AGENT
- CHANGE PRIORITY IN AGENT
- REMOVE FROM SKILLSET
- CHANGE PRIORITY IN SKILLSET

The call must first be given treatment through the use of any other command before any of the above commands are used in a script.

Calls are not being presented to agents

If calls are not being presented to agents, the Variable Wrap feature might be configured incorrectly. If you are using the Variable Wrap option, agents go into Variable Wrap state after completing a call. (You must configure the length of the variable wrap period on the switch.) When the variable wrap period elapses, agents go into Ready state. If no calls are presented to agents when they return to Ready state, ensure that you have configured the correct settings for the Variable Wrap option on the switch. For information about how to configure variable wrap, refer to the *Symposium and DMS Switch Guide*.

List of validation errors

Introduction

This appendix contains a list of errors that you can receive when you validate a script. It lists the error numbers and the text of the error messages, and gives a brief explanation of each error.

Error number	14
Message	An end-of-file is encountered before the comment is terminated with an end-of-comment "*/".
Meaning	An end-of-comment symbol (*/) is missing. Insert the symbol at the end of commented text.
Error number	15
Message	Syntax error or unsupported command
Meaning	A syntax error can occur when the Script Validator cannot understand a command due to a missing or incorrect keyword. If you cannot see the error in the line of the script indicated by the Script Validator, check the line above for errors. For the correct syntax of commands, use the Script Command Reference panel, or refer to the description of the command as documented in this guide. An unsupported command is one that cannot be used in the Network script. **Note:** The Network script applies only if you have purchased the Network Skill-Based Routing (NSBR) feature for the Meridian 1 switch.
Error number	16
Message	An end-of-file is encountered in the script in the middle of a statement.

Meaning	There is an incomplete command in the line indicated by the Script Validator. For the correct syntax of the command, use the Script Command Reference panel, or see Chapter 6, “Basic script commands,” and Chapter 7, “Advanced script commands. ”
Error number	17
Message	A carriage return is encountered in a string literal before the end quotes.
Meaning	A string literal is a string of characters surrounded by quotation marks. For example, “This is a string.” There is a hard return inside a string of quoted text. Remove the hard return.
Error number	18
Message	An end-of-file is encountered in a string literal before the end quotes.
Meaning	A string literal is a string of characters surrounded by quotation marks. For example, “This is a string.” An end quote (”) is missing from the line indicated by the Script Validator.
Error number	19
Message	String literals cannot be longer than 80 characters.
Meaning	A string literal is a string of characters surrounded by quotation marks. For example, “This is a string.” There are too many characters in the string of text. Reduce the number of characters in the string to 80 or less.
Error number	20
Message	Valid integer constants cannot be greater than 1999999999 or less than –1999999999.
Meaning	Ensure that the number on the line indicated by the Script Validator is valid.
Error number	21
Message	WildCLID constants cannot be longer than 32 digits.

Meaning	There is an incorrect WildCLID constant specified in the line indicated by the Script Validator. Make sure that the WildCLID constant is 1–32 digits in length.
Error number	22
Message	CDN or DNIS range is 10 digits.
Meaning	Ensure that the CDN or DNIS number on the line indicated by the Script Validator is exactly10 digits in length.
Error number	23
Message	Agent IDs can be maximum 4 digits, skillsets 30.
Meaning	There is an invalid agent ID or skillset specified in the line indicated by the Script Validator. Make sure that the agent is not longer than 4 digits, or the skillset is not longer than 30 alphanumeric characters.
Error number	24
Message	Max day of month is 31.
Meaning	There is an invalid day of month specified in the line indicated by the Script Validator. Make sure that the day of month is 1–31.
Error number	25
Message	Time of day ranges from 00:01 to 23:59.
Meaning	There is an invalid time of day specified in the line indicated by the Script Validator. Make sure that the time of day is between 00:01 and 23:59.
Error number	26
Message	Time format was incorrect (three digit hours, and so on).

Meaning	There is an invalid time format specified in the line indicated by the Script Validator. Make sure that you specify the time in the following format: hh:mm For example 05:30
Error number	27
Message	Valid music, RAN, Route number is 0–512.
Meaning	There is an invalid music, RAN, or route number specified in the line indicated by the Script Validator. Make sure that the number is between 0–512.
Error number	28
Message	DNs or CLIDs can be max 32 digits.
Meaning	There is an invalid DN or CLID number specified in the line indicated by the Script Validator. Make sure that the number is 1–32 digits in length.
Error number	31
Message	Valid priority 1 to 6.
Meaning	There is an invalid priority specified in the line indicated by the Script Validator. Make sure that the priority is between 1–6.
Error number	32
Message	IDs (variable names) can be 30 characters max.
Meaning	There is an invalid variable name specified in the line indicated by the Script Validator. Make sure that the variable name is between 1–30 characters in length.
Error number	33
Message	Incorrect format for WildCLID was used (Wildcards @/? in WildCLID were improperly used.

Meaning	There is an invalid WildCLID specified in the line indicated by the Script Validator. For information about using wildcard characters, see “Wildcards and placeholders” on page 264.
Error number	34
Message	Char not valid in language ({, |, and so on.) was encountered.
Meaning	There is an invalid character in the line indicated by the Script Validator. Remove invalid characters.
Error number	36
Message	Sets (lists and ranges) must contain elements of the same type, and variables cannot (currently) be mixed with constants, even if they are of the same type.
Meaning	The set of values in the line indicated by the Script Validator contains values of different types, or mixes constants with variables. Make sure that all of the values are the same type, and remove either the constants or the variables.
Error number	37
Message	The variable specified is not the expected type.
Meaning	The variable in the line indicated by the Script Validator is not the correct type. Use a different variable, or change the variable type. For more information, see Chapter 4, “Working with script variables.”
Error number	39
Message	An Execute statement references a non-existent label.
Meaning	There is an Execute statement in the line specified by the Script Validator that references a section that does not exist. Make sure that the section exists, and that it is referenced correctly (that is, exact spelling) by the Execute statement.
Error number	40
Message	A label is defined in more than one place.

Meaning	There is a section name that is repeated in the script. Rename one of the sections.
Error number	41
Message	In the Event Handler, no event can be “handled” more than once.
Meaning	An event is repeated in the Event Handler statement. Remove the repeated event.
Error number	42
Message	This statement cannot be the first statement in a script.
Meaning	There is an invalid first statement in the script. For a list of statements that cannot be used as the first statements in a script, see “First command rule” on page 41.
Error number	44
Message	A set (variable or constant) cannot be used in a prompt.
Meaning	You can only use a variable of class Item or a single constant number for a prompt. If you are using a variable for the prompt indicated (by line number) by the Script Validator, make sure its class is set to Item.
Error number	47
Message	Day in date is invalid (three digits, and so on).
Meaning	There is an invalid day specified in the date. Make sure that the correct day is specified. It should contain no more than 2 digits.Valid date formats are as follows: May 4, 16 July, Dec 13
Error number	49
Message	Invalid date for month specified.
Meaning	The date specified for the month in the line indicated by the Script Validator is invalid. For example, Feb. 30 and Nov. 31 are invalid. Make sure that you specify the correct date.

Error number	50
Message	Max 20 agents IDs in set (list).
Meaning	There are too many agent IDs in the list specified by the Script Validator (by line number). Make sure that the list contains no more than 20 agent IDs.
Error number	51
Meaning	Too many skillsets are listed for this command.
Meaning	Too many skillsets are listed in the line indicated. Skillset-related commands can use no more than 20 skillsets in the list of skillsets. If the command relates to a NSBR feature (Meridian 1 only), no more than ten skillsets can be listed.
Error number	52
Message	Invalid type in Unary Minus operation.
Meaning	A Unary Minus operation is an operation in which you change the sign of a mathematical expression using a minus sign (–) in front of the expression. (For example, ASSIGN –variable2 TO variable1.) The variable being negated must be type Integer or Seconds.
Error number	53
Message	Invalid type (left side) in Addition operation.
Meaning	The value on the left side of the Addition operation is invalid. The left and right sides must be type Integer or Seconds. For more information, see “Mathematical expressions” on page 280.
Error number	54
Message	Invalid type (right side) in Addition operation.
Meaning	The value on the right side of the Addition operation is invalid. The left and right sides must be type Integer or Seconds. For more information, see “Mathematical expressions” on page 280.
Error number	55

Message	Invalid type (left side) in Subtraction operation.
Meaning	The value on the left side of the Subtraction operation is invalid. The left and right sides must be type Integer or Seconds. For more information, see “Mathematical expressions” on page 280.
Error number	56
Message	Invalid type (right side) in Subtraction operation.
Meaning	The value on the right side of the Subtraction operation is invalid. The left and right sides must be type Integer or Seconds. For more information, see “Mathematical expressions” on page 280.
Error number	57
Message	Invalid type (left side) in Multiplication operation.
Meaning	The value on the left side of the Multiplication operation is invalid. The left and right sides must be type Integer or Seconds. For more information, see “Mathematical expressions” on page 280.
Error number	58
Message	Invalid type (right side) in Multiplication operation.
Meaning	The value on the right side of the Multiplication operation is invalid. The left and right sides must be type Integer or Seconds. For more information, see “Mathematical expressions” on page 280.
Error number	59
Message	Invalid type (left side) in Division operation.
Meaning	The value on the left side of the Division operation is invalid. The left and right sides must be type Integer or Seconds. For more information, see “Mathematical expressions” on page 280.
Error number	60
Message	Invalid type (right side) in Division operation.

Meaning	The value on the right side of the Division operation is invalid. The left and right sides must be type Integer or Seconds. For more information, see “Mathematical expressions” on page 280.
Error number	61
Message	Sets (left side) are not allowed in Addition operations.
Meaning	A set (that is, more than one value) is included in the Addition operation. This is invalid. Ensure that there is only one value. If there is a variable on the left side of the equation, ensure that the class of the variable is set to Item.
Error number	62
Message	Sets (right side) are not allowed in Addition operations.
Meaning	A set (that is, more than one value) is included in the Addition operation. This is invalid. Ensure that there is only one value. If there is a variable on the right side of the equation, ensure that the class of the variable is set to Item.
Error number	80
Message	Incompatible types in Less Than or Equal To relational expression.
Meaning	The values in the Less Than or Equal To expression cannot be compared. Only types Integer and Seconds are allowed. For more information, see “Relational expressions” on page 282.
Error number	81
Message	Incompatible types in Greater Than relational expression.
Meaning	The values in the Greater Than expression cannot be compared. Only types Integer and Seconds are allowed. For more information, see “Relational expressions” on page 282.
Error number	82
Message	Incompatible types in Greater Than or Equal To relational expression.

Meaning	The values in the Greater Than or Equal To expression cannot be compared. Only types Integer and Seconds are allowed. For more information, see “Relational expressions” on page 282.
Error number	83
Message	Constants in range must be of same type.
Meaning	Both ends in the range are not the same type. (For example, Monday .. March is an invalid range because both ends of the range are not the same type.) Make sure that all of the values in the range are the same type.
Error number	84
Message	In a Where-Equals statement the Where expression must be of the same type as the value lists.
Meaning	Either the Where statement or one of the value statements in the Where-Equals command contains data of the wrong type. The Where value must be the same type as the values in the values list. For more information, see “Where-Equals” on page 182.
Error number	85
Message	The Where expression in the Where-Equals statement cannot be a set.
Meaning	The Where statement in the line indicated by the Script Validator contains invalid data. Make sure that the value in the Where statement is a single value only. Sets of values are not allowed. If you are using a variable, make sure that its class is set to Item.
Error number	86
Message	Sets cannot be used on the left side of the Equal To operator.
Meaning	The Equal To operator in the line indicated by the Script Validator contains invalid data. Make sure that the value on the left side of the operator is a single value only. Sets of values are not allowed. If you are using a variable, make sure that its class is set to Item.

Error number	87
Message	Sets cannot be used on the left side of the Not Equal To operator.
Meaning	The Not Equal To operator in the line indicated by the Script Validator contains invalid data. Make sure that the value on the left side of the operator is a single value only. Sets of values are not allowed.
Error number	88
Message	WildCLID constants are not allowed to be assigned to call variables in an Assign To command.
Meaning	The Assign To command in the line indicated by the Script Validator contains an invalid value. Do not use a WildCLID with the Assign To command.
Error number	89
Message	A set cannot be assigned to a call variable in an Assign To command.
Meaning	The Assign To command in the line indicated by the Script Validator contains an invalid value. Make sure that a single value only is assigned to the call variable. Sets of values are not allowed.
Error number	90
Message	Variable cannot be in a set.
Meaning	One of the following problems is indicated: • The variable class is set to Set, but the variable is of a type that is not allowed to be a set (for example, Boolean). • The command does not allow the use of a variable whose class is set to Set, or a constant list, or range.
Error number	91
Message	Unknown variable type.

Meaning	The variable in the line indicated by the Script Validator is not defined or is of an unknown type. Make sure that the variable has been created properly and is the correct type for the command. If this error still appears after revalidating the script, it can indicate a product problem. If this problem persists, you should report this error to your Nortel Networks customer support representative.
Error number	92
Message	Incompatible types in Assign To command.
Meaning	The values in the Assign To command in the line indicated by the Script Validator are not compatible. For more information, see “Assign To” on page 128.
Error number	95
Message	<MaxDigits> cannot be a set type.
Meaning	The maximum digits parameter must be a single value. Make sure that you have only one value (not a set of values) specified. If you are using a variable to indicate the number of digits, make sure that the class is set to Item.
Error number	97
Message	An undefined skillset is used.
Meaning	An invalid skillset is specified in the line indicated by the Script Validator. Make sure that the skillset exists, and is properly referenced in the script. If you are using a variable, make sure that it has been created, that it is type Skillset, and that its value is a valid skillset.
Error number	98
Message	An undefined agent ID is used.

Meaning One of the following problems has occurred:

- A variable in the line indicated by the Script Validator has not been defined. This variable is expected to be type Agent.
- No agent is defined with the agent login ID specified in the line indicated by the Script Validator.

Make sure that the agent exists and has the correct ID assigned. If you are using a variable, ensure that it has been defined correctly and is type Agent ID.

Error number 99

Message An undefined script is referenced in the Execute Script command.

Meaning There is an invalid script name in the Execute Script command in the line indicated by the Script Validator. Make sure that the script exists, and that it is properly referenced (that is, it uses correct spelling) in the Execute Script command.

Error number 100

Message An undefined identifier is encountered.

Meaning This error occurs when a name (or an “identifier”) is used in the script, but there is no skillset, variable, or script defined in the system with this name.

Error number 102

Message Illegal statement is used in Event statement.

Meaning There is an invalid statement in the Event Handler. For a list of valid statements that you can use in the Event Handler, see “Event Handler” on page 175.

Error number 106

Message Division by zero is not allowed.

Meaning There is an error in the division operation in the line indicated by the Script Validator. You cannot divide by zero.

Error number	107
Message	An Execute/Section label should not be the same as a variable name or a skillset name.
Meaning	Change the name of the Section label in the line indicated by the Script Validator so that it is not the same as a skillset or variable name. The system is case-insensitive, so the label names must differ by more than just the case of the letters.
Error number	108
Message	Only call variables can be used on the left side of the Assign To command.
Meaning	The variable used with the Assign To command in the line indicated by the Script Validator is a global variable. This is not allowed. You must delete the variable and create it again, this time choosing Call as the variable type. For more information, see “Creating script variables” on page 73.
Error number	109
Message	The variables used in Third Party statements (Send Info, Send Request, Get Response) cannot be sets.
Meaning	The variable class for the Send Info, Send Request, or Get Response variable should be set to Item. For more information, see “Changing script variable properties” on page 84.
Error number	111
Message	A Get Response statement must appear immediately after a Send Request statement.
Meaning	A Get Response statement is missing after the Send Request statement. For more information, see “Get Response” on page 200.
Error number	112
Message	No more than ten variables are allowed as parameters in each Third Party statement.

Meaning	There are too many variables listed in the Send Info, Send Request, or Get Response command in the line number indicated by the Script Validator. Remove as many variables as necessary.
Error number	113
Message	Third Party variables can be of the following types: CLID, CDN, DNIS, String, Integer.
Meaning	The variable in the line indicated by the Script Validator is the wrong type. Change the type to one of those listed above. For more information, see “Changing script variable properties” on page 84.
Error number	114
Message	Invalid use of keywords or punctuation.
Meaning	The line indicated by the Script Validator contains an invalid keyword or invalid punctuation. For the correct syntax of a command, use the Script Command Reference panel, or see Chapter 6, “Basic script commands,” and Chapter 7, “Advanced script commands. ”
Error number	115
Message	Missing parentheses.
Meaning	A parenthesis is missing in the line indicated by the Script Validator. Insert the parenthesis where appropriate.
Error number	116
Message	Only variables of Boolean type are allowed.
Meaning	The variable in the line indicated by the Script Validator should be a Boolean-type variable. For more information, see “Changing script variable properties” on page 84.
Error number	117
Message	Invalid Boolean value. Must be True/1 or False/0.

Meaning	In a Where-Equals command, if you are using a Boolean-type variable for the expression, the Value clauses must also be Boolean. However, the system also allows you to use a constant 1 or 0 to represent True or False. No other constants are allowed.
Error number	118
Message	Timer variable must be seconds type.
Meaning	The timer variable in the line indicated by the Script Validator should be a Seconds-type variable.
Error number	120
Message	Only variables of DN type are allowed. Variables cannot be sets.
Meaning	The variable type in the line indicated by the Script Validator is invalid. The variables indicated must be DN type. The class must be set to Item. For more information, see “Changing script variable properties” on page 84.
Error number	121
Message	Statement is part of the Advanced Script Elements package. Package not purchased.
Meaning	The script command in the line indicated by the Script Validator is an advanced command. You must purchase the Nortel Networks Symposium Call Center Server 200 application software package to use advanced commands.
Error number	122
Message	Advanced Scripts Package Query failed.
Meaning	This error can indicate a server installation problem. Try to revalidate the script. If the problem persists, contact your Nortel Networks customer support representative.
Error number	123
Message	No Send Request before Get Response.

Meaning	A Send Request statement is missing before the Get Response statement. For more information, see “Send Request” on page 197.
Error number	124
Message	Invalid value assigned to this variable. Check limit for variables of this type.
Meaning	The value assigned to the variable in the line indicated by the Script Validator is invalid. Change the value of the variable. For more information, see “Assigning values to variables” on page 76.
Error number	125
Message	Range of seconds is 0–65535
Meaning	An incorrect number of seconds is specified in the line indicated by the Script Validator. Make sure that the number of seconds you specify is between 0 and 65535.
Error number	126
Message	Treatment range is 1–10 digits
Meaning	There is an incorrect treatment number in the line indicated by the Script Validator. Make sure that the treatment number is 1–7 digits in length.
Error number	127
Message	ACD range is 2–10 digits
Meaning	There is an incorrect ACD number in the line indicated by the Script Validator. Make sure that the ACD is 2–7 digits in length.
Error number	128
Message	Variables used in this context must be boolean
Meaning	There is an invalid variable type in the line indicated by the Script Validator. Make sure that the variable is a Boolean-type (that is, has a True or False value) variable.

Error number	129
Message	Numbers cannot be greater than 32 digits.
Meaning	The number in the line indicated by the Script Validator is not valid. Make sure that it is no longer than 32 digits.
Error number	130
Message	Agent ID expression cannot be compared with constant ranges. Use variable instead.
Meaning	An expression or a statement with a return type of Agent ID cannot be compared with a list of constant Agent ID values or a range of Agent ID values. That is, IF LONGEST IDLE AGENT skillset1 = 1111, 2222 is not allowed. Use a set type variable instead.
Error number	131
Message	Statement is part of Host Data Exchange API feature. Feature not purchased or error accessing database.
Meaning	A third-party statement (SEND INFO, SEND REQUEST, GET RESPONSE) was encountered in the script. Either the feature has not been purchased by the customer or there was a database error accessing the features purchased.
Error number	137
Message	Call Center Manager feature not purchased or error accessing database.
Meaning	The package that includes the set of basic features for the system to work has not been purchased by the customer or there was a database error accessing the features purchased.
Error number	138
Message	Intrinsic not supported for ICM protocol.
Meaning	An intrinsic statement not supported for the ICM protocol was encountered in the script.

Error number	139
Message	Call Data index out of range. Valid range is 1 .. 10.
Meaning	The value of the Call Data intrinsic is incorrect. Ensure that the value is an integer between 1–10.
Error number	141
Message	Incomplete comment. Check that a previous comment was terminated properly.
Meaning	“/*” specifies the beginning of a comment and “*/” specifies the end of a comment. This error is logged when a comment is not terminated properly or another comment is started before ending a previous comment.
Error number	143
Message	A Wildcard constant/variable cannot be used on the LHS of an Equal comparison statement.
Meaning	In an IF Statement comparison, a wildcard (constant or a variable) cannot be on the left side of the “=”. i.e IF @23 = c_wildcard THEN... is not allowed.
Error number	144
Message	A Wildcard constant/variable cannot be used on the LHS of a Not Equal comparison statement.
Meaning	In an IF Statement comparison, a wildcard (constant or a variable) cannot be on the left side of the “<>” that is IF @23 <> c_wildcard THEN... is not allowed.
Error number	145
Message	A Wildcard constant/variable cannot be used in the <exp> in a WHERE <exp> EQUALS statement.
Meaning	In a WHERE Statement, a wildcard (constant or a variable) cannot be in the <exp> i.e WHERE c_wildcard EQUALS ... is not allowed.

Validation option rules

Introduction

This section of the appendix contains a list of errors that you might receive if you configure the Validation Options dialog box to inform you when you are breaking scriptwriting rules. It lists the rule numbers, provides the text of the error messages, and gives tips on how to avoid breaking rules.

Rule number	1
Message	Scripts cannot begin with this command.
Tip	Please choose a different command as the first command in the script.
Rule number	2
Message	Any statements following this command will be ignored.
Tip	Please remove all statements after this command.
Rule number	3
Message	This statement must be the first statement in the script
Tip	Please move this statement to the beginning of the script.
Rule number	4
Message	This clause should be included in WHERE...EQUALS element.
Tip	Please insert DEFAULT clause into WHERE...EQUALS element.
Rule number	5
Message	This command must not be followed by a GIVE BUSY or GIVE OVERFLOW command.
Tip	Please remove GIVE BUSY or GIVE OVERFLOW or insert another command before it.

Rule number	6
Message	Code structure tip
Tip	You can simplify this statement similar to the following: IF OUT OF SERVICE <skillset_1>, <skillset_2>, <skillset 3>,...THEN...
Rule number	8
Message	You should use the OUT OF SERVICE <skillset> intrinsic to test the skillset queue before the QUEUE TO SKILLSET command.
Tip	This can be done by inserting IF NOT OUT OF SERVICE <skillset> THEN before the QUEUE TO SKILLSET command.
Rule number	9
Message	Loops must contain at least one statement that suspends script execution for a finite time, such as WAIT or an announcement treatment
Tip	This can be done by inserting a WAIT command or an announcement treatment inside the loop.
Rule number	10
Message	Loops must have an exit so that they will not loop indefinitely if the call is not queued.
Tip	This can be done by inserting a test of the QUEUED intrinsic (except when using NACD) or AGE OF CALL intrinsic (with NACD), and jump out of the loop if the call is not queued.
Rule number	11
Message	There must be a WAIT command immediately following a QUEUE TO commands.
Tip	Please insert WAIT 2 (minimum) after the QUEUE TO command.
Rule number	12

Message	After a QUEUE TO SKILLSET command has been executed, it is not necessary to execute it again for the same skillset, unless the next QUEUE TO SKILLSET command follows an IF NOT QUEUED THEN statement.
Tip	Please remove the duplicate QUEUE TO SKILLSET command or insert an IF NOT QUEUED THEN statement before it.
Rule number	13
Message	Once a QUEUE TO AGENT has been executed, there is no need to execute it again for the same agent, unless the second (or next) QUEUE TO AGENT follows an IF NOT QUEUED THEN statement.
Tip	Please remove the duplicate QUEUE TO AGENT command or insert an IF NOT QUEUED THEN statement before it.

Appendix B

Scripting keywords

In this appendix

List of scripting keywords

Introduction

The following keywords are reserved words within the Symposium Call Center Server's scripting language. Do not use them as user-defined names or labels.

ABANDON
ACD
ACTIVE
AGE
AGENT
AGENTS
AHEAD
AND
ANNOUNCEMENT
ANSWER
ANSWERED
APPLICATION
APR
APRIL
ASSIGN
ASSIGNED
AUG
AUGUST
AVERAGE
BROADCAST
BUSY
BY
CALL
CALLS
CDN
CHANGE
CHARACTER
CHINESE
CLID
COLLECT
CONFERENCED
CONSULTED
CONTINUOUS
CONTROLLED
COUNT
DATA
DATE
DAY
DEC
DECEMBER
DEFAULT
DEQUEUE
DIALED
DIGIT
DIGITS
DIRECT
DISCONNECT
DISTURB
DN
DNIS
DO
ELSE
END
ENGLISH
EQUALS
EVENT
EXECUTE
EXPECTED
FAIL
FALSE
FEB
FEBRUARY
FOR
FOREVER
FORWARD
FRENCH
FRIDAY
FROM
GERMAN

GET
GIVE
HANDLER
HOLD
ID
IDLE
IF
IN
INCOMING
INFO
INTER
INTERRUPTIBLE
INTERNATIONAL
INTO
IVR
JAN
JANUARY
JAPANESE
JUL
JULY
JUN
JUNE
LANGUAGE
LEAST
LOC
LOCATION
LOG
LOGGED
LONGEST
MAR
MARCH
MAY
MONDAY
MONTH
MOST
MUSIC
NACD
NAME
NETWORK
NIGHT
NO
NON
NONCONTROLLED
NOT
NOV
NOVEMBER
NPA
NPANXX
NUMBER
NUMBERBYDIGIT
NXX
OCT
OCTOBER
OF
OLDEST
ON
OPEN
OR
OUT
OVERFLOW
PLAY
PORTUGUESE
POSITION
PRESENT
PRIORITY
PROMPT
QUEUE
QUEUED
QUEUES
QUIT
RAN
RATE
RECEIVED
RELEASED
REMOVE
REQUEST
RESPONSE
RETRIEVE
RETURN
RINGBACK
RINGING
ROUTE
SATURDAY
SCRIPT
SECTION
SEGMENT
SEND
SEP
SEPTEMBER
SERVICE

SESSION
SET
SILENCE
SKILLSET
SPANISH
SPEED
SUCCESS
SUNDAY
TERMINATING
THEN
THURSDAY
TIME
TIMEOUT
TIMER
TO
TOTAL
TRANSFERRED
TREATMENT
TRUE
TUESDAY
TYPE
VALUE
VOICE
WAIT
WAITING
WEDNESDAY
WEEK
WHERE
WHILE
WITH
YEAR

Glossary

A

accelerator key

A key on a phoneset that an agent can use to place a call quickly. When an agent presses an accelerator key, the system places the call to the configured number associated with the key. For example, if an agent presses the Emergency key, the system places a call to the agent’s supervisor.

access class

A collection of access levels that defines the actions a member of the access class can perform within the system. For example, a member of the Administrator access class might be given a collection of Read/Write access levels.

access level

A level of access or permission given to a particular user for a particular application or function. For example, a user might be given View Only access to historical reports.

ACD call

See Automatic call distribution call.

ACD-DN

See Automatic call distribution directory number.

ACD group

See Automatic call distribution group.

ACD routing table

See Automatic call distribution routing table.

ACD subgroup

See Automatic call distribution subgroup.

acquired resource

A resource configured on the switch that is under the control of the Symposium Call Center Server. Resources must be configured with matching values on both the switch and the Symposium Call Center Server.

activated script

A script that is processing calls or is ready to process calls. Before you can activate a script, you must first validate it.

activity code

A number that an agent enters on his or her phoneset during a call. Activity codes provide a way of tracking the time agents spend on various types of incoming calls.They are also known as Line of Business (LOB) codes. For example, the activity code 720 might be used to track sales calls. Agents can then enter 720 on their phonesets during sales calls, and this information can be generated in an Activity Code report.

administrator

A user who is responsible for maintaining the Symposium Call Center Server.

agent

A user who is responsible for handling customer calls.

agent login ID

A unique identification number assigned to a particular agent. The agent uses this number when logging in. The agent ID is not associated with any particular phoneset.

agent to skillset assignment

A matrix that, when you run it, sets the priority of one or more agents for a skillset. Agent to skillset assignments can be scheduled.

agent to supervisor assignment

A definition that, when you run it, assigns one or more agents to specific supervisors. Agent to supervisor assignments can be scheduled.

application

1. A logical entity that represents a Symposium Call Center Server script for reporting purposes. The master script and each primary script have an associated application.The application has the same name as the script it represents. 2. A program that runs on a computer.

application program interface

A set of routines, protocols, and tools that programmers use to develop software applications. APIs simplify the development process by providing commonly used programming procedures.

associated supervisor

A supervisor who is available for an agent if the agent’s reporting supervisor is unavailable. *See also* reporting supervisor.

Automatic call distribution call

A call to an ACD-DN. ACD calls are distributed to agents in an ACD group based on the ACD routing table on the switch.

Automatic call distribution directory number

Primary and supplementary DNs associated with an ACD group. Calls made to these DNs are distributed to agents belonging to the group, based on the ACD routing table on the switch.

Automatic call distribution group

An entity defined on the switch for the purpose of call distribution. When a customer dials an ACD group, the call is routed to any agent who is a member of that group.

Automatic call distribution routing table

A table configured on the switch that contains a list of ACD-DNs used to define routes for incoming calls. This ensures that incoming calls not processed by Symposium Call Center Server will be queued to ACD groups and handled by available agents.

Automatic call distribution subgroup

An entity defined on the switch to assign supervisory responsibilities. Each subgroup has one supervisor phoneset and a number of agent phonesets associated with it. Agents can log on to any phoneset within their ACD subgroup. The supervisor must log on to the supervisor phoneset to monitor his or her assigned agents.

C

call age

The amount of time a call was waiting in the system before being answered by an agent.

call intrinsic

A script element that stores call-related information assigned when a call enters the Symposium Call Center Server. *See also* intrinsic, skillset intrinsic, time intrinsic, and traffic intrinsic.

call presentation class

A collection of preferences that determines how calls are presented to an agent.

call priority

A numerical value assigned in a script that defines the relative importance of a call. If two calls are in the queue when an agent becomes available, and one call is queued with a higher priority than the other, the agent receives the higher priority call first. *See also* skillset priority.

call treatment

A script element that enables you to provide handling to a call while it is waiting to be answered by a call center agent. For example, a caller can hear a recorded announcement or music while waiting for an agent.

call variable

A script variable that applies to a specific call. A call variable follows the call through the system and is passed from one script to another with the call. *See also* global variable, variable.

Calling Line Identification

This is an optional service that identifies the telephone number of the caller. This information can then be used to route the call to the appropriate agent or skillset. The CLID can also be displayed on an agent’s phoneset.

CDN

See controlled directory number.

CLAN

See Customer local area network.

CLID

See Calling Line Identification.

client

The part of Symposium Call Center Server that runs on a personal computer or workstation and relies on the server to perform some operations. *See also* server.

command

A building block used with expressions, variables, and intrinsics to create scripts. Commands perform distinct functions, such as routing a call to a specific destination, playing music to a caller, or disconnecting a caller.

controlled directory number

A special directory number that allows calls arriving at the switch to be queued when the CDN is controlled by an application such as Symposium Call Center Server. When a call arrives at this number, the switch notifies the application and waits for routing instructions, which are performed by scripts in Symposium Call Center Server.

Customer local area network

The LAN to which your corporate services and resources connect. The Symposium Call Center Server and client both connect to the CLAN. Third-party applications that interface with the server also connect to this LAN.

D

DBMS

Database Management System

deactivated script

A script that does not process any new calls. If a script is in use when it is deactivated, calls continue to be processed by the script until they are completed.

default activity code

The activity code that is assigned to a call if an agent does not enter an activity code manually, or when an agent presses the activity code button twice on his or her phoneset.

default skillset

The skillset to which calls are queued if they have not been queued to a skillset or a specific agent by the end of a script.

desktop user

A configured user who can log on to the Symposium Call Center Server from a client PC.

DHCP

See dynamic host configuration protocol.

Dial-Up Networking

See Remote Access Services.

Dialed Number Identification Service

An optional service that allows Symposium Call Center Server to identify the phone number dialed by the incoming caller.

directory number

The number that identifies a phoneset on a switch. The directory number (DN) can be a local extension (local DN), a public network telephone number, or an automatic call distribution directory number (ACD-DN).

directory number call

A call that is presented to the DN key on an agent’s phoneset.

display threshold

A threshold used in real-time displays to highlight a value below or above the normal range.

DMS

Digital Multiplex Switch.

DN

See directory number.

DN call

See directory number call.

DNIS

See Dialed Number Identification Service.

dongle

The attachment plugged into the parallel port of a server connected to a DMS/MSL-100 switch that authenticates the serial number required at the time of server installation.

dynamic host configuration protocol

A protocol for dynamically assigning IP addresses to devices on a network.

dynamic link library

A library of executable functions or data that can be used by a Windows application. Typically, a DLL provides one or more particular functions and a program accesses the functions by creating either a static or dynamic link to the DLL. A DLL can be used by several applications at the same time.

E

ELAN

See embedded local area network.

embedded local area network

A dedicated Ethernet TCP/IP LAN that connects the Symposium Call Center Server and the switch.

Emergency key

A key on an agent's phoneset that, when pressed by an agent, automatically calls his or her supervisor to notify the supervisor of a problem with a caller.

event

1. An occurrence or action on the Symposium Call Center Server, such as the sending or receiving of a message, the opening or closing of an application, or the reporting of an error. Some events are for information only, while others can indicate a problem. Events are categorized by severity: information, minor, major, and critical. 2. An action generated by a script command, such as queuing a call to a skillset or playing music.

expression

A building block used in scripts to test for conditions, perform calculations, or compare values within scripts. *See also* logical expression, mathematical expression, and relational expression.

F

first-level threshold

The value that represents the lowest value of the normal range for a statistic in a threshold class. The system tracks how often the value for the statistic falls outside this value.

G

global settings

Settings that apply to all skillsets that are configured on your system.

global variable

A variable that contains values that can be used by any script on the system. The value of a global variable can only be changed in the Script Variable Properties sheet. It cannot be changed in a script. *See also* call variable, variable.

I

ICM

See Intelligent Call Manager.

Incalls key

The key on an agent phoneset to which incoming ACD and Symposium Call Center Server calls are presented.

Intelligent Call Manager

A high capacity call center TCP/IP interface to the switch that enables the exchange of messages between the switch and a remote host computer.

Internet Protocol address

An identifier for a computer or device on a TCP/IP network. Networks use the TCP/IP protocol to route messages based on the IP address of the destination. The format of an IP address is a 32-bit numeric address written as four values separated by periods. Each value can be 0 to 255. For example, 1.160.10.240 could be an IP address.

intrinsic

A word or phrase used in a script to gain access to system information about skillsets, agents, time, and call traffic that can then be used in formulas and decision-making statements. *See also* call intrinsic, skillset intrinsic, time intrinsic, and traffic intrinsic.

IP address

See Internet Protocol address.

L

LAN

See Local area network.

Line of Business code

See activity code.

LOB code

See activity code.

Local area network

A computer network that spans a relatively small area. Most LANs connect workstations and personal computers and are confined to a single building or group of buildings.

logical expression

A symbol used in scripts to test for different conditions. Logical expressions are AND, OR, and NOT. *See also* expression, mathematical expression, and relational expression.

M

master script

The first script executed when a call arrives at the Symposium Call Center Server. A default master script is provided with Symposium Call Center Server, but it can be customized by an authorized user. It can be deactivated but not deleted. *See also* primary script, script, and secondary script.

mathematical expression

An expression used in scripts to add, subtract, multiply, and divide values. Mathematical expressions are addition (+), subtraction (-), division (/), and multiplication (*). *See also* expression, logical expression, and relational expression.

music route

A resource installed on the switch that provides music to callers while they wait for an agent.

N

night mode

A skillset state in which the server does not queue incoming calls to the skillset, and in which all queued calls are given night treatment. A skillset goes into night mode automatically when the last agent logs off, or the administrator can put it into night mode manually. *See also* out-of-service mode, transition mode.

NPA

See Number Plan Area.

Number Plan Area

Area code

O

object linking and embedding

A compound document standard that enables you to create objects with one application and then link or embed them in a second application.

ODBC

See Open Database Connectivity.

OEM

Original equipment manufacturer

OLE

See object linking and embedding.

Open Database Connectivity

A Microsoft-defined database application program interface (API) standard.

out-of-service mode

A skillset state in which the skillset does not take calls. A skillset is out of service if there are no agents logged on or if the supervisor puts the skillset into out-of-service mode manually. *See also* night mode, transition mode.

out-of-service skillset

A skillset that is not taking any new calls. While a skillset is out of service, incoming calls cannot be queued to the skillset. *See also* skillset.

P

pegging

The action of incrementing statistical counters to track and report on system events.

pegging threshold

A threshold used to define a cut-off value for statistics such as short call and service level. Pegging thresholds are used in reports.

PEP

See Performance Enhancement Package.

Performance Enhancement Package

A Symposium Call Center Server supplementary software application that enhances the functionality of previously released software by improving performance, adding functionality, or correcting a problem discovered since the original release.

phoneset

The physical device, connected to the switch, to which calls are presented. Each agent and supervisor must have a phoneset.

phoneset display

The display area on an agent’s phoneset where information about incoming calls can be communicated.

Position ID

1. A unique identifier for a phoneset, used by the switch to route calls to the phoneset. 2. Referred to as Telephony/Port Address in Symposium Call Center Server.

primary ACD-DN

A directory number that callers can dial to reach an ACD group.

primary script

A script that is executed or referenced by the master script. A primary script can route calls to skillsets, or it can transfer routing control to a secondary script. *See also* master script, script, and secondary script.

R

RAN

recorded announcement

RAN route

See recorded announcement route.

RAS

See Remote Access Services.

recorded announcement route

A resource installed on the switch that offers a recorded announcement to callers.

relational expression

An expression used in scripts to test for different conditions. Relational expressions are less than (<), greater than (>), less than or equal to (< =), greater than or equal to (> =), and not equal to (< >). *See also* expression, logical expression, and mathematical expression.

Remote Access Services

A feature built into Windows NT and Windows 95 that enables users to log on to an NT-based LAN using a modem, X.25 connection, or WAN link. This feature is also known as Dial-Up Networking.

reporting supervisor

The supervisor who has primary responsibility for an agent. When an agent presses the Emergency key on the phoneset, the emergency call is presented to the agent's reporting supervisor. *See also* associated supervisor.

S

sample script

A script that is installed with the Symposium Call Center Server client. Sample scripts are stored as text files in a special folder on the client. The contents of these scripts can be imported or copied into user scripts to create scripts for typical call center scenarios.

SCM

See Service Control Manager.

script

A set of instructions that relates to a particular type of call, caller, or set of conditions, such as time of day or day of week. *See also* master script, primary script, and secondary script.

script variable

See variable.

second-level threshold

The value used in display thresholds that represents the highest value of the normal range for a given statistic. The system tracks how often the value for the statistic falls outside this value.

secondary directory number

A DN defined on the agent’s phoneset as a Centrex line for incoming and outgoing non-ACD calls.

secondary script

Any script (other than a master or primary script) that is referenced from a primary script or any other secondary script. There is no pegging of statistics for actions occurring during a secondary script. *See also* master script, primary script, and script.

server

A computer or device on a network that manages network resources. Examples of servers include file servers, print servers, network servers, and database servers. The Symposium Call Center Server is used to configure the operations of the call center. *See also* client.

service

A process that adheres to a Windows NT structure and requirements. A service provides system functionality.

Service Control Manager

A Windows NT process that manages the different services on the PC.

service level

The percentage of incoming calls answered within a configured number of seconds.

service level threshold

A parameter that defines the number of seconds within which incoming calls should be answered.

Simple Network Management Protocol

A set of protocols for managing complex networks. SNMP works by sending messages, called protocol data units (PDUs), to different parts of a network and then analyzing the responses.

site

A system using Symposium Call Center Server that can be accessed using SMI.

skillset

A group of capabilities or knowledge required to answer a specific type of call.

skillset intrinsic

A script element that inserts information about a skillset in a script. Skillset intrinsics return values such as skillsets, integers, and agent IDs. These values are then used in queuing commands. *See also* call intrinsic, intrinsic, time intrinsic, and traffic intrinsic.

skillset priority

An attribute of a skillset assignment that determines the order in which calls from different skillsets are presented to an agent. When an agent becomes available, calls might be waiting for several of the skillsets to which the agent belongs. The server presents the call queued for the skillset for which the agent has the highest priority.

standby

In skillset assignments, a property that grants an agent membership in a skillset, but makes the agent inactive for that skillset.

supervisor

A user who manages a group of agents. *See also* associated supervisor, reporting supervisor.

supplementary ACD-DN

A DN associated with a primary DN. Any calls to the supplementary DN are automatically routed to the primary DN. A supplementary DN can be a toll-free (1-800) number.

switch

The hardware that receives incoming calls and routes them to their destination.

switch resource

A device that is configured on the switch. For example, a CDN is configured on the switch, and then is used as a resource with Symposium Call Center Server. *See also* acquired resource.

Symposium Call Center Server call

A call to a CDN that is controlled by the Symposium Call Center Server. The call is presented to the Incalls key on an agent’s phoneset.

system-defined script

The Master_Script. This script can be customized or deactivated by a user, but cannot be deleted. This script is the first script executed for every call arriving at the call center.

T

TCP/IP

See Transport Control Protocol/Internet Protocol.

telephony

The science of translating sound into electrical signals, transmitting them, and then converting them back to sound. The term is used frequently to refer to computer hardware and software that perform functions traditionally performed by telephone equipment.

threshold

A value for a statistic at which system handling of the statistic changes.

threshold class

A set of options that specifies how statistics are treated in reports and real-time displays. *See also* display threshold, pegging threshold.

time intrinsic

A script element that stores information about system time, including time of day, day of week, and week of year. *See also* call intrinsic, intrinsic, skillset intrinsic, and traffic intrinsic.

Token Ring

A PC network protocol developed by IBM. A Token Ring network is a type of computer network in which all the computers are arranged schematically in a circle.

traffic intrinsic

An intrinsic that inserts information about system-level traffic in a script. *See also* call intrinsic, intrinsic, skillset intrinsic, and time intrinsic.

transition mode

A skillset state in which the server presents already queued calls to a skillset. New calls queued to the skillset are given out-of-service treatment. *See also* night mode, out-of-service mode.

Transport Control Protocol/Internet Protocol

The communication protocol used to connect devices on the Internet. TCP/IP is the standard protocol for transmitting data over networks.

treatment

See call treatment.

U

user-created script

A script that is created by an authorized user on the Symposium Call Center Server system. Primary and secondary scripts are user-created scripts.

user-defined script

A script that is modified by an authorized user on the Symposium Call Center Server system.

utility

A program that performs a specific task, usually related to managing system resources. Operating systems contain a number of utilities for managing disk drives, printers, and other devices.

V

validation

The process of checking a script to ensure that all the syntax and semantics are correct. A script must be validated before it can be activated.

variable

A placeholder for values calculated within a script, such as CLID. Variables are defined in the Script Variable Properties sheet and can be used in multiple scripts to determine treatment and routing of calls entering the Symposium Call Center Server. *See also* call variable, global variable.

W

WAN

See Wide area network.

Wide area network

A computer network that spans a relatively large geographical area. Typically, a WAN consists of two or more local area networks (LANs). The largest WAN in existence is the Internet.

workload scenarios

Sets of configuration values defined for typical patterns of system operations. Five typical workload scenarios (entry, small, medium, large, and upper end) are used in the Capacity Assessment Tool for capacity analysis for the Symposium Call Center Server.

Index

NORTEL NETWORKS™

How the world shares ideas.

Reader Response Form

Nortel Networks Symposium Call Center Server Product release 3.0
Scripting Guide
P0911816

Tell us about yourself:

Name: ____________________

Company: ____________________

Address: ____________________

____

Occupation: ____________________ **Phone:** ____________________

1. What is your level of experience with this product?

   □ New user □ Intermediate □ Experienced □ Programmer

2. How do you use this book?

   □ Learning □ Procedural □ Reference □ Problem solving

3. Did this book meet your needs?

   □ Yes □ No

   If you answered No to this question, please answer the following questions.

4. What chapters, sections, or procedures did you find hard to understand?

   ____

   ____

   ____

5. What information (if any) was missing from this book?

   ____

   ____

   ____

6. How could we improve this book?

   ____

   ____

   ____

Please return your comments by fax to (416) 597-7104, or mail your comments to
Toronto Information Products, Nortel Networks, 522 University Avenue, 14th Floor, Toronto, ON,
Canada, M5G 1W7.

NORTEL NETWORKS™

How the world shares ideas.

Reader Response Form

Nortel Networks Symposium Call Center Server

for the DMS/MSL-100

Scripting Guide

Toronto Information Products
Nortel Networks
522 University Avenue, 14th Floor
Toronto, Ontario, Canada
M5G 1W7

Information is subject to change without notice. Nortel Networks reserves the right to make changes in design or components as progress in engineering and manufacturing may warrant.

The process of transmitting data and call messaging between the DMS/MSL-100 and the Nortel Networks Symposium Call Center Server is proprietary to Nortel Networks. Any other use of the data and the transmission process is a violation of the user license unless specifically authorized in writing by Nortel Networks prior to such use. Violations of the license by alternative usage of any portion of this process or the related hardware constitutes grounds for an immediate termination of the license and Nortel Networks reserves the right to seek all allowable remedies for such breach.

Publication number: P0911816
Product release: 3.0
Document release: Standard 1.0
Date: April 2000

Printed in the United States of America

NORTEL NETWORKS™

How the world shares ideas.