Lucent Technologies
Bell Labs Innovations

DEFINITY®
Enterprise Communications Server

Release 7
Call Vectoring/Expert Agent Selection
(EAS) Guide

555-230-521
Comcode 108485947
Issue 3
June 1999

Copyright © 1999, Lucent Technologies
All Rights Reserved
Printed in U.S.A.

Notice
Every effort was made to ensure that the information in this book was complete and accurate at the time of printing. However, information is subject to change.

Your Responsibility for Your System's Security
Toll fraud is the unauthorized use of your telecommunications system by an unauthorized party, for example, persons other than your company's employees, agents, subcontractors, or persons working on your company's behalf. Note that there may be a risk of toll fraud associated with your telecommunications system and, if toll fraud occurs, it can result in substantial additional charges for your telecommunications services.

You and your system manager are responsible for the security of your system, such as programming and configuring your equipment to prevent unauthorized use. The system manager is also responsible for reading all installation, instruction, and system administration documents provided with this product in order to fully understand the features that can introduce risk of toll fraud and the steps that can be taken to reduce that risk. Lucent Technologies does not warrant that this product is immune from or will prevent unauthorized use of common-carrier telecommunication services or facilities accessed through or connected to it. Lucent Technologies will not be responsible for any charges that result from such unauthorized use.

Lucent Technologies Fraud Intervention
If you *suspect that you are being victimized* by toll fraud and you need technical support or assistance, call Technical Service Center Toll Fraud Intervention Hotline at 1 800 643-2353.

Federal Communications Commission Statement
Part 15: Class A Statement. This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio-frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at his own expense.

Part 68: Network Registration Number. This equipment is registered with the FCC in accordance with Part 68 of the FCC Rules. It is identified by FCC registration number AS593M-13283-MF-E.

Part 68: Answer-Supervision Signaling. Allowing this equipment to be operated in a manner that does not provide proper answer-supervision signaling is in violation of Part 68 Rules. This equipment returns answer-supervision signals to the public switched network when:

- Answered by the called station
- Answered by the attendant
- Routed to a recorded announcement that can be administered by the CPE user

This equipment returns answer-supervision signals on all DID calls forwarded back to the public switched telephone network. Permissible exceptions are:

- A call is unanswered
- A busy tone is received
- A reorder tone is received

Canadian Department of Communications (DOC) Interference Information
This digital apparatus does not exceed the Class A limits for radio noise emissions set out in the radio interference regulations of the Canadian Department of Communications.

Le Présent Appareil Nomérique n'émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de la class A préscrites dans le reglement sur le brouillage radioélectrique édicté par le ministére des Communications du Canada.

Trademarks
See the preface of this document.

Ordering Information
Call: Lucent Technologies BCS Publications Center
Voice 1 800 457-1235 International Voice 317 322-6416
Fax 1 800 457-1764 International Fax 317 322-6699
Write: Lucent Technologies BCS Publications Center
2855 N. Franklin Road
Indianapolis, IN 46219
Order: Document No. 555-230-521
Comcode 108485947
Issue 3, June 1999
For additional documents, refer to the section in "About This Document" entitled "Related Resources."

You can be placed on a standing order list for this and other documents you may need. Standing order will enable you to automatically receive updated versions of individual documents or document sets, billed to account information that you provide. For more information on standing orders, or to be put on a list to receive future issues of this document, contact the Lucent Technologies Publications Center.

European Union Declaration of Conformity
The "CE" mark affixed to the DEFINITY® equipment described in this book indicates that the equipment conforms to the following European Union (EU) Directives:

- Electromagnetic Compatibility (89/336/EEC)
- Low Voltage (73/23/EEC)
- Telecommunications Terminal Equipment (TTE) i-CTR3 BRI and i-CTR4 PRI

For more information on standards compliance, contact your local distributor.

Comments
To comment on this document, return the comment card at the front of the document.

Acknowledgment
This document was prepared by Product Documentation Development, Lucent Technologies, Denver, CO.

Contents

Contents **iii**

About This Document **xix**

- Feature Availability xix
- What is Call Vectoring? xix
- What is Expert Agent Selection (EAS)? xxii
- Call Vectoring Features xxiii
- Contents and Organization of the Guide xxv
- Intended Audience and Use of the Guide xxv
- References xxvi

1 Tutorial **1-1**

- Introduction 1-1
- Entering the Vector On-Line 1-1
  - Procedures for Basic Screen Administration 1-1
- Constructing a Vector: One Approach 1-6
  - Phase 1: Queuing a Call to the Main Split 1-7
  - Phase 2: Providing Feedback and Delay Announcement 1-8
  - Phase 3: Repeating Delay Announcement and Feedback 1-9
  - Phase 4: Queuing a Call to a Backup Split 1-11
  - Phase 5: Checking the Queue Capacity 1-12
  - Phase 6: Checking for Non-Business Hours 1-13
- Benefits of Call Vectoring 1-15

2 Fundamentals of Call Vectoring **2-1**

- Introduction 2-1
- Call Management 2-1
  - Call Flow 2-2
  - Caller Control 2-3
  - Call Queuing to Splits 2-4
  - Agent Work Mode 2-5
  - Calling Party Feedback 2-6
  - Dialed Number Identification Service (DNIS) 2-7
- Vector Processing 2-8
  - Vector Directory Number 2-8

Vector Control Flow 2-13
Programming Capabilities 2-15

3 Basic Call Vectoring 3-1

- Introduction 3-1
- Command Set 3-1
- Functions and Examples 3-2
  - Providing Call Treatments 3-2
  - Routing Calls 3-13
  - Branching/Programming 3-20
- Vector Chaining 3-22

4 Call Prompting 4-1

- Introduction 4-1
- Command Set 4-2
- Touch-Tone Collection Requirements 4-2
- Call Prompting Digit Entry 4-3
  - Removing Incorrect Digit Strings 4-4
  - Entering Variable-Length Digit Strings 4-4
  - Entering Dial-Ahead Digits 4-5
- Functions and Examples 4-5
  - Treating Digits as a Destination 4-6
  - Using Digits to Collect Branching Information 4-7
  - Using Digits to Select Options 4-10
  - Displaying Digits on the Agent's Set 4-10
  - Passing Digits to an Adjunct 4-12
  - Creating Service Observing Vectors 4-13
- Dial-Ahead Digits 4-14
- ASAI-Requested Digit Collection 4-18
- ASAI-Provided Dial-Ahead Digits 4-19

5 Advanced Vector Routing 5-1

- Introduction 5-1
- Command Set 5-1
- Expected Wait Time (EWT) 5-2
  - EWT for a Split 5-2
  - EWT for a Call 5-3
  - Passing EWT to a VRU 5-3

The EWT Algorithm 5-4
When to Use Wait Time Predictions 5-4
Examples 5-6
Factors that Affect the Value of EWT 5-9
- Simple Troubleshooting for EWT 5-10
- Rolling Average Speed of Answer (ASA) 5-10
Rolling ASA Split Calculation 5-10
Rolling ASA VDN Calculation 5-11
Rolling ASA Considerations 5-11
Example 5-12
- VDN Calls 5-12
Counted Calls 5-13
Example 5-13

6 ANI /II-Digits Routing and Caller Information Forwarding (CINFO) 6-1
- Introduction 6-1
- Command Sets 6-1
- ANI Routing 6-2
ANI Routing Example 6-3
Vector Routing Tables with ANI 6-4
- II-Digits Routing 6-5
II-Digits Routing Example 6-7
- Caller Information Forwarding 6-8
Detailed Operation 6-8
CINFO Vector Example 6-10
CINFO Interactions 6-10

7 Look-Ahead Interflow 7-1
- Before You Start 7-1
- A Quick Example 7-2
- What is Look-Ahead Interflow? 7-3
- How Traditional Look-Ahead Interflow Works 7-5
Example 1: Traditional LAI 7-7
Receiving Switch Operation 7-7
- How Enhanced Look-Ahead Interflow Works 7-9
The simple way to achieve FIFO 7-9

Detailed information about the new conditional 7-9
When does a call not interflow? 7-11
How is the minimum EWT set? 7-12
Example 2: Single-Queue Multi-Site Operation 7-13
Example 3: Maintaining FIFO Processing with LAI 7-14
Single-Queue FIFO Considerations 7-14
Example 4: LAI in a Tandem Switch Configuration 7-15
Sending Switch Operation 7-15
Tandem Switch Operation 7-15
Far End Switch Operation 7-16
- DNIS and VDN Override in an LAI Environment 7-17
  - Answering Agent's Display 7-17
  - Originator's Display 7-18
- LAI with Network ADR 7-18
- Multisite Applications for Enhanced LAI 7-20
- Look-Ahead Interflow Considerations 7-20
- Simple Troubleshooting for LAI 7-21

8 Best Service Routing™ 8-1

- An Overview of Best Service Routing 8-1
- What's in This Chapter? 8-1
- Benefits of Best Service Routing 8-2
- Before You Start 8-5
  - Switch Requirements 8-5
  - Network Requirements 8-6
- Terms to Know 8-7
- Single-Site BSR 8-10
  - The Elements of Single-Site BSR 8-10
  - How BSR Determines the Best Resource 8-12
  - Example 1: Basic Single-Site BSR 8-14
  - User Adjustments in Single-site BSR 8-16
  - Example 2: Single-Site BSR with Adjustments 8-17
- Simple Troubleshooting for Single-Site BSR 8-21
- Multi-site BSR 8-21
  - The Elements of Multi-site BSR Applications 8-22
  - Multi-site BSR Applications 8-26

Example 3: Multi-Site BSR with Two Switches 8-29
BSR Available Agent Strategies 8-34
More on Status Poll and Interflow Vectors 8-34
User Adjustments in Multi-site BSR 8-35
Example 4: Multi-site BSR with Limited Trunking 8-36
Example 5: Multi-site BSR with Slow Networks 8-41
Example 6: Handling Excessive Wait Times 8-45
- Simple Troubleshooting for Multi-Site BSR 8-45
- Planning and Administering Single-Site BSR 8-46
- Planning and Administering Multi-Site BSR 8-47
- Tips for Writing BSR Vectors 8-52

9 Information Forwarding 9-1
- Benefits of enhanced information forwarding 9-2
- Network requirements 9-3
- Enhanced Information Forwarding 9-4
  - Forwarding of call-related information 9-5
  - Forwarding collected digits with interflowed call 9-5
  - Forwarding accumulated in-VDN time 9-5
  - Transport via globally-supported methods 9-6
  - Providing LAI backward compatibility 9-6
- Determining User Information Needs 9-7
  - A Quick Example 9-9
- Simple Troubleshooting for Information Forwarding 9-10

10 Adjunct Routing 10-1
- Introduction 10-1
- Functions and Examples 10-1
  - Sending the Call Route Request 10-2
  - Awaiting the Response to the Call Route Request 10-6
  - Receiving and Implementing the Call Route 10-7
- Phantom Calls 10-8
  - How Do Phantom Calls Work? 10-8
  - How Are Phantom Calls Used? 10-9
  - How Do Phantom Calls Affect Call Vectoring? 10-9
  - Phantom Call Administration 10-10
- Single-Step Conference 10-10

How Does SSC Work With Call Vectoring? 10-11
- Multiple Outstanding Route Requests 10-11
User Scenarios 10-11

11 Expert Agent Selection 11-1
- Introduction 11-1
- Identifying Caller Needs 11-3
DNIS/ISDN Called Party 11-4
Call Prompting/VRU Digits/CINFO Digits 11-4
Host Database Lookup 11-5
Direct Agent Calling 11-5
- Functions and Examples 11-5
Administering Skills 11-5
Delivering the Call to the Skill Queue 11-13
Routing the Call to an Agent 11-18
- Interactions Involving EAS 11-24
Feature Interactions 11-24
Adjunct Interactions 11-30
- Listing Agents Logged into a Split or Skill 11-34
- Other Forms that Support EAS Agent LoginID 11-35
- Upgrading to a DEFINITY ECS EAS Environment 11-38

12 Call Vectoring Applications 12-1
- Introduction 12-1
- Customer Service Center 12-2
- Automated Attendant 12-4
- DIVA and Data/Message Collection 12-5
- Distributed Call Centers 12-8
- Help Desk 12-10
- Insurance Agency/Service Agency 12-11
- Warranty Service (with EAS) 12-15
- Resort Reservation Service (with EAS) 12-19
Placing the Reservation 12-19
Call-Back Provisions 12-21
- Vector Exercises 12-22
Exercise 1: Emergency and Routine Service 12-23
Exercise 2: Late Caller Treatment 12-25

Exercise 3: Messaging Option 12-27

A Call Vectoring Commands A-1

- Introduction A-1
- Command Description/Reference A-1
- Command/Option Summary A-3
- Command Job Aid A-6
- Command Directory A-16
- Adjunct Routing Command A-17
  - Purpose A-17
  - Syntax A-17
  - Valid Entries A-17
  - Requirements A-17
  - Example A-17
  - Operation A-17
  - Answer Supervision Considerations A-21
  - Feature Interactions A-21
  - CMS Interactions A-22
  - BCMS Interactions A-23
- Announcement Command A-24
  - Purpose A-24
  - Syntax A-24
  - Valid Entries A-24
  - Requirements A-24
  - Example A-24
  - Operation A-24
  - Answer Supervision Considerations A-25
  - Feature Interactions A-25
  - BCMS/CMS Interactions A-25
- Busy Command A-26
  - Purpose A-26
  - Syntax A-26
  - Requirements A-26
  - Operation A-26
  - Answer Supervision Considerations A-26
  - Feature Interactions A-26

CMS Interactions A-27
BCMS Interactions A-27
■ Check Command A-28
Purpose A-28
Syntax A-28
Valid Entries A-28
Requirements A-29
Examples A-29
Operation A-29
Answer Supervision Considerations A-31
Feature Interactions A-31
CMS Interactions A-31
BCMS Interactions A-33
■ Collect Digits Command A-34
Purpose A-34
Syntax A-34
Valid Entries A-34
Requirements A-34
Example A-34
Operation A-34
Answer Supervision Considerations A-38
Feature Interactions A-38
CMS/BCMS Interactions A-38
■ Consider Command A-39
Purpose A-39
Syntax A-39
Valid Entries A-39
Requirements A-40
Examples A-40
Operation A-40
Answer Supervision Considerations A-45
Feature Interactions A-45
CMS/BCMS Interactions A-45
■ Converse-on Command A-46
Purpose A-46

Syntax A-46
Valid Entries A-46
Requirements A-46
Examples A-47
Operation A-47
Answer Supervision Considerations A-50
Feature Interactions A-50
CMS Interactions A-57
BCMS Interactions A-57
■ Disconnect Command A-58
Purpose A-58
Syntax A-58
Valid Entries A-58
Requirements A-58
Example A-58
Operation A-58
Answer Supervision Considerations A-58
Feature Interactions A-59
CMS Interactions A-59
BCMS Interactions A-59
■ Goto Step Command A-60
Purpose A-60
Syntax A-60
Valid Entries A-62
Requirements A-63
Examples A-64
Operation A-65
Answer Supervision Considerations A-66
Feature Interactions A-66
CMS/BCMS Interactions A-66
■ Goto Vector Command A-67
Purpose A-67
Syntax A-67
Valid Entries A-69
Requirements A-70

Examples A-71
Operation A-71
Answer Supervision Considerations A-73
Feature Interactions A-73
CMS/BCMS Interactions A-73
■ Messaging Command A-74
Purpose A-74
Syntax A-74
Valid Entries A-74
Requirements A-74
Examples A-74
Operation A-74
Answer Supervision Considerations A-75
Feature Interactions A-76
CMS Interactions A-76
BCMS Interactions A-77
■ Queue-to Command A-78
Purpose A-78
Syntax A-78
Valid Entries A-78
Requirements A-78
Examples A-78
Operation A-78
Answer Supervision Considerations A-80
Feature Interactions A-80
CMS Interactions A-80
BCMS Interactions A-82
■ Reply-best A-83
Purpose A-83
Syntax A-83
Requirements A-83
Example A-83
Operation A-83
Answer Supervision Considerations A-84
CMS/BCMS Interactions A-84

- Route-to Command A-85
  - Purpose A-85
  - Syntax A-85
  - Valid Entries A-85
  - Requirements A-85
  - Examples A-86
  - Operation A-86
  - Answer Supervision Considerations A-88
  - Feature Interactions A-88
  - CMS Interactions A-91
  - BCMS Interactions A-93
- Stop Command A-94
  - Purpose A-94
  - Syntax A-94
  - Requirements A-94
  - Operation A-94
  - Answer Supervision Considerations A-94
  - Feature Interactions A-94
  - CMS Interactions A-95
- Wait-time Command A-96
  - Purpose A-96
  - Syntax A-96
  - Valid Entries A-96
  - Requirements A-97
  - Examples A-97
  - Operation A-97
  - Answer Supervision Considerations A-98
  - Feature Interactions A-98
  - CMS/BCMS Interactions A-99
- Criteria for Success/Failure of Call Vectoring Commands A-100

B Call Vectoring Management and Monitoring B-1

- Introduction B-1
- Implementation Requirements for the Call Vectoring Features B-1
- Enabling the Vector Disconnect Timer B-8

- Upgrading to a Call Vectoring Environment B-8
- Changing and Testing a Vector B-9
- Identifying Links to a Vector B-9
- Finding All Occurences of a Digit String B-10

C Considerations for the Vectoring Features C-1

- Introduction C-1
- Basic Call Vectoring Considerations C-1
- Call Prompting Considerations C-2
- Adjunct Routing Considerations C-3
- Transferring Calls to VDNs Considerations C-3
- VDN Return Destination Considerations C-4
  - User Scenario — Remote Access with Host Provided Security C-5
  - User Scenario — Saving in Trunk Facilities Between Call Centers C-7

D Advanced Multi-Site Routing D-1

- Introduction D-1
- Application Architecture in Multi-Site BSR D-1
- User Adjustments D-2
- Status Polling in BSR D-4
  - How Long do Status Polls Take? D-4
  - Intelligent Polling D-6
- Efficient Polling Patterns in Large Networks D-7
  - How Many Switches Should One Switch Poll? D-7
  - Which Remote Switches Should Each Switch Poll? D-8
- Considerations for Low Volume Splits and Skills D-10
  - Minimizing Variations in Wait Time D-11
- Advanced Information Forwarding D-14
  - Non-QSIG Protocol D-15
  - QSIG trunk group D-16
  - “Send Codeset 6/7 LAI IE” Option Interactions D-16

E Troubleshooting Vectors E-1

- Introduction E-1
- Unexpected Feature Operations E-1
- Unexpected Command Operations E-4

- Converse Command Debugging E-14
- Tracking Unexpected Vector Events E-16
  - Display Events Form E-17
  - Display Events Report E-18
  - Summary of Vector Events E-19
- Clearing Events E-32
- Tracing Call Flow E-32
- Tracing Expected Wait Time E-33

F Functional Differences for G2 and DEFINITY ECS Call Vectoring and EAS F-1

- Introduction F-1
- Differences in Command Function F-1
  - Queue-to Split and Check Split F-2
  - Goto Step and Goto Vector F-4
  - Route-to Number F-5
  - Announcement F-6
  - Wait-time F-6
  - Busy F-7
- General Call Vectoring Functional Differences F-7
- Differences in Defining/Interpreting Split Flows F-10
  - R3 CMS Standards F-11
- Differences Between G2 and DEFINITY ECS EAS F-12

G Interactions Between Call Vectoring/EAS and BCMS/CMS G-1

- Introduction G-1
- BCMS/CMS Tracking in a Call Vectoring Environment G-2
  - Defining and Interpreting Call Flows G-2
- Using BCMS/CMS Reports to Evaluate Call Vectoring Activity G-12
  - CMS Reports G-13
  - BCMS Reports G-13
- Using CMS in an EAS Environment G-14
  - Tracking Entities G-14

H Operation Details for the Route-to Command **H-1**

- Introduction H-1

I Detailed Call Flow and Specifications for Converse—VRI Calls **I-1**

- Introduction I-1
- Converse Call Placement I-2
- Data Passing I-3
- VRU Data Collection I-6
- Script Execution I-7
- Data Return I-7
- Script Completion I-10
- DEFINITY System Data Collection I-10

J Security Issues **J-1**

- Introduction J-1
- Remote Access J-1
  - Front-Ending Remote Access J-2
  - Replacing Remote Access J-2
- EAS J-3
- Limiting Outside Access Using VDN COR Restrictions J-3
- Vector Initiated Service Observing J-4
- Voice Response Integration J-4

K Setting Up a Call Center **K-1**

- Introduction K-1
- Call Vectoring/Non-EAS Option K-2
- Call Vectoring/ EAS Option K-11

L Converting a Call Center to EAS **L-1**

- Introduction L-1
- Step 1: Pre-EAS Cutover Administration for the System L-2
- Step 2: Pre-EAS Cutover Administration for CMS L-7
- Step 3: Pre-EAS Cutover Administration for AUDIX L-7
- Step 4: Pre-EAS Cutover Administration for Messaging Server L-7
- Step 5: Pre-EAS Cutover Administration for ASAI L-8
- Step 6: EAS Cutover L-8

M **Feature Availability** **M-1**

N **Improving Performance** **N-1**

- Introduction N-1
- Looping Examples N-3
  - Audible Feedback N-3
  - Lookahead Interflow N-4
  - Check N-5
- Other Examples N-8
  - After Business Hours N-8
  - Lookahead Interflows N-9
- Relative Processing Cost of Vector Commands N-10

O **Call Vectoring System Parameters** **O-1**

P **References** **P-1**

- Basic DEFINITY ECS documents P-1
  - Administration P-1
  - Installation and maintenance P-3
  - Call center documents P-6
  - Application-specific documents P-7
  - Non-U.S. audiences P-10

GL **Glossary and Abbreviations** **GL-1**

IN **Index** **IN-1**

About This Document

Feature Availability

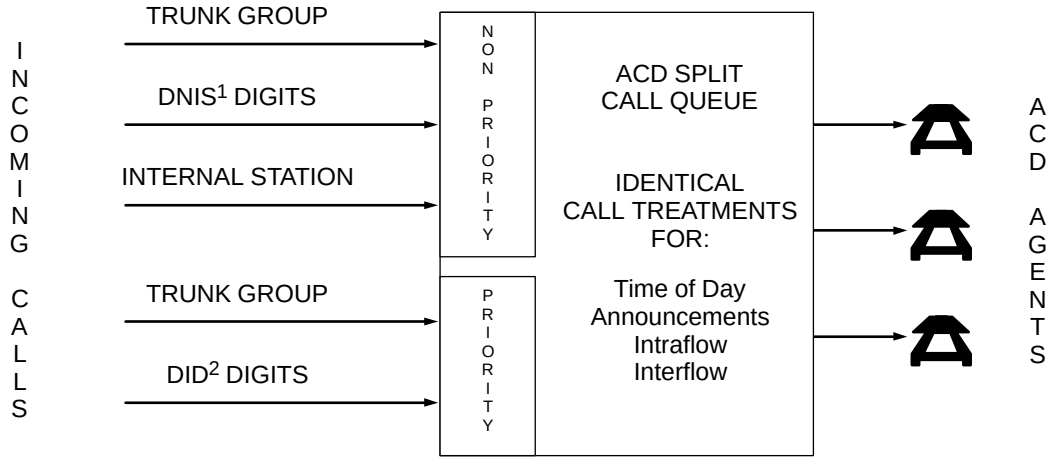
Some Call Vectoring features must be purchased separately. See Appendix M for descriptions of these optional features.

What is Call Vectoring?

NOTE:
Sample vectors are provided throughout this manual to illustrate vectoring features and capabilities. Because they are simplified to clearly demonstrate specific features, they are not complete and should not be used without modification at your call center.

Call Vectoring provides a highly flexible approach for managing incoming call traffic to the DEFINITY Enterprise Communications Server (ECS). By using a series of user-defined commands, you can direct or route internal and network calls as desired and thereby determine how these calls are processed. Calls can be directed to on- or off-network destinations, to Automatic Call Distribution (ACD) agents, or to various treatments. Call Vectoring also can be used with CallVisor ASAI.

Call Vectoring enhances traditional ACD call processing, which may be illustrated as in Figure 1.



1. Dialed Number Identification Service
2. Direct Inward Dialing

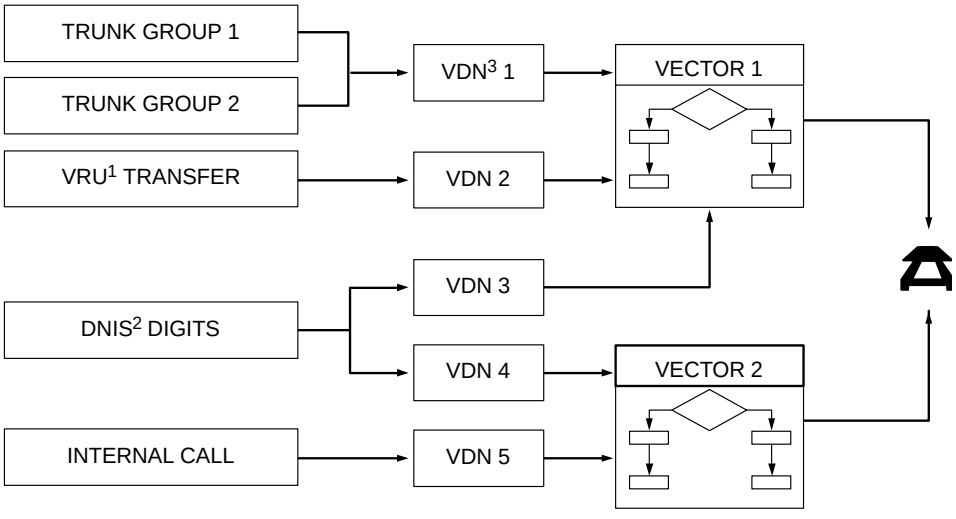
Figure 1. Traditional ACD Call Processing

As Figure 1 illustrates, the traditional ACD approach is rather limited in the way it handles queued calls (that is, all calls within a specific queue receive identical announcements, intraflow parameters, etc.). Call Vectoring, on the other hand, permits each call to be treated uniquely according to a number of factors, including the number the caller dials, the number the caller calls from, the number of calls in queue, and the time of day and/or day of the week. This even applies to all calls that are ultimately handled by the same agent group.

Call Vectoring is comprised of three basic components:

- Vector Directory Numbers
- Vectors
- Vector commands

Working together, these components direct incoming calls and ASAI event reports and requests to the desired answering destinations, and they specify how each call is processed. Call Vectoring may be illustrated as in Figure 2.



1. Voice Response Unit
2. Dialed Number Identification Service
3. Vector Directory Number

Figure 2. Call Vectoring

As Figure 2 illustrates, an incoming call to the DEFINITY System with Call Vectoring enabled is first directed to a Vector Directory Number (VDN). A VDN is an internal telephone number that, in turn, directs the call to a specific vector. The VDN represents the call type or category (for example, billing, customer service, etc.), and thus, it defines the service desired by the caller. Multiple VDNs may point to the same or to different vectors, depending upon whether the relevant calls are to receive the same or different treatment.

The vector is a set of commands that define the processing of a call. The processing a call receives depends on the commands in the vector. For example, a call can be queued and then routed to another destination.

Screen 1 shows an example of a vector.

```
1. goto step 3 if calls-queued in split 9 pri l < 20
2. busy
3. queue-to split 9 pri l
4. wait-time 12 seconds hearing ringback
5. announcement 2921
6. wait-time 998 seconds hearing music
```

Screen 1. Vector Example

Each individual vector can contain up to 32 command steps. Multiple vectors can be chained together to extend processing capabilities or to process calls to the same or different answering destinations. Any number of calls can use the same multiple vectors and process steps independently. Understanding your goals and planning your system before you begin writing vectors is crucial. A planning guide is provided in Appendix K, "Setting Up a Call Center".

What is Expert Agent Selection (EAS)?

Expert Agent Selection (EAS) allows call center managers to provide the best possible telephone service to callers by matching their needs with the skills or talents of the agents. Caller needs and agent skills are matched via Call Vectoring. All the Call Vectoring features described in this guide can be used with EAS.

Matching the call to an agent with the appropriate skills reduces transfers and call-holding time. Accordingly, customer satisfaction is increased. Also, since an entire agent group need not be trained at the same time for the same skills, employee satisfaction is increased.

In addition to matching the skills required for a call to an agent with one of these skills, EAS provides other capabilities:

- **Logical Agent** associates hardware (voice terminal) with an agent only when the agent is logged in. While the agent is logged in, calls to the Agent Login ID are directed to the agent.
- **Direct Agent Calling** allows a user to call a particular agent and have the call treated as an ACD call.

Most EAS administration can be completed before you activate it, thus minimizing the down time for upgrading to EAS. EAS is described in Chapter 11.

Call Vectoring Features

Call Vectoring provides the following features:

- **Basic Call Vectoring** allows you to "program" the type of processing applied to a call by arranging a set of vector commands in the desired sequence. Depending on the command, you can do the following:
  - Place the call in queue until an agent is available to answer the call.
  - Provide a recorded information or delay announcement to the caller.
  - Allow the caller to leave a recorded message.
  - Access a Voice Response Unit (VRU) to start a script.
- **Call Prompting** allows you to collect digits and give some call control to the caller. Specifically, this feature allows callers with touch-tone or rotary phones to enter digits that are subsequently processed by the vector. Among other tasks, Call Prompting allows the caller to do the following:
  - Select one or more options from a menu in order to access recorded information or be routed to the correct split or agent.
  - Enter an extension to which a call can be routed.
  - Provide the call center with data (such as a credit card number) that the center can use to process the call. This data also can be displayed on the voice terminal of the agent who answers the call.
- **G3V4 Enhanced** provides for the following:
  - Specification of a priority level with the oldest-call-wait conditional.
  - Use of enhanced comparators.
  - Use of wildcards in digit strings for matching collected digits and ANI or II-digits.
  - Use of Vector Routing Tables.
  - Multiple Audio/Music Sources for use with the *wait-time* command.
- **Advanced Vector Routing** allows you to route calls based on three additional conditions:
  - Rolling Average Speed of Answer for a split, skill, or VDN.
  - Expected Wait Time for a split (skill) or for a call.
  - The number of calls that are active in a specified VDN.

- **ANI/II Digits Routing** allows you to route calls based on either:
  - The caller identity (ANI) or,
  - The type of line where the call was originated (II-digits).
- **Call Information Forwarding (CINFO)** allows you to collect caller-entered digits (ced) and customer database provided digits (cdpd) from the network. These digits can then be used in the same way as digits collected with Call Prompting.
- **Enhanced Information Forwarding** allows you to forward the following information with each call: ANI, II-Digits, CINFO, ASAI-provided user information, Look-Ahead Interflow (LAI) information (such as VDN name), Universal Call ID (UCID), Best Service Routing™ data, Collected Digits, and elapsed in-VDN time
- **Look-Ahead Interflow** allows the call center to intelligently off-load some or all calls to another ACD switch. When this feature is enabled, a call arriving at a vector that is unable to handle the call (due to preset limits) can interflow to a switch in a remote location whenever the latter switch is able to receive the call. By using this feature, you can establish a “load balancing” of calls among multiple locations primarily for lower-traffic, reduced-agent staffing periods. By using a certain conditional in a vectoring command, you can reduce processing and achieve First-In First-Out (FIFO) call distribution across the call center sites.
- **Best Service Routing™ (BSR)** allows DEFINITY® ECS to compare specified splits or skills, determine which will provide the best service to a call, and deliver the call to that resource. If no agents are currently available in that split or skill, the call is queued. BSR is available in single-site and multi-site versions. Single-site BSR compares splits or skills on the DEFINITY ECS where it resides to find the best resource to service a call. Multi-site BSR, activated via the Look-Ahead Interflow feature, extends this capability across a network of DEFINITY switches, comparing local splits or skills, remote splits or skills, or both, and routing calls to the resource that provides the best service. To respond to changing conditions and operate more efficiently, BSR monitors the status of the specified resources and adjusts call processing appropriately.
- **Adjunct Routing** provides you with a means of evaluating calls before the calls are processed and implementing complex call center applications. Specifically, this feature allows a DEFINITY System to request instructions from an associated adjunct, which is a processor that performs one or more tasks for another processor (the switch, in this case). The adjunct makes a routing decision according to agent availability and/or caller information sent by the switch, and it returns the routing response to the switch. By using this feature, the call center ensures that each call is delivered to the appropriate destination.

Contents and Organization of the Guide

This guide discusses all facets of Call Vectoring and EAS. It is organized as follows:

- Overview (About This Document)
- Tutorial (Chapter 1)
- Reference (Chapter 2 through Chapter 12, Appendix A through Appendix O)
- A brief catalog of DEFINITY system documents (Appendix P, "References")
- Glossary and Abbreviations
- Index

The guide first concentrates on illustrating Call Vectoring principles (Chapter 1 through Chapter 10). Chapter 11 presents a thorough discussion of EAS, which builds on Call Vectoring. Chapter 12 presents several Call Vectoring and EAS applications. Finally, the appendices, glossary, and index provide information and references to both Call Vectoring and EAS topics.

Intended Audience and Use of the Guide

The guide is intended primarily for personnel who opt to use Call Vectoring and/or EAS. You should use this guide as an information source for implementing Call Vectoring and/or EAS. A knowledge of Automatic Call Distribution (ACD) is assumed.

The level of your expertise in Call Vectoring and/or EAS should determine how you use the guide. Users who are unfamiliar with Call Vectoring should read the overview, then study the tutorial. Users who will be using EAS should read Chapter 11. Users who want to learn more about Call Vectoring should review Chapter 2 through Chapter 12 in order to get a good grasp of how the Call Vectoring features function. Finally, advanced users of Call Vectoring and/or EAS may only find it necessary to periodically reference a specific appendix or two (such as Appendix A, which contains a set of Call Vectoring/EAS command "manual pages") to get the information needed.

Users who want to set up a Call Center (EAS and non-EAS) should read Appendix K, and users who want to convert a Call Center to EAS should read Appendix L.

References

The publications listed in this section should be used to supplement the information presented in this guide:

- *DEFINITY Enterprise Communications Server Release 6 Administration and Feature Description*, 555-230-522
- *DEFINITY Enterprise Communications Server CallVisor ASAI Technical Reference*, 555-230-220
- *DEFINITY Enterprise Communications Server Release 5 Basic Call Management System (BCMS)*, 555-230-706
- *CentreVu® Call Management System R3V6 Administration,* 585-215-850
- *CentreVu® Supervisor R3V6 Reports,* 585-215-851 (or previous version CMS documentation that came with your system)
- *GBCS Products Security Handbook*, 555-025-600.

Tutorial

1

Introduction

This chapter gives you a practical start using Call Vectoring, presenting the basics you need to write a representative vector and enter it on-line. The last section of the chapter summarizes the benefits of Call Vectoring, and it identifies example vectors in the reference section of the guide that illustrate these benefits.

Entering the Vector On-Line

A vector can be entered on-line via the following two methods:

- Basic Screen Administration (on the Manager I or G3-MA)
- CentreVu® Call Management System

NOTE:
All references to CMS in this manual, unless otherwise noted, refer to CMS Release 2, CMS Release 3 through Version 2 or CenterVu® CMS.

The following section discusses the Basic Screen Administration method for entering a vector on-line at your DEFINITY system. For complete details for creating a vector with CMS, consult the *CentreVu® Call Management System Release 3 Version 6 Administration* manual, 585-215-850 (or earlier version CMS documents that you received with your system).

Procedures for Basic Screen Administration

A vector is entered on-line via Basic Screen Administration by completing the Call Vector Form. This form appears on three screens, as follows in Screen 1-1, Screen 1-2 and Screen 1-3.

```
                                                  Page 1 of 3
                          CALL VECTOR
Number: 20                Name:________________   Multimedia? n   Lock? n


    Basic?y  EAS?n   G3V4 Enhanced? n  ANI/II-Digits? n  ASAI Routing?n
Prompting?n  LAI?n  G3V4 Adv Route? n          CINFO? n          BSR? y

01 ________________
02 ________________
03 ________________
04 ________________
05 ________________
06 ________________
07 ________________
08 ________________
09 ________________
10 ________________
11 ________________
```

Screen 1-1. Call Vector Form (Page 1 of 3)

```
                                                  Page 2 of 3
                          CALL VECTOR

12 ________________
13 ________________
14 ________________
15 ________________
16 ________________
17 ________________
18 ________________
19 ________________
20 ________________
21 ________________
22 ________________
```

Screen 1-2. Call Vector Form (Page 2 of 3)

```
                                                    Page 3 of 3
                              CALL VECTOR

 23 _________________
 24 _________________
 25 _________________
 26 _________________
 27 _________________
 28 _________________
 29 _________________
 30 _________________
 31 _________________
 32 _________________
```

Screen 1-3. Call Vector Form (Page 3 of 3)

The following list summarizes how you can enter a vector on-line via Basic Screen Administration. For complete details on this process, consult *DEFINITY Enterprise Communications Server Release 6 Administration and Feature Guide,* 555-230-522.

1. Access the Call Vector Form by executing the **change vector *x*** command, where ***x*** is the number of the vector you want to access. Use the change vector command either to change an existing vector, or to create a new vector.

   If you are not certain of the number or name of a vector, enter the **list vector** command to view a complete list of all vectors that have been administered for your system.

2. Assign a name to your vector by completing the blank next to the `Name` field. The vector name can contain up to 27 alphanumeric characters.

   NOTE:
   The vector number, which appears next to the `Number` field, is automatically assigned by the system.

3. In the `Multimedia?` field, indicate whether the vector should receive early answer treatment for multimedia calls. Valid values are **y** or **n**.

   NOTE:
   This only applies if Multimedia Call Handling is enabled.

   - If you expect this vector to receive multimedia calls, set this field to **y**. The call is considered to be answered at the start of vector processing, and billing for the call starts at that time.

- If you do not expect the vector to receive multimedia calls, set this field to **n**.

4. In the Lock? field, indicate whether you will allow this vector to be displayed on and edited from a client application such as CentreVu® Visual Vectors.
   - If you enter **y**, the vector is locked and can only be displayed and modified in the DEFINITY ECS switch administration software.
   - If you enter **n**, the vector is not communicated to client software such as CentreVu® Visual Vectors or CentreVu® CMS and may not be displayed and modified from these programs.

   NOTE:
   Always lock vectors that contain secure information (for example, access codes).

5. Look at the next fields and note where a **y** (yes) appears. These fields indicate the Call Vectoring features and corresponding commands you can use. (The Call Vectoring features are optioned from the Customer Options Screen.) On the other hand, if an **n** (no) appears in one of these fields, you cannot use the corresponding feature. A **y** in one of the fields indicates the following

Basic	You can use the Basic Call Vectoring commands. See Chapter 3, ‘‘Basic Call Vectoring’’.
EAS	Expert Agent Selection is enabled. See Chapter 11, ‘‘Expert Agent Selection’’.
G3V4 Enhanced	You can use the G3V4 Enhanced Vector Routing commands and features. See Appendix M, ‘‘Feature Availability’’ for an explanation of which features are included with G3V4 Enhanced Vector Routing.
ANI/II-Digits	You can use the ANI and II-Digits Vector Routing commands. See Chapter 6, ‘‘ANI /II-Digits Routing and Caller Information Forwarding (CINFO)’’. ANI/II-Digits Routing requires G3V4 Enhanced Vector Routing.
ASAI Routing	You can use the Adjunct Routing command. See Chapter 10, ‘‘Adjunct Routing’’.
Prompting	You can use the Call Prompting commands. See Chapter 4, ‘‘Call Prompting’’.
LAI	Look-Ahead Interflow is enabled. If BSR is also **y**, then multi-site Best Service Routing™ is enabled. See Chapter 7, ‘‘Look-Ahead Interflow’’ and Chapter 8, ‘‘Best Service Routing™’’.

G3V4 Adv Route	You can use the G3V4 Advanced Vector Routing commands. See Chapter 5, "Advanced Vector Routing".
CINFO	You can collect ced and cdpd digits with the collect digits step. See Chapter 6, "ANI /II-Digits Routing and Caller Information Forwarding (CINFO)".
BSR	Best Service Routing™ is enabled, and you can use the BSR commands. The available commands vary depending on whether you are using single-site or multi-site BSR. See Chapter 8, "Best Service Routing™".

6. Enter a maximum of 32 vector commands in the blanks next to the step numbers. See Appendix A for a complete description of all Call Vectoring commands.

   NOTE:
   You need not type every letter of each command that you enter. If you type just the first few letters of a command and press RETURN or TAB, the system spells out the entire command.

7. Save the vector in the system by pressing ENTER.

Enhanced Vector Editing

Enhanced Vector Editing allows you to insert and delete vector steps while editing a vector on the switch.

NOTE:
After editing a vector, be certain to verify that the vector will work as you intend it to. This is particularly important if you deleted a step that was the target of a *goto* step.

Inserting a Vector Step

To insert a vector step complete the following procedure:

1. After entering the change vector command, press F6 (edit).
2. At the command line, type **i** followed by a space and the number of the step you would like to add. You cannot add a range of vector steps. Enter the command. For example, to insert a new vector step 3, type **i 3** and enter the command.
3. Type the new vector step.

When a new vector step is inserted, the system automatically renumbers all succeeding steps and renumbers *goto* step references as necessary. Under certain conditions, attempts to renumber *goto* step references will result in an ambiguous renumbering situation. In this case, the step reference is replaced by

a *. You will receive a warning indicating that you must resolve the ambiguous references and your cursor automatically moves to the first reference that needs to be resolved. You cannot save a vector with unresolved goto references.

You cannot insert a new vector step if 32 steps are already entered in the vector. However, you can extend the vector program to another vector by using the *goto vector unconditionally* command at step 32.

Deleting a Vector Step

To delete a vector step complete the following procedure:

1. After entering the change vector command, press F6 (edit)
2. At the command line, type **d** followed by a space and the number of the step you would like to delete. Enter the command. You can delete a range of vector steps. For example, to delete steps 2 through 5, type **d 2-5** and enter the command.

When a vector step is deleted, the system automatically renumbers all succeeding steps and renumbers *goto* step references as necessary. Under certain conditions, attempts to renumber *goto* step references will result in an ambiguous renumbering situation. In this case, the step reference is replaced by a *.

For example, if a vector step that is the target of a goto step is deleted, the goto references are replaced by *s. For example, if you delete step 7 when you have a vector step *goto step 7 if ...*, the 7 is replaced by a *.

You will receive a warning indicating that you must resolve ambiguous references and your cursor automatically moves to the first reference that needs to be resolved. You cannot save a vector with unresolved goto references.

Constructing a Vector: One Approach

This section is intended to provide you with one logical approach to constructing a vector. In so doing, the section presents a starting vector that consists of one step and then builds upon this vector to produce a new vector that provides additional functions. This vector building process continues through several phases until a final complete vector is constructed. As each phase is presented, you are introduced to one or more new vector commands and/or approaches to vector processing. While it is not practical to present all such commands and approaches along the way to constructing a single final vector, those presented in this tutorial should allow you to get a good grasp of how to use Call Vectoring.

Phase 1: Queuing a Call to the Main Split

If a call cannot be immediately answered by an agent (or operator), the call is usually queued until an agent becomes available. A call can be connected to an available agent or queued via the vector in Screen 1-4.

```
                                                    Page 1 of 1
                          CALL VECTOR
Number: 27            Name: base                Multimedia? n    Lock? n

    Basic? y EAS? n  G3V4 Enhanced? n  ANI/II-Digits? n  ASAI Routing? n
Prompting? n LAI? n G3V4 Adv Route? n          CINFO? n           BSR? y

01 queue-to split 5 pri l
02 _________________
03 _________________
04 _________________
05 _________________
06 _________________
07 _________________
08 _________________
09 _________________
10 _________________
11 _________________
```

Screen 1-4. Queuing Call to Main Split

If an agent is available, the *queue-to split* command automatically sends the call to the agent without queuing the call. However, if no agent is available, the command queues the call to the main split (or group) of agents. Once the call is sent to the main split queue, the call remains there until either it is answered by an agent or some other treatment is provided.

Each call queued to a split occupies one queue slot in that split. Calls are queued sequentially as they arrive according to the assignment of the priority level. In our vector, note the priority level *low* is assigned to the call. The priority level establishes the order of selection for each call that is queued. A call can be assigned one of four priority levels: *top, high, medium,* or *low*. Within a given split (the main split, in our vector), calls are delivered to the agent sequentially as they arrive to the split queue and according to the priority level assigned. Accordingly, calls assigned a *top* priority (if any) are delivered to an agent first, calls that are assigned a *high* priority are delivered second, etc.

Finally, note that the call is queued to Split 5.

Phase 2: Providing Feedback and Delay Announcement

In the last section, we mentioned that a call remains queued until an agent becomes available to answer the call. In the meantime, the caller would no doubt like to hear some feedback assuring him or her that the call is being processed. The vector in Screen 1-5 provides one solution.

```
                                                    Page 1 of 3
                          CALL VECTOR
Number: 27             Name: base                 Multimedia? n    Lock? n

    Basic? y EAS? n  G3V4 Enhanced? n  ANI/II-Digits? n  ASAI Routing? n
Prompting? n LAI? n G3V4 Adv Route? n          CINFO? n            BSR? y

01 queue-to split 5 pri l
02 wait-time 10 seconds hearing ringback
03 announcement 2771
04 _________________
05 _________________
06 _________________
07 _________________
08 _________________
09 _________________
10 _________________
11 _________________
```

Screen 1-5. Providing Feedback and Delay Announcement

NOTE:
Announcement 2771 could contain this message: "We're sorry. All of our operators are busy at the moment. Please hold."

The *wait-time* command in step 2 provides a delay of a specified number of seconds before the next vector step is processed. The time parameter may be assigned any even number in the range of 0 through 998. In our vector, the time specified is 10 seconds.

In addition to the delay period, the *wait-time* command provides the caller with feedback. In our vector, *ringback* is provided. Other types of feedback that can be provided with the *wait-time* command are: silence; system music; or an alternate audio/music source. For more information see, "Delays with Audible Feedback" on page 3-5.

Theoretically, then, the *wait-time* command in our vector provides the caller with 10 seconds of ringback. But what happens if an agent answers the call before the *wait-time* command runs its course? If this happens, the command is terminated (that is, the delay period is ended and the accompanying feedback is stopped). So, returning to our example, let's presume the call is delivered to an agent after four seconds. In such a case, the following is true:

- Caller does not hear the remaining six seconds of ringback, inasmuch as the delivery of the call to the agent is the primary objective.
- Announcement in step 3 (discussed next) is not played.

If the call is not answered by the time the *wait-time* command in step 2 is completed, vector processing continues with the *announcement* command in step 3.

The *announcement* command consists of a recorded message, and it is often used to encourage the caller to stay on the phone or to provide information to the caller. If a call is delivered to an agent during the *announcement* command, the announcement is interrupted. Otherwise, the announcement is played from beginning to end. Thereafter, the call remains in queue until it is answered by an agent or until the caller hangs up. Multiple callers can be connected to an announcement at any time. See “Recorded Announcement” in the *DEFINITY Enterprise Communications Server Release 6 Administration and Feature Guide,* 555-230-522, for more information about announcements.

Phase 3: Repeating Delay Announcement and Feedback

The vector in the previous section provides feedback to the caller after the call is queued. However, if the announcement in step 3 is played, and if the agent does not answer the call soon after the announcement is complete, the caller may end up holding the line for too long a time without receiving any further feedback or treatment. The vector in Screen 1-6 provides one solution.

```
                                                        Page 1 of 1
                           CALL VECTOR
Number: 27            Name: base                Multimedia? n    Lock? n

    Basic? y EAS? n  G3V4 Enhanced? n  ANI/II-Digits? n  ASAI Routing? n
Prompting? n LAI? n G3V4 Adv Route? n          CINFO? n           BSR? y

01 queue-to split 5 pri l
02 wait-time 10 seconds hearing ringback
03 announcement 2771
04 wait-time 60 seconds hearing music
05 goto step 3 if unconditionally
06 _________________
07 _________________
08 _________________
09 _________________
10 _________________
11 _________________
```

Screen 1-6. Repeating Delay Announcement and Feedback

The *wait-time* command in step 4 of this vector provides additional feedback (this time, music) to the caller. If the call is not answered by the time step 4 completes, the *goto step* command in step 5 is processed.

Up to this point, we have discussed and illustrated Call Vectoring commands that cause *sequential flow* (that is, the passing of vector processing control from the current vector step to the next sequential vector step). The *goto step* command is an example of a Call Vectoring command that causes *branching* (that is, the passing of vector processing control from the current vector step to either a preceding or succeeding vector step).

The *goto step* command in step 5 allows you to establish an announcement-wait loop that continues until the agent answers the call. Specifically, the command makes an unconditional branch to the *announcement* command in step 3. If the call is not answered by the time the announcement in step 3 is complete, control is passed to the *wait-time* command in step 4. If the call is still not answered by the time this command completes, control is passed to step 5, where the unconditional branch is once again made to step 3. As a result of the established loop, the caller is provided with constant feedback.

Phase 4: Queuing a Call to a Backup Split

Up to this point, we have dealt with a call queued to one split: the main split. However, Call Vectoring allows a call to be queued to a maximum of three splits simultaneously. If a call is queued to multiple splits, the call has a better chance of being answered more quickly. Multiple split queuing is especially useful during periods of heavy call traffic.

The vector in Screen 1-7 allows a call to queue to two splits.

```
                                                        Page 1 of 1
                          CALL VECTOR
Number: 27             Name: base                 Multimedia? n    Lock? n

    Basic? y EAS? n  G3V4 Enhanced? n  ANI/II-Digits? n  ASAI Routing? n
Prompting? n LAI? n G3V4 Adv Route? n           CINFO? n           BSR? y

01 queue-to split 5 pri l
02 wait-time 10 seconds hearing ringback
03 announcement 2771
04 wait-time 10 seconds hearing music
05 check split 7 pri m if calls-queued < 5
06 wait-time 60 seconds hearing music
07 announcement 2881
08 goto step 5 if unconditionally
09 _________________
10 _________________
11 _________________
```

Screen 1-7. Queuing Call to Backup Split

We have already discussed how the *queue-to split* command in step 1 queues the call to the main split. If the call is not answered by the time the *wait-time* command in step 4 completes, the *check split* command in step 5 attempts to queue the call to backup Split 7 at a medium priority. The condition expressed in the command (*if calls-queued < 5*) determines whether or not the call is to be queued to the backup split. Specifically, if the number of calls currently queued to Split 7 at a medium or higher priority is less than 5, the call is queued to the split. Note that if the call is queued, the call in this case is assigned a *medium* priority instead of a *low* priority, which is assigned if the call is queued by the *queue-to split* command in step 1. It is a good practice to raise the priority level in subsequent queuing steps in order to accommodate callers who have been holding the line for a period of time. (We could have even assigned a *high* priority instead of just a *medium* priority in step 5.)

The *calls-queued* condition is one of seven conditions that can be included in the *check split* command. The other conditions are *unconditionally, average speed of answer (rolling-asa), available agents, staffed agents, expected wait time* and *oldest call waiting*. As is true for the *queue-to split* command, the *check split* command can queue a call at one of four priorities: *low, medium, high* or *top*.

We are including a queuing step within the loop, thus giving the call repeated opportunities to queue (if necessary). The call queues to split 7 only once.

Phase 5: Checking the Queue Capacity

It is a good practice to check the main split queue for the number of calls already queued before allowing another call to queue to the split. The reason for this is that there is a limited number of queue slots assigned to each split. The number of such slots assigned to each split is defined in the queue length field on the hunt group screen. A call that attempts to queue to a split with no available queue slots cannot be queued to that split and, accordingly, the *queue-to split* command fails. Vector processing would then continue with the next vector step. The vector in Screen 1-8 contains provisions for checking queue capacity.

```
                                                        Page 1 of 1
                          CALL VECTOR
Number: 27             Name: base                 Multimedia? n    Lock? n

    Basic? y EAS? n  G3V4 Enhanced? n  ANI/II-Digits? n  ASAI Routing? n
Prompting? n LAI? n G3V4 Adv Route? n          CINFO? n           BSR? y

01 goto step 10 if calls-queued in split 5 pri l > 20
02 queue-to split 5 pri l
03 wait-time 10 seconds hearing ringback
04 announcement 2771
05 wait-time 10 seconds hearing music
06 check split 7 pri m if calls-queued < 5
07 wait-time 60 seconds hearing music
08 announcement 2881
09 goto step 6 if unconditionally
10 busy
11 _______________
```

Screen 1-8. Checking Queue Capacity

A check of split 5 is implemented by the *goto step* command in step 1. In Screen 1-8, 21 slots are assigned to split 5 (that is, the queue length for split 5 is *21*). Accordingly, the *goto step* command tests whether the split contains more than 20 calls via the condition *if calls-queued in split 5 > 20 pri l*. If this test is successful, control is passed to the *busy* command in step 10. The *busy* command gives the caller a busy signal and eventually causes the call to drop.

On the other hand, if 20 or fewer calls at a medium priority are already queued to the main split when step 1 executes, the *queue-to split* command in step 2 queues the call, and vector processing continues at step 3.

NOTE:
Instead of providing the caller with a busy tone if the *queue-to split* step cannot queue the call, we can queue the call to another split that is designed to serve as a backup split. To do this, we can change the step parameter for the *goto step* command from 10 to 6 (so that the command reads *goto step 6.....*). In such a case, control is passed from step 1 to the *check split* step (step 6). Inasmuch as this queuing step is included within a continuous loop of steps (steps 6 through 9), continuous attempts to queue the call are now made (if necessary).

Phase 6: Checking for Non-Business Hours

If a caller calls during non-business hours, you can still provide the caller with some information for calling back during working hours by playing the appropriate recorded message. The following vector, Screen 1-9 and Screen 1-10, illustrates one approach in this regard. This vector would be used for a company that was open seven days a week, from 8:00 A.M to 5:00 P.M., including Saturday and Sunday.

```
                                                        Page 1 of 2
                          CALL VECTOR
Number: 27             Name: base                 Multimedia? n    Lock? n

    Basic? y EAS? n  G3V4 Enhanced? n  ANI/II-Digits? n  ASAI Routing? n
Prompting? n LAI? n G3V4 Adv Route? n          CINFO? n           BSR? y

01 goto step 12 if time of day is all 17:00 to all 8:00
02 goto step 11 if calls queued in split 5 pri l > 10
03 queue-to split 5 pri l
04 wait-time 10 seconds hearing ringback
05 announcement 2771
06 wait-time 10 seconds hearing music
07 check split 7 pri m if calls-queued < 5
08 wait-time 60 seconds hearing music
09 announcement 2881
10 goto step 6 if unconditionally
11 busy
```

Screen 1-9. Checking for Non-Business Hours (Screen 1 of 2)

```
                                                    Page 2 of 2
                              CALL VECTOR

 12 disconnect after announcement 3222
 13 _________________
 14 _________________
 15 _________________
 16 _________________
 17 _________________
 18 _________________
 19 _________________
 20 _________________
 21 _________________
 22 _________________
```

Screen 1-10. Checking for Non-Business Hours (Screen 2 of 2)

NOTE:
Announcement 3222 could contain this message: "We're sorry. Our office is closed. Please call back any day between 8:00 A.M. and 5:00 P.M."

The *goto step* command in step 1 checks if the call arrives during non-business hours. Specifically, if the call arrives between 5:00 P.M. and 8:00 A.M. on any day of the week, the command passes control to step 12. The *disconnect* command in step 12 includes and provides an announcement that first gives the caller the appropriate information and then advises him or her to call back at the appropriate time. The command then disconnects the caller.

On the other hand, if the call does not arrive during the specified hours, control is passed to step 2, and vector processing continues. On step 2, split 5 is checked for calls waiting at priority low and above (that is, for all priorities).

NOTE:
As an alternative to disconnecting callers who place a call during non-business hours, you can allow callers to leave a message by including the *messaging split* command within the vector. See Chapter 3 for more details.

Benefits of Call Vectoring

Coupled with Automatic Call Distribution (ACD), Call Vectoring enables calls to be processed at a faster rate within an intelligent, real-time system. As a result, Call Vectoring provides an appreciable cost saving to the user.

Table 1-1 summarizes the benefits of Call Vectoring. The last column in Table 1-1 identifies the vector(s) [via the appropriate screen(s)] in the reference portion of the manual that illustrate(s) these benefits.

Table 1-1. Benefits of Call Vectoring

Category	Call Vectoring Benefits	Screen
Call Treatment	Implement special treatment based on the time of day and the day of the week (for example, providing night service).	3-19, 12-1
	Automatically change treatment according to either how long the call has been waiting or to changing traffic or staffing conditions.	3-16, 3-18, 7-1, 7-2, 12-3, 12-4
	Provide appropriate caller feedback during waiting (for example, music or announcements during heavy calling periods).	3-1, 3-2, 3-3, 3-4, 3-5, 3-6, 3-7
	Provide multiple and/or recurring informational or delay announcements that are selected according to the time of day/day of the week, call volume, or staffing conditions.	3-15, 3-16, 7-1, 12-1
	Provide 24 hour/day, 7 day/week automated information announcements.	3-4, 3-5
	Remove selected calls (by providing busy or disconnect).	3-10, 3-11, 3-12, 3-14, 3-18
	Set up and test, in advance, special call treatments for events such as sales, advertising campaigns, holidays, snow days, etc.	3-4, 3-11
	Provide the caller with a menu of choices.	4-6, 4-7, 4-11, 12-3, 12-5
	Execute a VRU script.	3-12, 12-8
	Notify callers of their expected delay in queue.	5-3, 5-4
	Provide multiple audio/music sources.	3-6, 3-7

Continued on next page

Table 1-1. Benefits of Call Vectoring — *Continued*

Category	Call Vectoring Benefits	Screen
Call Routing	Queue calls to up to three splits simultaneously, consequently improving the average speed of answer and agent productivity.	3-13, 12-1, 12-4
	Implement routing to local or distant destinations.	3-16, 4-1, 4-2, 7-1, 12-2, 12-3, 12-4, 12-5
	Connect callers to a voice-mail or messaging system either automatically or at their request.	3-14, 3-15, 12-3, 12-5
	Reduce call transfers by accurately routing callers to the desired destination.	4-1, 4-2, 12-3
	Provide up to four ACD queuing priority levels and the ability to change the queuing priority dynamically, as a result, providing faster service for selected callers.	12-1, 12-3, 12-4
	Reduce agent and/or attendant staffing requirements by: (1) automating some tasks; (2) reducing caller hold time; (3) having agents in one split service multiple call types.	3-4, 3-5, 4-1, 4-11, 4-12, 12-2, 12-3
	Intelligently balance ACD call loads across multiple locations.	7-1, 7-2, 7-5, 8-12, 8-15, 8-18, 12-5, D-2
	Determine the expected wait time in ACD queues.	5-3, 5-5
	Limit the number of simultaneous incoming trunk calls to a VDN.	5-7
	Route calls based on the caller's ANI or the type of the originating line.	6-1, 6-3, 6-4
	Route calls based on CINFO digits provided by the network.	6-5
Information Collection	Provide customized and/or personalized call treatment via information collection and messaging.	4-1, 4-6, 4-11, 12-2, 12-3, 12-5
	Collect information for use by an adjunct or by agent display.	4-7, 12-5
	Collect caller entered or customer database provided CINFO digits from the network.	6-5

Fundamentals of Call Vectoring

2

Introduction

The manner in which a call is processed depends upon a number of components within both the DEFINITY System and the Call Vectoring software. Some of these components include the following:

- Resources available to process a call (for example, agents, splits, software, hardware)
- Vector control flow
- Commands used within the relevant vector(s)

A prudent utilization of these components will produce an effective means of processing telephone calls. This chapter discusses these components, which constitute the fundamentals of Call Vectoring.

Call Management

When a call is placed to a system with Call Vectoring activated, the call accesses the appropriate vector(s) via a Vector Directory Number (VDN). A VDN is a soft extension number not assigned to an equipment location. Each VDN maps to one vector, and several VDNs may map to the same vector. (The VDN is fully discussed later in this chapter).

Once the call goes to a vector, the call's routing and treatment are determined by the commands in the vector. Processing starts at the first step and then proceeds usually sequentially through the vector. Any steps left blank are skipped, and the process automatically stops after the last step in the vector.

Call Vectoring allows the chaining of vector steps and vectors. Accordingly, one vector can direct the call to another vector or VDN, which in turn can direct the call to yet another vector, etc. Note, however, that a maximum of 1,000 vector steps can be executed for any call. When a call enters vector processing, a loop counter keeps track of the number of vector steps executed. If the loop counter exceeds 1,000, a *stop* command is executed. When the *interflow-qpos* conditional is used, however, the execution limit is automatically increased to 3,000 steps (because this conditional is designed to make rapid LAI loops practical).

When a call is delivered to an available agent, the agent can see the information associated with the VDN (for example, the VDN name) on his/her display (if present) and, as a result, can respond to the call with knowledge of the service or response required.

In the real world, of course, not every call placed to a site is immediately answered by an agent. (The customer often has fewer agents than the maximum simultaneous call capacity. Therefore, calls will have to be queued.) The following sections discuss how calls are routed and/or queued via Call Vectoring. Subsequent sections discuss agent states, priority levels, caller feedback, and caller control.

Call Flow

Calls enter a vector and execute steps sequentially beginning with step 1, unless there is a *goto* step. Most steps take microseconds to execute. The exception is steps with *announcement*, *wait-time* and *collect digits* commands. A 0.2 second wait occurs after every seven executed steps unless an explicit wait has occurred. Note that *wait-time* with 0 seconds is not an explicit wait.

Call Vectoring uses several call flow methods to redirect and/or queue calls. These methods involve the use of the Call Vectoring commands, which are described later in this chapter. The methods for queuing and redirecting calls follow:

- **Multiple split queuing** allows a call to queue to up to three splits.
- **Intraflow** allows calls unanswered at a split within a predefined time frame to be redirected to one or more other splits on the same switch. If redirection depends upon a condition to be tested, the process is referred to as *conditional intraflow.*
- **Interflow** allows calls directed to a vector to be redirected to an external or non-local split destination. This destination is represented by a number programmed in the relevant vector. Calls can be routed to an attendant (or attendant queue), a local extension, a remote [that is, Uniform Dialing Plan (UDP)] extension, an external number, or a VDN.

- **Look-Ahead Interflow** can be implemented for call centers with multiple ACD locations connected via an ISDN-PRI. This method allows a call to interflow only if a remote location is better equipped to handle the call. Look-Ahead Interflow (LAI) can occur only when the proper conditions at the receiving switch are met.
- **Best Service Routing**™ allows DEFINITY® ECS to compare specified splits or skills, identify the split or skill that will provide the best service to a call, and deliver the call to that resource. If no agents are currently available in that split or skill, the call is queued. BSR is available in single and multi-site versions. Single-site BSR compares splits or skills on the DEFINITY ECS where it resides to find the best resource to service a call. Multi-site BSR extends this capability across a network of DEFINITY switches, comparing local splits or skills, remote splits or skills, or both, and routing calls to the resource that will provide the best service.
- **Adjunct Routing** allows the switch to request a routing destination from an adjunct processor via ASAI. When this feature is enabled, the switch sends the ASAI adjunct a message containing information about the calling party. The adjunct uses this information to determine, from its databases, the best place for the switch to send the call. The adjunct then passes this routing information back to the switch.

Each of these call control flow methods is fully discussed in the upcoming chapters.

Caller Control

Call Vectoring allows for the temporary transfer of call management control to the caller via several means, as follows:

- **Caller-Selected Routing**. If Call Prompting is enabled, the customer can prompt the caller to input information in the form of dialed digits from a touch-tone telephone or from an internal rotary telephone that is located on the same switch. (A recorded announcement is usually used for prompting purposes.) Once the caller inputs the digits, the call is efficiently and accurately routed to the correct department or destination. This procedure can significantly reduce the number of transferred calls and thus better satisfy the caller's needs.

  In addition, if Call Prompting and Vectoring (CINFO) are enabled, the vector can collect caller entered digits that are passed from the network by way of an ISDN message. These digits can be used to enhance caller control in the same way as digits collected directly by the DEFINITY System.
- **Messaging** is a means of satisfying customer demand during peak calling periods. The caller can leave a voice message in the event that the call cannot be or has not yet been answered. When messaging is enabled,

control is eventually passed to the Audio Information Exchange (AUDIX) or message service split. AUDIX is a voice mail adjunct that allows you to record, edit, forward, and retrieve voice messages to and from callers.

Subsequent chapters discuss these procedures in more detail.

Call Queuing to Splits

Basic Call Vectoring is used primarily to control the call activity of ACD splits. Basic Call Vectoring can queue calls to up to three such splits simultaneously at any one of four priority levels. This process is called *multiple split queuing*. The first split to which a call is queued via this process is called the *main split*, while the second split and the third split (if necessary) are called *backup splits*.

Multiple split queuing serves to provide better service to the caller, and it also enables a better utilization of agents. A call remains queued until either vector processing terminates or the call reaches an agent or another destination. (Vector processing termination is discussed later in this chapter.)

When an agent becomes available in any split to which the call is queued, the following events take place:

- The call begins alerting the agent (or connects if it is automatically answered).
- The call is removed from any other queues. Announcements, music, ringback, or other audio source are also removed.
- Vector processing terminates.

Note that these actions always happen *immediately*, even if the caller is receiving call treatment (for example, hearing an announcement). (Call treatments are discussed later in this chapter.)

Multiple split queuing is illustrated in Chapter 3, “Basic Call Vectoring”.

Split Queue Priority Levels

If a call is queued without Call Vectoring enabled, the call is tracked at one of two priority levels: *Medium* and *High*. On the other hand, if a call is queued via Call Vectoring, the call can be assigned one of four priority levels: *Top, High, Medium,* and *Low*. Within each priority level, calls are processed sequentially as they arrive. This is equivalent to a FIFO (first-in, first-out) order. A vector can be administered to queue calls at any of the four priority levels.

NOTE:
A direct agent call is always given the highest priority and, as a result, it is usually delivered before a call that is directed to a split. The exception is when skill-level Call Handling Preference is optioned and the skill administered to receive direct agent calls is not administered as the agent’s

highest skill level. (A direct agent call is an ACD call that is directed to a specific ACD agent rather than to any available ACD agent in the split. See "Direct Agent Calling" on page 11-27 for more information.)

NOTE:
If a call is already queued to one or more splits that are currently intended to serve as backup splits, the call could be requeued at the new priority level indicated in the command step. (For further details on requeuing, see Appendix A.)

Agent Work Mode

Call Vectoring can make call management decisions according to real-time agent work modes. These states, *available-agents* and *staffed-agents*, can appear as conditions within the *check split* and *goto* Call Vectoring commands (that is, the commands can check for the number of available agents or staffed agents).

For ACD splits, *staffed-agents* represents the number of agents logged-in. *Available-agents* represents the number of agents logged-in and ready to receive an ACD call.

For non-ACD hunt groups, *staffed-agents* is synonymous with *administered*, since hunt groups do not have any log-in, log-out, or work modes. *Available-agents* is the number of agents ready to receive a hunt group call.

For ACD calls, an agent's state is further defined by the relevant *work mode*. The following list describes these modes:

- *After-Call-Work Mode* makes the agent unavailable to receive any ACD calls for any split. This mode can be used when the agent is doing ACD call-related work and can be implemented on a timed basis (Timed ACW). The system automatically places the agent into ACW after the agent completes a call that was received while in the manual-in work mode. In addition, the system can be administered through the Vector Directory Number or Hunt Group forms to automatically place agents into ACW for an administered period of time following the completion of each ACD call received while in the auto-in work mode.
- *Auto-In Work Mode* makes the agent available to receive calls and allows the agent to receive a new ACD call immediately after disconnecting from the previous call. When Multiple Call Handling is enabled an agent in Auto-In Work Mode can receive additional ACD calls while still active on a call.
- *Auxiliary-Work Mode* makes the agent unavailable to receive any ACD calls for the specified split. This mode can be used when an agent is performing non-ACD activities, such as going on a break.

- *Manual-In Work Mode* makes the agent available to receive calls and automatically puts the agent into the *After Call Work Mode* after disconnecting from an ACD call. When Multiple Call Handling is enabled an agent in Manual-In Work Mode can receive additional ACD calls while still active on a call.

See the *DEFINITY Enterprise Communications Server Release 6 Administration and Feature Description*, 555-230-522, for a more complete description of agent work modes and Multiple Call Handling.

Calling Party Feedback

The initial feedback a caller hears as the call is being processed by a vector depends upon the origin classification of the call, which can be one of the following:

- Internal (internal call from another Generic 3 user).
- Non-CO (incoming call over a DID or tie trunk over which incoming digits are received).
- CO (incoming call over a CO or automatic type tie trunk over which no digits are received).

For an internal or a non-CO call, the caller hears silence until one of the following vector steps is reached:

- Wait with system music, ringback, or an alternate audio/music source (Caller hears system music, ringing, or the music or audio associated with an administered port.)
- Announcement (Caller hears the announcement.)
- Busy (Caller hears a busy tone.)
- Call alerting an agent or at a station (Caller hears ringing or the agent answering the call.)

For a CO call, the caller hears CO ringback until one of the following vector steps is reached:

- Announcement (Caller hears the announcement.)
- Wait with system music or alternate audio/music source (Caller hears system music, or the music or audio associated with an administered port.)
- Call answered (Caller hears the agent or voice response answering the call.)

For a CO call for which answer supervision has already been supplied (via the processing of an announcement or the issuing of a *wait-time* command), the caller may hear any of the following:

- Announcement when any *announcement* command is processed.
- Ringback, silence, system music, or an alternate audio/music source when a *wait-time* command is processed.
- Busy when a *busy* command is processed.
- Ringback when the call is alerting a station.

Regardless of the call's origin, the caller can expect to hear different forms of the feedback described in this section as the relevant vector steps are processed. Examples of how subsequent caller feedback is provided in the vector appear in Chapter 3, ''Basic Call Vectoring'' and in several of the following chapters.

Dialed Number Identification Service (DNIS)

In the traditional ACD arrangement, each agent in a given split is trained to answer calls relevant to one specific purpose in an efficient and professional manner. However, ACD managers have recognized the need to enhance this arrangement in which each split is limited to a single call-answering task.

To this end, there is a split arrangement available in which each group of agents is proficient in dealing with several types of calls. The intent is to service multiple call types with the use of fewer agents overall and with less administrative intervention by the ACD manager. Usual economies of scale come into play here. For example, where five agents might be needed in each of three smaller splits (15 agents total) to handle three types of calls, only 11 or 12 agents might be needed in the combined split.

To aid in providing capabilities such as the one just presented, a network service known as Dialed Number Identification Service (DNIS) is available. DNIS enables a unique multidigit number (of usually four digits) that is based on the dialed number to be associated with the call (sent to a customer's telephone, sent to a host computer with ASAI applications, used to provide different treatments for the call, etc.). The number that is sent depends upon the telephone number dialed by the caller. Each DNIS number in your telephone system can be programmed to route to an ACD split comprised of agents who are proficient in handling several types of calls.

Call Vectoring takes the DNIS number from the network and interprets this number as a VDN. When the call is delivered to the agent terminal, the unique name assigned to the particular VDN is displayed on the agent's terminal. This allows the agent to know the specific purpose of the call. As a result, the agent can answer with the appropriate greeting and be immediately prepared to service the customer.

Vector Processing

If Call Vectoring is in effect, telephone calls are processed by one or more programmed sequences of commands called vectors.

The following sections provide a general overview of vector processing. To this end, the following topics are discussed:

- Vector Directory Number (VDN)
- Vector control flow
- Programming capabilities

Vector Directory Number

Within Call Vectoring, calls access the appropriate vector(s) via a Vector Directory Number (VDN). A VDN is a soft extension number that is not assigned to an equipment location. In effect, the digits dialed by a caller or sent to the DEFINITY System from an external network are translated within the system as a VDN.

The VDN points to the vector, and it defines the service desired by the caller. The VDN also serves as the application number, and it allows for specific call-handling and agent-handling statistical reporting within both the DEFINITY Basic Call Management System (BCMS) and the CentreVu™ Call Management System (CMS) for each application handled by the call center.

VDNs are assigned to different vectors for different services or applications that require specific treatments. Any number of VDNs can be assigned to the same vector. As a result, the same sequence of treatments can be given to calls that reach the system via different numbers or from different locations.

The VDN has several properties. These properties are administered by the System Manager on the Vector Directory Number administration form, Screen 2-1.

```
                                                    Page  1 of   1
                        VECTOR DIRECTORY NUMBER

                           Extension:
                                Name:
                   Allow VDN Override?
                                 COR:
                                  TN:
                        Vector Number:
                           AUDIX Name:
                Messaging Server Name:
                            Measured:
      Acceptable Service Level (sec):
       VDN of Origin Annc. Extension:
                           1st Skill:
                           2nd Skill:
                           3rd Skill:

                   Return Destination:
               VDN Timed ACW Interval:
                      BSR Application:
        BSR Available Agent Strategy:
```

Screen 2-1. Vector Directory Number Administration

- **Extension**. Extension number used to identify the VDN.
- **Name**. Name that is associated with the VDN. This name, which is shown on agents' displays, is optional and can contain up to 27 characters. The name may be truncated on agents' displays depending on the application. When information is forwarded with an interflowed call, only the first 15 characters are sent.
- **Allow VDN Override**. Option that allows the name and other attributes of a subsequently routed to VDN to be used instead of the name and attributes of the current VDN. See "VDN Override" on page 2-11 for more information.
- **COR (Class of Restriction)**. 1- or 2-digit number that specifies the COR of the VDN.

  NOTE:
  As a security measure, you can deny incoming callers access to outgoing facility paths by configuring the COR of the VDN to prohibit outgoing access. For details, refer to the *GBCS Products Security Handbook*, 555-025-600.

- **TN**. The Tenant Partition Number for this VDN.
- **Vector Number**. Identification number that determines which vector is activated when a call comes into a VDN. Several VDNs may send calls to the same vector.

- **AUDIX Name**. Only displayed for G3r. The name of the AUDIX that is associated with the VDN as it appears on the Adjunct Names form.
- **Messaging Server Name**. Only displayed for G3r. The name of the messaging server as it appears in the Adjunct names form.
- **Measured**. Indicates whether data about the VDN is being measured internally by BCMS, externally by CMS, by both BCMS and CMS, or not at all.
- **Acceptable Service Level**. The number of seconds within which calls to this VDN should be answered.
- **VDN of Origin Annc. Extension**. The extension number of the VDN of Origin announcement. Only displayed if VDN of Origin Announcement is optioned.
- **1st/2nd/3rd Skill**. Only displayed with Expert Agent Selection (EAS). Gives the skill numbers associated with the VDN.
- **Return Destination**. The VDN extension number to which an incoming trunk call will be routed if it returns to vector processing after the agent drops the call. Only displayed if VDN Return Destination is optioned.
- **VDN Timed ACW Interval** — When a value is entered in this field, an agent in auto-in work mode who receives a call from this VDN is automatically placed into After Call Work (ACW) when the call drops. The interval is the number of seconds the agent will remain in ACW following the call. This setting will override the Hunt Group form setting for Timed ACW.
- **BSR Application** — To use multi-site Best Service Routing with this VDN, enter a 1- to 3-digit number to specify an application plan for the VDN.
- **BSR Available Agent Strategy** — The available agent strategy determines how Best Service Routing identifies the “best” split or skill to service a call when available agents are found. To use Best Service Routing with this VDN, enter an agent selection strategy in this field. Acceptable entries are 1st-found, UCD-LOA, UCD-MIA, EAD-LOA, and EAD-MIA.

VDNs can be preassigned to incoming (automatic) trunk groups, or they can be sent in digit form to the DEFINITY System by the public or a private network. The digits sent to the system can come from the serving Central Office (CO) or toll office via the Direct Inward Dialing (DID) feature or DNIS. The digits can also come from another location via dial-repeating tie trunks, or they can be dialed by an internal caller. For a non-ISDN call, the last four digits of the number are sent to the system, while for an ISDN call, the entire 10-digit number is sent.

The last four or five digits of the destination address passed to the PBX/ACD on a DID/DNIS or on a dial tie-trunk call comprise the VDN. Automatic trunks do not pass destination address digits. Instead, each such trunk always routes to a specific incoming destination that is programmed for the corresponding

automatic trunk group. The destination can be an attendant queue, an extension, a hunt group number, or a VDN.

VDN Override

VDN Override allows information about a subsequently routed to VDN (if any) to be used instead of the information about the current VDN. This information includes:

- The name of the subsequent VDN
- Skill sets
- Messaging split command with the “active” entry
- VDN of Origin Announcement
- Tenant number
- VDN Timed ACW Interval
- VDN Return Destination with the condition that once the call leaves vector processing for the first time, the Return Destination never changes. See Appendix C, ‘‘VDN Return Destination Considerations’’ for more information.
- BSR Application
- BSR Available Agent Strategy

NOTE:
Throughout this document the “active” VDN is the active called VDN as modified by VDN override rules. The “latest” VDN is the most recent VDN to which the call was routed.

VDN Override can be used in conjunction with a vector that prompts the caller for a particular service. Let’s say, for example, a call is placed to an automobile dealer. Like any such dealer, this one consists of several departments, including “Sales” and “Parts.” Let’s presume the caller wants to talk to someone in “Sales.” In such a case, the call comes into the “Main” vector (whose VDN name is “Main”) and is eventually routed to the “Sales” vector (whose VDN name is “Sales”). If VDN Override is assigned to the “Main” VDN, the “Sales” VDN name appears on the agent’s display when the call is finally connected to the agent. This process is illustrated in Figure 2-1. In this example, the “Sales” VDN is the active VDN as well as the latest VDN. If VDN override had not been assigned to the “Main” VDN, the agent’s display would have shown “Main.” In this case, “Main” would be the active VDN while “Sales” would be the latest VDN.

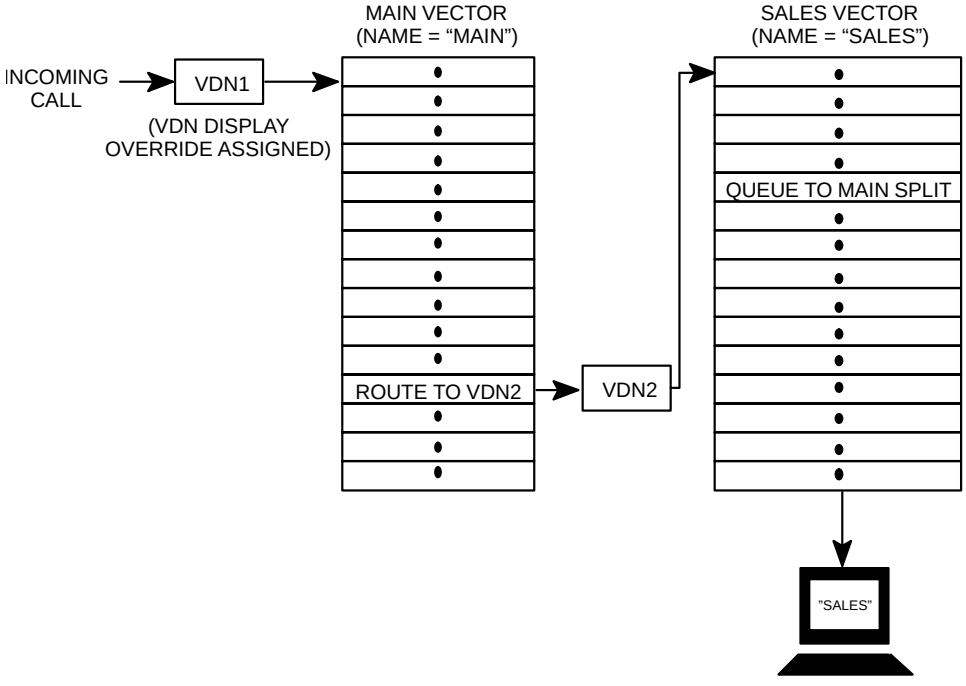


Figure 2-1. VDN Override Assigned to Originally Called VDN

VDN in a Coverage Path

A VDN can be assigned as the last point in a coverage path. Whenever a VDN is assigned as such, a call goes to coverage and can then be processed by Call Vectoring or Call Prompting (if either is enabled). Accordingly, the Call Coverage treatment for the call is extended (that is, coverage can be sent to an external location, or the type of coverage can be controlled by the caller).

VDN in a coverage path is used for a number of applications, including:

- Sending direct agent calls or personal calls to an agent (EAS required).
- Routing coverage calls off-premises via the *route-to* command.
- Serving as a coverage point for specific call operations (for example, sending calls to a secretary during the day and to AUDIX at night).

Using a VDN as a coverage point is illustrated in Chapter 3, "Basic Call Vectoring". For information about interactions, see the *DEFINITY ECS Release 6 Administration and Feature Description Guide*, 555-230-522.

RONA Redirect to a VDN

The Redirection on No Answer (RONA) feature redirects a ringing ACD call after an administered number of rings. It prevents a call from ringing indefinitely at a terminal when an agent does not answer. When a call is redirected, the system puts the agent into AUX work so that the agent is no longer available to receive ACD calls unless the agent has an active or held ACD call. In the case of Auto-Available Splits, the system logs the agent out when a call is redirected.

A VDN can be administered as the destination of a RONA redirected call. In this way, a call that is not answered can be redirected to a VDN to receive special treatment. Enter the number of the destination VDN for a RONA call in the Redirect to VDN field on the Hunt Group form. All calls that are redirected by RONA from that split are sent to the same administered VDN. If no destination VDN is administered, but the number of rings for redirection is entered, the call redirects back to the split/skill.

Direct Agent calls that are not answered follow the agent's coverage path. If no coverage path is administered calls will redirect to the VDN administered on the agent's first primary skill.

See the Redirection on No Answer description in the *DEFINITY ECS Release 6 Administration and Feature Description Guide*, 555-230-522*,* for a more detailed description of RONA.

Service Observing VDNs

The Service Observing feature provides the option of being able to observe VDNs. With this option an observer selects a specific VDN and bridges onto calls (one call at a time) that have just started vector processing for that VDN. The observer hears all tones, announcements, music, and speech that the caller and the agent hear and say, including Call Prompting and caller dialing. Also, the observer hears VDN of Origin Announcements. Once the system makes an observing connection to a call in vector processing, it maintains the connection throughout the life of the call until the call is disconnected or until the observer hangs up. This is true even if the call is routed or transferred externally. See "Service Observing" in the *DEFINITY Enterprise Communications Server Release 5 Feature Description,* 555-230-301 for complete information about Service Observing VDNs.

Vector Control Flow

Vector Processing starts at the first step in the vector and then proceeds sequentially through the vector unless a *goto* command is encountered. Any steps left blank are skipped, and the process automatically stops after the last step in the vector.

The Call Vectoring "programming language" provides three types of "control flow" that serve to pass vector-processing control from one vector step to another. Control flow types are described in the following list.

- **Sequential flow** passes vector-processing control from the current vector step to the following step. Most vector commands allow for a sequential flow through the vector.

  NOTE:
  Any vector command that fails automatically passes control to the following step. For success and/or failure criteria for the Call Vectoring commands see "Criteria for Success/Failure of Call Vectoring Commands".

- **Unconditional branching** *unconditionally* passes control from the current vector step to either a preceding and/or succeeding vector step or to another vector (for example, *goto step 6 if unconditionally*).
- **Conditional branching** *conditionally* passes control from the current vector step to either a preceding and/or succeeding vector step or to a different vector. This type of branching is based on the testing of threshold conditions (for example, *goto vector 29 if staffed-agents in split 6 < 1*).

Each of these control flow types is fully described in the upcoming chapters.

NOTE:
With one exception, call vectoring has an execution limit of 1,000 steps. Once a call enters vector processing, a "loop counter" keeps track of the number of vector steps executed. If the loop counter exceeds 1,000, a *stop* command is executed. When the *interflow-qpos* conditional is used, however, the execution limit is automatically increased to 3,000 steps (because this conditional is designed to make rapid LAI loops practical).

NOTE:
An implicit wait of one second is provided after every seven vector steps if vector processing is not suspended during any one of these steps (see the *wait-time* command manual pages in Appendix A, "Call Vectoring Commands").

Termination vs. Stopping

For the purposes of this guide, the expression *vector processing terminates* means a call has completely left vector processing. This occurs when the call is alerting at an agent's station, is abandoned by the calling party, receives a forced disconnect or a forced busy, or is successfully routed to an extension or to an off-premises number.

It is important to differentiate between vector processing termination and *stopping*, the latter of which is caused by the *stop* command or by the execution of the final step in the vector. Whereas vector processing termination removes

the call from the queue if the call is queued, the *stop* command prevents the processing of new vector steps but leaves the call in queue as the calling party receives feedback, such as ringback. If vector processing stops and the call is not queued, the call is dropped.

Vector processing termination and the *stop* command are discussed and illustrated later in this guide.

Programming Capabilities

The Call Vectoring commands can perform a number of functions relevant to processing telephone calls. A brief explanation for each of these functions follows.

- **Providing call treatments**. The caller can be provided with a recorded announcement explaining that, at the moment, an agent cannot answer the call for some reason (for example, there are no agents available, the work day is over, etc.). Announcements also provide the caller with instructions and encouragement. Also, audible feedback (silence, ringback, system music, or an alternate audio or music source) or a busy tone can be provided to the caller. Provisions can also be made to delay vector processing a specific number of seconds before the next vector step is executed. Also, when necessary, the call can be disconnected. Finally, a session with voice mail (AUDIX) can be initiated.
- **Routing calls**. Calls not immediately answered by an agent can be queued to one or more splits, as explained earlier in this chapter. A caller can also leave a recorded message if he or she chooses to do so. Finally, a call can be routed to a number programmed in the vector or to digits collected from the caller.
- **Branching/programming**. Branches can be made from one vector step to another such step or to another vector. This can be done unconditionally as well as conditionally. Conditional branching is done according to a number of conditions (for example, number of available agents in a split, number of calls in a split queue, the number of the phone the call is made from, etc.). Finally, vector processing can be stopped when necessary.
- **Collecting and acting on information**. Optionally, touch-tone digits can be collected and serve as the basis for further vector processing (for example, a specific agent can be reached via touch-tone digit(s) entered by the caller).
- **Executing VRU scripts**. Voice scripts housed within a Voice Response Unit (VRU) can be executed for the caller. Voice scripts provide the caller with information or instructions, and the caller can often make an appropriate response thereto (by, for example, entering touch-tone digits).

Command Summary

This section lists and describes the commands used by the Call Vectoring features. The list is meant to help familiarize the reader with these commands. The commands are explained further in Chapter 3 through Chapter 10 and also in Appendix A.

- **Adjunct Routing** is available only when the CallVisor ASAI capabilities and Basic Call Vectoring are optioned. The command causes a message to be sent to an ASAI adjunct requesting routing instructions.
- **Announcement** provides the caller with a recorded announcement.
- **Busy** gives the caller a busy signal and causes termination of vector processing.
- **Check** conditionally checks the status of a split or skill for possible termination of the call to that resource. The command either connects to an agent in the split/skill or puts the call into its queue (at the specified queuing priority level) if the condition specified as part of the command is met. A call may be queued to up to three different splits or skills simultaneously.
- **Collect Digits** collects up to 16 digits that are either entered by the caller during vector processing, sent by the network, or received from an adjunct. An optional announcement can be played first when the digits are being collected directly from the caller.
- **Consider Location** obtains the EWT and agent data needed to identify the best remote location in multi-site Best Service Routing applications. One *consider* step must be written for each location you want to check.
- **Consider Split/Skill** obtains the EWT and agent data needed to identify the best local split or skill in single-site Best Service Routing vectors. One *consider* step must be written for each split or skill you want to check.
- **Converse-on Split** integrates Voice Response Units (VRUs) with the DEFINITY switch. Specifically, the command allows voice response scripts to be executed while the call remains in queue, and it allows the passing of data between the DEFINITY switch and the VRU.
- **Disconnect** ends treatment of a call and removes the call from the switch. The command also allows the optional assignment of an announcement that will play immediately before the disconnect.
- **Goto Step** is a branching step that allows conditional or unconditional movement to a preceding or succeeding step in the vector. Conditional branching is determined by a number of factors (for example, number of calls queued in the split, number of staffed agents in the split, etc.).
- **Goto Vector** is a branching step that allows conditional or unconditional movement to another vector. Conditional branching is determined by a number of factors (for example, number of calls queued in the split, number of staffed agents in the split, etc.).

- **Messaging Split** allows the caller to leave a message for a specified extension or the VDN extension (default).
- **Queue-to** unconditionally queues a call to a split or skill and assigns a queuing priority level to the call in case no agents are available. A call sent with this command either connects to an agent in the split or skill or enters its queue.
- **Reply-best** returns data to another switch in response to a status poll. *Reply-best* is only used in status poll vectors in multi-site Best Service Routing applications.
- **Route-to Digits** routes the call to the destination specified by a set of digits collected from the caller by the previous *collect digits* step.
- **Route-to Number** routes the call to the destination specified by the administered digit string.
- **Stop** terminates the processing of any subsequent vector steps.
- **Wait-Time** is used to specify whether the caller will hear ringback, system music, silence, or an alternate audio or music source while the call is waiting in queue. The command also delays the processing of the next vector step by the specified delay time that is included in the command's syntax.

NOTE:
Complete operation details for the *route-to* commands are included in Appendix H.

Condition Testing within the Commands

As was mentioned in the previous section, a number of the Call Vectoring commands are implemented according to a tested condition that comprises part of the command. In other words, for example, if the condition expressed in the command is true, the command action is executed. On the other hand, if the condition expressed in the command is false, the command action is *not* implemented, and the next vector step is processed.

The following list provides a set of conditions that might comprise the conditional portion of a Call Vectoring command. The available set of conditions is dependent upon the optional features that have been enabled. See Appendix M, "Feature Availability" for more information. Refer to Appendix A for the syntax of each condition.

- Number of staffed agents in a split (explained earlier in this chapter)
- Number of available agents in a split (explained earlier in this chapter)
- Number of calls queued at a given priority to a split
- Amount of time that the oldest call has been waiting in a split
- Average Speed of Answer for a split or a VDN

- Expected Wait Time for a split or for a call that has entered vector processing
- Reduction in Expected Wait Time if a call is queued to a backup resource
- The number of calls in a queue that are eligible for interflow processing (using *interflow q-pos*)
- Number of calls active in a VDN
- Caller identity (ANI)
- Type of originating line (ii-digits)
- Digits entered by the caller, sent in an ISDN message from the network (CINFO), or received from an ASAI or VRU adjunct
- Time-of-day and day of the week that the call is placed

NOTE:
The syntax for this condition can be illustrated by a couple of examples, as follows: *mon 8:01 to fri 17:00* means “anytime between 8:01 A.M. Monday through 5:00 P.M. Friday,” and *all 17:00 to all 8:00* means “between 5:00 P.M. and 8:00 A.M. on any day of the week.”

Depending upon the condition, specific comparison operators ([for example, < (less than), > (greater than), = (equal to), <= (less than or equal to), >= (greater than or equal to), <> (not equal to)]) and a threshold (that is, a range of accepted numerical entries) might be in effect.

The chapters on the Call Vectoring features illustrate condition checking in more detail.

Basic Call Vectoring

3

Introduction

Basic Call Vectoring allows you to program the type of treatment a telephone call receives. You can program accordingly by using a set of vector commands.

Vector commands can direct calls to various destinations, such as adjuncts and splits. The commands can also direct calls to various treatments, such as announcements, a forced disconnect, a forced busy, or a delay treatment.

Command Set

Table 3-1 illustrates the commands used for Basic Call Vectoring.

Table 3-1. Basic Call Vectoring Command Set

Command Category	Action Taken	Command
TREATMENT	Play an announcement.	*announcement*
	Delay with audible feedback of silence, ringback, system music, or alternate audio/music source.	*wait-time*
	Play a busy tone and stop vector processing.	*busy*
	Disconnect the call.	*disconnect*
	Execute a Voice Response Unit (VRU) script.	*converse-on split*

Continued on next page

Table 3-1. Basic Call Vectoring Command Set

Command Category	Action Taken	Command
ROUTING	Queue the call to an ACD split.	*queue-to split*
	Queue the call to a backup ACD split.	*check split*
	Leave a message.	*messaging split*
	Route the call to a number programmed in the vector or to a Service Observing Feature Access Code.	*route-to number*
	Send to an adjunct a message requesting routing instructions for the call.	*adjunct routing*
BRANCHING/ PROGRAMMING	Go to a vector step.	*goto step*
	Go to another vector.	*goto vector*
	Stop vector processing.	*stop*

Functions and Examples

Basic Call Vectoring allows the user to employ a number of different functions to process telephone calls. These functions include the following:

- Providing call treatments
- Routing calls
- Branching/Programming

The following sections explain these functions.

Providing Call Treatments

In this guide, the term treatment is used to indicate the type of feedback the caller receives if the caller is not immediately connected to an agent, or if the call center is too busy or not in operation. Basic Call Vectoring provides several types of treatment, as follows:

- Announcements
- Delays with audible feedback
- Delays for queueing faxes and e-mails
- Busy tone
- Disconnect

- Voice Response Scripts

The sections that follow explain these treatments.

Announcements

If a caller is not able to connect to an agent immediately, it is logical to provide the caller with a recorded message in order to accomplish one of the following, depending upon the circumstances:

- Encourage the caller to continue to hold the line.
- Provide the caller with information that will satisfy his or her needs, thereby keeping him or her from waiting a long time for service and also allowing him or her to hang up as soon as possible.

Such a recorded message is referred to as an *announcement*, and it is provided via the *announcement* command.

Whenever a call is connected to an announcement, any previous treatment is discontinued, and answer supervision is sent (unless it has already been provided). If, during an announcement, the call is moved from waiting in a split's queue to alerting or connecting to an agent's station, the announcement is disconnected, and the caller hears ringback. When the announcement completes and is disconnected, the caller hears silence until either a vector step with alternate treatment is processed or the call reaches an agent's station.

Announcements can be classified into three groups, as follows:

- Delay announcements
- Forced announcements
- Information announcements

Delay Announcements

Screen 3-1 shows an example of a delay announcement.

```
announcement 2556 (''All our agents are busy.
Please hold.'')
```

Screen 3-1. Delay Announcement

If the caller does as suggested but ends up waiting an appreciable amount of time without receiving further feedback, he or she may tire of waiting and hang up. To keep the caller on the phone at least a little longer, a supplementary delay announcement similar to Screen 3-2 might be used:

```
announcement 2557 (‘‘Thanks for holding.  All
our agents are still busy.  Please hold.’’)
```

Screen 3-2. Supplementary Delay Announcement

A delay announcement is usually coupled with a delay step, which is provided by the *wait-time* command (discussed later).

The customer should incorporate as many supplementary delay announcements as he or she deems necessary, given the resources available.

Forced Announcements

There are times when the customer may find it advantageous to have the agents not answer calls. Usually, this option is exercised whenever the customer anticipates a barrage of calls concerning an emergency or a service problem of which the customer is already aware. Accordingly, the customer can incorporate an appropriate announcement as the very first step in the vector. Such an announcement is referred to as a *forced announcement* as illustrated in Screen 3-3.

```
announcement 1050 (‘‘We are aware of the current
situation and are working to rectify the problem.  If your
call is not urgent, please call back later.’’)
```

Screen 3-3. Forced Announcement

Information Announcements

Under certain circumstances, the customer may find it necessary to provide the caller with recorded information that, by its very content, resolves a problem with such finality that the caller feels no need to follow up on his or her call. Such a recorded message is referred to as an *information announcement*, as illustrated in Screen 3-4.

```
disconnect after announcement 2918 (‘‘Today has
been declared a snow day.  Please report for work tomorrow
at 8 A.M.’’)
```

Screen 3-4. Information Announcement

Note that the *disconnect* command is used with the announcement. After the announcement, the caller is disconnected, since he or she need not stay on the line any longer.

Delays with Audible Feedback

In presenting an example of a delay announcement earlier in this chapter, we mentioned that this type of announcement is usually coupled with a delay step. A delay step is provided by the *wait-time* command, which allows the caller to remain on hold for at least the amount of time indicated in the command.

Let's take another look at our delay announcement. However, this time, let's couple the announcement with a delay step as illustrated in Screen 3-5.

```
announcement 2556 (''All of our agents are busy.
Please hold.'')
wait-time 20 seconds hearing music
```

Screen 3-5. Delay with Audible Feedback

Here, the caller is allowed to wait at least 20 seconds for the call to be answered by an agent. During this wait period, the caller is provided with system music, which is one type of feedback available via the *wait-time* command.

If the delay step is the final effective step in the vector, the audible feedback continues beyond the specified duration. (A "final effective step" in a vector is either the last vector step or a vector step that is followed by a *stop* step.) Under normal circumstances, the audible feedback continues until the call is either answered or abandoned. However, if the call is not queued when vector processing stops, the call is dropped. Feedback also continues while a call is queued to a converse split, that is, any split routed to by a converse-on split command, and while data is being passed to a Voice Response Unit (VRU) (see the "Voice Response Scripts" section later in this chapter). Finally, feedback also continues during the wait period before the connection of an announcement and/or a Touch-Tone Receiver (TTR). (TTRs are used in conjunction with the Call Prompting feature and are discussed in Chapter 4.)

Multiple Audio or Music Sources on Delay

You can specify an alternative audio or music source for a vector *wait-time* step. This alternative source can be any extension number that is administered on the Announcements/Audio Sources form. For instructions for entering an audio or music source on this form see *DEFINITY Enterprise Communications Server Release 6 Administration and Feature Guide,* 555-230-522.

With the Multiple Audio/Music Sources feature, you can tailor the *wait-time* feedback to the interests, tastes, or requirements of the audience. You can provide specific types of music or music with overlays of advertising that relate to

the service provided by the splits or skills that the vector serves. Or, additional advertising messages can be heard by the caller as they wait for an available agent.

Let's look again at our delay announcement. However, this time let's assign an alternate audio/music source to the *wait-time* step as illustrated in Screen 3-6.

```
announcement 2556 (''All of our agents are busy.
Please hold.'')
wait-time 20 seconds hearing 55558 then music
```

Screen 3-6. Delay with Multiple Audio/Music Source Feedback

When the *wait-time* step is encountered, the caller is connected to extension 55558 for 20 seconds. At the end of 20 seconds the next vector step is executed. The "then" option in the *wait-time* step specifies what the caller will hear if the caller cannot be connected to the specified source. Or, when the call is waiting in queue, the "then" option specifies what the caller will hear if the call is not answered in 20 seconds. In this example, if the call is not answered in 20 seconds, the caller will hear system music until a subsequent *announcement, busy, collect, converse-on, disconnect* or *wait-time* step is encountered.

You can specify *music* (system music), *ringback*, *silence*, or *continue* for the "then" option. When continue is specified, the caller continues to hear the alternative audio or music source until it is replaced by a subsequent vector step regardless of the time specified in the *wait-time* step.

You can use alternate audio/music sources in vector loops to provide continuous audible feedback as illustrated in Screen 3-7.

```
1. ...
2. ...
3. ...
4. wait-time 30 secs hearing 55558 then continue
5. route-to number 913034532212 with cov n
6. goto step 4 if unconditionally
```

Screen 3-7. Delay with Continuous Audible Feedback

In Screen 3-7 a lookahead call attempt is placed every 30 seconds on behalf of the caller. If extension 55558 is a long, barge-in, repeating announcement, the caller hears announcement 55558 all the way to the end without the announcement being restarted each time vector processing returns to step 4.

Delays for message vectors

With Lucent's Message Care™ product, DEFINITY ECS Automatic Call Distribution systems can handle messages such as faxes and e-mail along with traditional voice calls. Using ASAI's Phantom Call capability, the system generates a call for each message. This call can then be routed and processed like a voice call. When the call is connected to an agent, the associated e-mail or fax may be displayed on the agent's computer.

Unlike voice calls, e-mails and faxes don't have to be handled in real-time. For vectors that handle non-real-time messages, the *wait-time* command allows you to suspend vector processing up to 8 hours. Screen 3-8 shows a simple vector for such an application.

```
1. queue-to main skill 5 pri h
2. wait-time   8 hrs hearing silence
3. disconnect after announcement none
```

Screen 3-8. 8-Hour Delay with No Feedback

This vector queues the ASAI phantom call corresponding to each message for 8 hours. If the call isn't answered in that time, step 3 disconnects the call.

If you need to queue a call for more than 8 hours, chain vectors together as shown in Screen 3-9 below. Such a chain could be used for processing e-mails and faxes received outside of business hours.

```
VECTOR 1

1. queue-to main skill 5 pri l
2. wait-time   8 hrs hearing silence
3. route to 41112

VECTOR 2 (extension 41112)

1. queue-to main skill 5 pri m
2. wait-time   8 hrs hearing silence
3. route to 41113

VECTOR 2 (extension 41113)

1. queue-to main skill 5 pri m
2. wait-time   8 hrs hearing silence
3. disconnect after announcement none
```

Screen 3-9. Chaining Vectors to Queue Calls for More than 8 Hours

If the call isn't answered in the first 8 hours, it's routed to Vector 2 where it is queued again to the same skill with higher priority. Vector 2 routes the call to a third VDN/vector pair after 8 hours.

This technique has another advantage. CentreVu CMS cannot track durations greater than 8 hours. If you want to queue non-real-time calls for more than 8 hours and you want to track each call's time-in-queue through CMS, a vector chain makes this possible. In the chain shown in Screen 3-9, any call processed by vector 2 will have waited 8 to 16 hours. Any call processed by vector 3 will have waited 16 to 24 hours. The VDN reports thus allow you to easily track calls with very long wait times.

CAUTION:
Always follow a wait-time step with a vector command that dequeues calls, such as a route-to or disconnect step. Use vector chains as described above for calls that may need to queue for more than 8 hours.

NOTE:
The Vector Disconnect Timer on the Feature-Related System Parameters form has a maximum setting of 240 minutes. In order for calls to continue in vector processing for more than 240 minutes, set the timer to 0. This effectively turns the timer off.

Multiple Music Sources on Hold

This feature uses the tenant partitioning TN (tenant number) to determine which music source to use when a call is put on hold. You can assign a different music source to each possible TN (up to 100 on G3r). For more information, see Tenant Partitioning in the *DEFINITY ECS Administration and Feature Description Guide*, 555-230-522.

The COR setting of the station/extension putting the call on hold determines whether music-on-hold is applied.[1] The TN assigned to the destination extension number determines the music source. You assign a music source number to the TN via the Tenant form. You assign the physical location (port) for the music source via the Music Sources form.

Tthe TN assigned to the active VDN via the VDN form determines the music source used. During vectoring, a *wait hearing music* command attaches the vector delay music source defined by the TN for the active VDN. A *wait hearing extension then...* command (the Multiple Music Sources for Vector Delay) applies the vector delay source defined by the Announcements/Audio Sources extension regardless of the TN assigned to the VDN.

1. With EAS, DEFINITY uses the COR of the logical agent ID (not the physical extension) to determine the music source.

NOTE:
The TN administered for extensions on the Announcement/Audio Sources form only applies to direct calls to the announcement extension. For these calls, the announcement or music source assigned to the TN is what the caller hears.

During vector processing, if the *converse* vector command connects the call to an agent (when the call remains under vector control) and the agent puts the call on hold, the active VDN applies music-on-hold.

When a vector routes a call to another destination (in other words, the call leaves vector processing via a *queue, check, route-to*, or *messaging split* command), DEFINITY uses the TN of the last active VDN to determine the music source for music-on-hold.

In ACD systems without vectoring, the TN assigned to the called hunt group extension determines which music source callers hear while in queue or on hold (if music-on-hold applies).

Busy Tone

A busy tone and subsequent termination of vector processing are produced via the *busy* command. An exception to this occurs on CO trunks where answer supervision has not been sent. Callers on such trunks do not hear the busy tone from the switch. Instead, these callers continue to hear ringback from the CO. The *busy* command eventually times out and drops the call after 45 seconds. With ISDN PRI, busy tone can be provided from the network switch.

You might want to force a busy tone to process a call that arrives at a time when there is a large number of calls queued in the main split, or when the call center is out of service or closed.

The vector in Screen 3-10 illustrates how you can use the *busy* command:

```
1. goto step 6 if calls-queued in split 1 pri h > 30
2. queue-to split 1 pri h
3. announcement 4000
4. wait-time 2 seconds hearing music
5. stop
6. busy
```

Screen 3-10. Providing Busy Tone

In this vector, the *goto step* command in step 1 sends call control to *busy* in step 6 if the conditions in the former command are met. Specifically, if the number of calls queued at a high priority is greater than 30, the *busy* command is accessed.

Disconnect

You can opt to have a call disconnected by incorporating the *disconnect* command. However, be careful to use good judgement in this regard lest you unnecessarily infuriate the caller. Also, as a courtesy to the caller, an announcement should be given to the caller before he or she is disconnected under any circumstances.

The *disconnect* command itself has a built-in announcement option. We saw an example of the command when we were discussing information announcements earlier in this chapter. Screen 3-11 shows the example again:

```
disconnect after announcement 2918 (‘‘Today has
been declared a snow day.  Please report for work tomorrow
at 8 P.M.’’)
```

Screen 3-11. Disconnecting a Call

This example presents an ideal use of the *disconnect* command. The caller is given recorded information that, by its very content, resolves a problem so that the caller feels no need to follow up on his or her call.

Voice Response Scripts

Voice Response Integration (VRI) is designed to enhance the integration of DEFINITY call center systems and to integrate DEFINITY Call Vectoring with the capabilities of voice response units (VRUs), particularly the CONVERSANT Voice Information System.

VRI can do the following:

- Execute a VRU script while retaining control of the call in the DEFINITY System vector processing.
- Execute a VRU script while the call remains in the split queue and retains its position in the queue.
- Pool CONVERSANT ports for multiple applications (which previously was possible only when ASAI was present).
- Use a VRU as a flexible external announcement device.
- Pass data between the DEFINITY switch and a VRU.
- Tandem VRU data through the DEFINITY switch to an ASAI host.

The capabilities in the previous list are provided by the *converse-on split* command, which is an enhancement to the Basic Call Vectoring customer option. The *converse-on split* step is specifically designed to integrate a VRU with the DEFINITY System. VRI allows VRU capabilities to be used while keeping control

of the call in the DEFINITY System. The inclusion of VRUs with vector processing provides the following advantages:

- Access to local and host databases.
- Validation of caller information.
- Text to speech capabilities.
- Speech recognition.
- Increased recorded announcement capacity.
- Audiotex applications.
- Interactive Voice Response (IVR) applications.
- Transaction processing applications.

One of the advantages of VRI is that it allows users to make more productive use of queuing time. For example, while the call is waiting in queue, the caller can listen to product information by using an audiotex application or by completing an interactive voice response transaction. In some cases, it may even be possible to resolve the customer's questions while the call is in queue. This can help reduce the queuing time for all other callers during peak intervals.

In addition, when Advanced Vector Routing is enabled, the Expected Wait Time for a call can be passed to the VRU. In this way, the caller can be told how much longer they can expect to wait before their call will be answered. See "Expected Wait Time (EWT)" on page 5-2 for a complete description of the EWT feature.

During the execution of a VRU script, if the caller previously queued to an ACD split, the caller retains his/her position in queue. If an agent on the DEFINITY switch becomes available to service the call, the line to the VRU is immediately dropped, and the calling party is connected to the available agent.

Screen 3-12 shows an example of a vector that can access voice response scripts from a VRU.

NOTE:
Recall that one or more VDNs can access the same vector. This capability is appropriate for the following example.

```
VDN (extension=1040   name=''car loans''     vector=40)
VDN (extension=1041   name=''equity loans''  vector=40)
Vector 40
     1. goto step 10 if calls-queued in split 1 pri h > 30
     2. queue-to split 1 pri h
     3. announcement 4000
     4. goto step 7 if calls-queued in split 1 pri h < 5
     5. wait-time 0 seconds hearing music
     6. converse-on split 11 pri h passing vdn and none
     7. wait-time 20 seconds hearing music
     8. announcement 4001
     9. goto step 7 if unconditionally
    10. busy
```

Screen 3-12. Accessing Voice Response Scripts

For this example, let's suppose first that a caller would like to hear information concerning car loans. Let's also assume the call is queued to split 1 (step 2) and that vector processing proceeds to step 6. In such a case, the *converse-on split* command in this step delivers the call to the converse split if there is a queue for the split and the queue is not full, or if a VRU port is available. (Otherwise, vector processing continues at the next vector step.) When the VRU port responds, the step then outpulses VDN 1040 to the VRU via the *passing vdn* parameter included in the command. In turn, the VRU executes the "car loans" voice response script for the caller. Note that it is important to provide a feedback step prior to the converse-on step in case there is a delay in reaching an available converse split port. In this example step 5 provides music for this purpose.

Now, let's suppose another caller wants information concerning equity loans. In such a case, if everything proceeds according to form, VDN 1041 is outpulsed to the VRU, which in turn executes the "equity loan" voice response script for the caller.

In either case, while interaction with the VRU is taking place, the call remains in the appropriate split's queue (split 1 in this example). If an agent answers the call while the voice response script is being executed, the voice response script is interrupted, the line to the VRU is dropped, and the caller is connected to the available agent. Once a voice response script starts, no further vector steps are executed until the voice response script is completed.

NOTE:
Refer to Appendix I for a detailed explanation of the call flow for converse—VRI calls.

Besides VDN extensions, the *converse-on split* command can outpulse to the VRU calling party extensions, collected (inputted) caller digits (if Call Prompting is enabled), Expected Wait Time (if Advanced Vector Routing is enabled) call queue positions, a string of a maximum of six digits or asterisks, a pound sign (#), or nothing. Further details are included in Chapter 4, "Call Prompting",

Chapter 5, "Advanced Vector Routing" and in Appendix A, "Call Vectoring Commands".

NOTE:
In Screen 3-12, the *calls-queued* condition in the second *goto* step (step 4) in effect serves as a checkpoint for determining whether or not there is enough time for the voice response script (activated by the *converse-on* step) to be executed. Specifically, if five or more calls are queued to split 1, it is considered feasible to execute the voice response script.

Routing Calls

Basic Call Vectoring offers several means of routing telephone calls, as follows:

- Queuing calls to ACD splits
- Leaving recorded messages
- Sending calls to a vector-programmed number (that represents an internal or external destination)
- Routing calls to a Feature Access Code that will activate Service Observing
- Adjunct Routing

The following sections discuss these routing procedures (except for adjunct routing, which is fully described in Chapter 10).

Queuing Calls to ACD Splits

Calls that come into the Call Vectoring system can be queued to a maximum of three ACD splits. Two commands are used to queue calls to splits.

The *queue-to split* command queues a call *unconditionally.* The command sends a call to a split and assigns a queuing priority level to the call in case all agents are busy.

The *check split* command *conditionally* checks the status of a split for possible termination of the call to that split. The command either connects the call to an agent in the split or puts the call into the split's queue (at the specified priority level) if the condition specified as part of the command is met.

Multiple Split Queuing

The term *multiple split queuing* refers to the queuing of a call to more than one split at the same time. The following vector, Screen 3-13, helps to illustrate this process.

```
1. goto step 4 if calls-queued in split 1 pri l >= 10
2. queue-to split 1 pri t
3. wait-time 12 seconds hearing ringback
4. check split 2 pri m if calls-queued < 5
5. check split 3 pri m if calls-queued < 5
6. announcement 3001
7. wait-time 50 secs hearing music
8. goto step 4 if unconditionally
```

Screen 3-13. Multiple Split Queuing

To avoid completing vector processing without queuing the call to a split, it is always good practice to check a split's queue before queuing to that split. If the queue is full, alternate treatment (such as queuing to an alternate split) should be provided. In this vector, if the main split's queue (which has 10 queue slots) is full, the *goto step* command in step 1 skips the main split and goes directly to step 4 to check the backup splits. Although calls are queued in step 2 at a top priority, a low priority is specified in step 1 so that calls in queue at all priority levels are counted. If there are 10 or fewer calls in the main split, control is passed to step 2, where the *queue-to split* command queues the call to split 1. Once the call is queued, vector processing continues at the next step.

Step 4 contains a *check split* command. (Recall that in the last paragraph we mention that this step is branched to if the main split queue is holding 10 or more calls.) If the call is not answered by the time step 4 is reached, the *check split* in the step attempts to queue the call to a second split. Specifically, the command first determines whether there are fewer than 5 calls queued to split 2. If so, the command then attempts to connect the call to an agent in the split. If such a connection cannot be made, the command puts the call into the split's queue (at the specified priority level). Vector processing then continues at the next step. On the other hand, if there are 5 or more calls queued to split 2, the command fails, and vector processing continues at step 5.

Step 5 contains another *check split* command and, accordingly, the process described in the previous paragraph is repeated, with one difference: the queuing attempt is made to split 3 instead of to split 2.

Except for the condition check, the circumstances under which the *check split* command cannot queue a call are identical to those for the *queue-to split* command.

Finally, note that whenever a call is queued to a backup split, the call remains queued to the main split and/or to another backup split (if already queued to either or both of these splits). Once the call is answered in a split to which it is queued, the call is automatically removed from all the other split(s) to which it is also queued.

NOTE:
The *check split, queue-to split,* and converse-on commands can access *only* those splits that are “vector-controlled.” A split is considered “vector-controlled” if *yes* is entered in the Vector field of the Hunt Group form. With Expert Agent Selection (EAS) activated, Multiple Split Queuing becomes Multiple Skill Queuing.

Leaving Recorded Messages

Basic Call Vectoring allows the caller to leave a message for the customer if the agents at the customer site are not available to take telephone calls. This is done with the help of the *messaging split* command. Let’s take a look at the example in Screen 3-14.

```
 1. goto step 8 if time-of-day is all 16:30 to all 7:30
 2. goto step 10 if calls-queued in split 47 pri l >= 20
 3. queue-to split 47 pri m
 4. wait-time 12 secs hearing ringback
 5. announcement 4001
 6. wait-time 60 secs hearing music
 7. goto step 5 if unconditionally
 8. announcement 4111(‘‘We’re sorry, our office
    is closed.  If you’d like to leave a message, please
    do so after the tone.  Otherwise, please call back
    weekdays between 7:30 A.M. and 4:30 P.M.  Thank you.’’)
 9. goto step 11 if unconditionally
10. announcement 4222 (“We’re sorry, all of our agents are busy,
    please leave a message after the tone and we will return your
    call.”)
11. messaging split 18 for extension 2000
12. disconnect after announcement 4333 (‘‘We’re sorry, we are
    unable to take your message at this time.  Please
    call back at your convenience weekdays between
    7:30 A.M. and 4:30 P.M.  Thank you.’’)
                        .
                        .
13. busy
```

Screen 3-14. Leaving Recorded Message

In this vector, the *goto step* command in step 1 checks to see if the office is open, and branches to step 8 if the office is closed. This is done to accommodate calls that are made during non-working hours, when there are no agents available to take telephone calls. Accordingly, step 8 provides the caller with an appropriate announcement and an opportunity to leave a recorded message.

Step 2 checks to see if split 47’s queue (which has 20 queue slots) is full, and branches to step 10 if it is. Steps 3 to 7 queue the call to split 47 and then give audible feedback to the caller.

If the caller chooses to leave a message, the *messaging split* command in step 11 is executed. Split 18 in the command is the Audio Information Exchange

(AUDIX) split. AUDIX is a voice mail adjunct that allows a customer to record, edit, store, forward, and retrieve voice messages to and/or from callers. Extension 2000 is the mailbox for split 47 (from step 2).

Upon execution of the *messaging split* command, an attempt is made to connect the caller to AUDIX so he or she can leave a recorded message. If the split queue is full, or if the AUDIX link is down, termination to AUDIX is unsuccessful, and vector processing continues at the next vector step, which (as is the case here) usually contains an announcement that provides the caller with the appropriate apology and subsequent directives. If the caller is successfully connected to AUDIX, vector processing terminates, and a message may be left for the specified mailbox (2000, in this case).

Finally, if the supervisor or a group of agents has an Automatic Message Waiting (AMW) Lamp for the mailbox used, and if the lamp lights, the relevant party, upon returning, knows a caller has left an AUDIX message.

Option with the VDN as the Coverage Point

Recall from Chapter 2 that the Vector Directory Number (VDN) can be used as the last point in a coverage path. This capability allows the call to first go to coverage and to then be processed by Call Vectoring and/ or Call Prompting. The capability also allows you to assign AUDIX or the Message Server to a vector-controlled hunt group and to therefore enable access to these servers via a *queue-to split* or *check split* command. The result of all this is that call handling flexibility is enhanced.

Screen 3-15 shows a vector, for which the VDN serves as a final coverage point, that allows the caller to leave a recorded message.

```
VDN 1 (used in a coverage path)
Vector 1
    1. goto step 7 if time-of-day is mon 8:01 to fri 17:00
    2. goto step 13 if staffed-agents in split 10 < 1
    3. queue-to split 10 pri 1 (AUDIX split)
    4. wait-time 20 seconds hearing ringback
    5. announcement 1000 (''Please wait for voice
       mail to take your message.'')
    6. goto step 4 if unconditionally
    7. goto step 2 if staffed-agents in split 20 < 1
    8. queue-to split 20 pri 1 (message server split)
    9. wait-time 12 seconds hearing ringback
   10. announcement 1005 (''Please wait for an attendant
       to take your message.'')
   11. wait-time 50 seconds hearing music
   12. goto step 10 if unconditionally
   13. disconnect after announcement 1008 (''We cannot
       take a message at this time.  Please call back tomorrow.'')
```

Screen 3-15. Leaving Recorded Messages (VDN as the coverage point option)

In steps 3 and 8 of the vector, the caller is given the option of leaving a recorded message. However, in accord with our discussion at the beginning of this section, the *queue-to split* command instead of the *messaging split* command is used in each case. The advantage here is that the call is actually *queued* to the AUDIX split or to the message server split. On the other hand, a *messaging split* command does not queue the call to the split; instead (if successful), it simply connects the caller to the split so the caller may leave a message for the specified extension. However, termination to the split may turn out to be unsuccessful due to a factor that cannot be checked by vector processing. (For example, the AUDIX link might be down, or all AUDIX ports might be out of service.)

As a result of the queuing process, a wait-announcement loop can be included after each *queue-to split* step, and the appropriate loop can be executed until the call is actually terminated to either an AUDIX voice port or to an available message service agent. In this vector, steps 4 through 6 comprise the first wait-announcement loop, and steps 10 through 12 comprise the second such loop.

Sending Calls to a Vector-Programmed Number

Earlier in this chapter, we mentioned calls can be queued to a maximum of three splits. Calls can also be routed to a programmed number in the vector via a process known as *interflow*.

Interflow

Interflow is a process that allows calls that are directed or redirected to one split to be redirected to an internal or an external destination. For Basic Call Vectoring, this destination is represented by a number programmed in the vector. The number is always included in the *route-to number* command, and it may represent any of the following destinations:

- Attendant (or attendant queue)
- Local extension
- Remote (that is, UDP) extension
- External number
- VDN

The following vectors in Screen 3-16 illustrate how interflow is used.

```
VDN (extension=1000   name=''Billing Service''  vector=55)
Vector 55:
    1. announcement 3001
    2. goto step 8 if oldest call-wait in split 1 pri l > 120
    3. goto step 8 if calls-queued in split 1 pri l > 10
    4. queue-to split 1 pri t
    5. wait-time 50 seconds hearing music
    6. announcement 3002
    7. goto step 5 if unconditionally
    8. route-to number 2020 with cov n if unconditionally

VDN (extension=2020 name=''Message Service''  vector=100)
Vector 100:
    1. announcement 3900 (''We're sorry, all our
       agents are busy. Please leave a message. Thank you.'')
    2. messaging split 18 for extension 3000
    3. disconnect after announcement 2505 (''We cannot
     take a message at this time. Please call back tomorrow.'')
```

Screen 3-16. Call Interflow

In the first vector, a branch is made to step 8 from step 2 if the condition in the latter step (*oldest call-wait in split 1 > 120 seconds*) is true. If the condition is false, a branch is made to step 8 from step 3 if the condition in the latter step (*calls-queued in split 1 > 10*) is true. If that condition is also false, the call is queued (step 4), and a wait-announcement loop becomes effective (steps 5 through 7).

If a successful branch to step 8 is made from step 2, the *route-to number* command is executed. The destination number (2020) in this particular command is a VDN. Accordingly, vector processing terminates in the first vector and begins at the first step of the second vector, to which the VDN points.

Once processing control is passed to the second vector, the caller is provided with the appropriate announcement (step 1). Thereafter, upon execution of the *messaging split* command in step 2, the system attempts to either queue the call to the message service split or else terminate the call to a message service agent or to an AUDIX voice port. If one of these attempts succeeds, the caller may leave a message. If none of the attempts succeed, the command fails, and vector processing continues at the next vector command (usually an announcement explaining that the necessary connection could not be made).

Service Observing

For a complete description of Service Observing see the *DEFINITY Enterprise Communications Server Release 6 Administration and Feature Guide,* 555-230-522.

Service Observing vectors allow users to observe calls either from a remote location or a local station. A Service Observe button is not required. The use of a

Service Observing vector limits users to listen-only or listen-talk observing. The observer cannot toggle between the two states.

Service Observing vectors can be used to observe physical extensions, EAS logical agent LoginIDs, and VDNs.

The calling permissions of the COR assigned to the Service Observing VDN in conjunction with the "can be observed" settings of the COR assigned to the destination determine what agents, terminals, or VDNs can be observed. For additional information about the security requirements with Service Observing vectors see Appendix J, "Security Issues".

You can construct Service Observing vectors in one of four ways. Vectors can route calls to:

1. A Service Observing FAC.
2. The Remote Access extension using Call Prompting to test against a user-entered security code.
3. A Service Observing FAC and extension entered by the user with Call Prompting enabled.
4. One of several Service Observing FACs and extensions programmed into route-to number vector steps. In this case Call Prompting can be used to allow the observer to select the extension to be observed.

The first vector type is discussed below. See Chapter 4, "Call Prompting" for examples of Service Observing vectors that use Call Prompting.

Service Observing FAC Vector

The vector in Screen 3-17 connects the user to a Service Observing FAC. Be aware that this vector does not provide security checks and should be used with great care and only in situations where security is not a concern.

```
1. wait-time 0 secs hearing ringback
2. route-to number #12 with cov n if unconditionally (Listen-only FAC)
3. busy
```

Screen 3-17. Vector for Service Observing FAC

In Screen 3-17 the caller is connected to a listen-only Service Observing FAC. Once connected, the user must dial the extension number to be observed. To observe in a listen/talk mode, the observer would dial a different VDN.

Branching/Programming

Basic Call Vectoring provides several programming methods that affect the processing flow within the vector. These methods, which are implemented via Call Vectoring commands, include the following:

- Unconditional branching
- Conditional branching
- Stopping vector processing

The following sections explain these programming methods.

Unconditional Branching

Unconditional branching is a method that always passes control from the current vector step to either a preceding or subsequent vector step or to another vector. This type of branching is enabled via the *goto step* and *goto vector* commands, each with a condition of *unconditionally* assigned.

Unconditional branching is illustrated in the following vector, Screen 3-18.

```
1. goto step 8 if calls-queued in split 3 pri m > 10
2. queue-to split 3 pri m
3. wait-time 12 seconds hearing ringback
4. announcement 3001
5. wait-time 30 seconds hearing music
6. announcement 3002
7. goto step 5 if unconditionally
8. busy
```

Screen 3-18. Unconditional Branching

The unconditional branch statement in step 7 establishes an apparent "endless loop" involving steps 5 through 7. The loop, however, really is not endless, since vector processing terminates if an agent answers the call. Vector processing also terminates when the system recognizes the caller has abandoned the call.

Conditional Branching

Conditional branching is a method that *conditionally* passes control from the current vector step to either a preceding or subsequent vector step or to a different vector. This type of branching is enabled via the *goto step* and *goto vector* commands, each with one of the following conditions assigned and tested: *available-agents, staffed-agents, calls-queued*, *oldest call-waiting*, or *time-of-day.* When Advanced Vector Routing is enabled, additional conditions can be tested: *rolling-asa*, *counted-calls*, *expected-wait*. See Chapter 5, "Advanced Vector Routing" for more information. When ANI and II-Digits Routing is enabled, the *ani* and *II-digits* conditions can also be tested with a *goto*

command. See, Chapter 6, "ANI /II-Digits Routing and Caller Information Forwarding (CINFO)" for more information. If the command's condition is not met, control is passed to the step that follows.

Conditional branching is illustrated in the following vector, Screen 3-19.

```
1. goto vector 100 if time-of-day is all 17:00 to all 8:00
2. goto vector 200 if time-of-day is fri 17:00 to mon 8:00
3. goto step 8 if calls-queued in split 1 pri l > 5
4. queue-to split 1 pri l
5. announcement 4000
6. wait-time 60 seconds hearing ringback
7. goto step 5 if unconditionally
8. busy
```

Screen 3-19. Conditional Branching

In this vector, a conditional branch test statement appears in steps 1, 2 and 3. If the call is placed during non-business hours (between 5:00 p.m. and 8:00 a.m.) on any day of the week, the *goto vector* command in step 1 routes the call to vector 100. However, if the call is placed during business hours, control is passed to step 2, where the *goto vector* command there checks whether the call is placed during the weekend. If this is the case, the call is routed to vector 200. If not, control is passed to step 3, where the *goto step* command checks for the number of calls queued to the main split. If the number of calls is greater than 5, control is passed to *busy* in step 8. If the number of calls is 5 or less, the call is queued (step 4). Thereafter, an announcement-wait cycle (steps 5 through 7) is implemented until an agent answers the call or the call is abandoned.

Stopping Vector Processing

Basic Call Vectoring provides a specific command that stops vector processing. The *stop* command halts the processing of any subsequent vector steps. If a call is not queued when vector processing stops, the call is dropped and tracked as an "abandon" by the Call Management System (CMS) and/or BCMS. After the *stop* command is processed, any calls that are already queued remain queued, and any wait treatment (silence, ringback, system music, or alternate audio/music source) is continued.

The following vector, Screen 3-20, illustrates how vector processing is stopped via the *stop* command.

```
1. goto step 6 if calls-queued in split 21 pri m > 10
2. queue-to split 21 pri m
3. announcement 4000
4. wait-time 30 seconds hearing ringback
5. stop
6. busy
```

Screen 3-20. Stopping Vector Processing

If the *stop* command is reached, the queued caller will continue to hear ringback. Also, if the *stop* command in step 5 is executed, step 6 is not executed immediately thereafter. The latter step can be executed only if the *goto* command in step 1 succeeds.

Note that an *implied stop* follows the last step within a vector. In addition, a vector will stop processing whenever 1,000 vector steps have been processed (3,000 for vectors using the interflow-qpos LAI conditional).

Vector Chaining

Multiple vectors can be chained together to enhance processing capabilities. In this regard, the following points involving two Basic Call Vectoring commands should be noted:

- *Route-to number*. If this command is used to point to a VDN, the following happens:

  1. Vector processing continues at the first step in the vector assigned to the routed-to VDN.
  2. Call (if queued) is dequeued.
  3. Wait treatment (if any) is disabled.

  Processing then continues in the receiving vector at step 1.

- *Goto vector*. If this command is used, the following happens:

  1. Vector processing continues at the first step in the branched-to vector.
  2. Call (if queued) remains in queue.
  3. Wait treatment (if any) is continued.

  Processing then continues in the receiving vector at step 1.

Call Prompting

4

Introduction

Call Prompting provides flexible call handling based on information collected from a calling party. This information comes in the form of dialed digits originating from an internal or external touch-tone telephone, or from an internal rotary telephone that is on the same switch as the vector. In effect, Call Prompting allows for the temporary transfer of call management control to the caller.

In addition, with Call Prompting and Vectoring (CINFO) enabled, the DEFINITY System can collect caller entered digits (ced) and customer database provided digits (cdpd) supplied by the network. The system can receive Call Information Forwarding (CINFO) digits in an incoming call's ISDN message when the AT&T Network Intelligent Call Processing (ICP) service is in use. A switch can collect digits and forward those digits to other switches via interflow commands. See "Caller Information Forwarding" on page 6-8 for more information.

With Voice Response Integration (VRI), digits may be returned to the switch by a Voice Response Unit (VRU) script accessed via a *converse-on split* command. Such digits can also be used for call management.

Call Prompting may be used in various applications to achieve a better and more flexible handling of telephone calls.

Command Set

Table 4-1 illustrates the commands used for Call Prompting.

Table 4-1. Call Prompting Command Set

Command Category	Action Taken	Command
INFORMATION COLLECTION	Collect information from the calling party, from the public network in an ISDN SETUP message, from a Voice Response Unit (VRU), or from CallVisor ASAI.	*collect digits*
TREATMENT	Play an announcement.	*announcement*
	Delay with audible feedback of silence, ringback, system music, or an alternate audio/music source.	*wait-time*
ROUTING	Leave a message.	*messaging split*
	Route the call to a number programmed in the vector.	*route-to number*
	Route the call to digits supplied by the calling party.	*route-to digits*
BRANCHING/ PROGRAMMING	Go to a vector step.	*goto step*
	Go to another vector.	*goto vector*
	Stop vector processing.	*stop*

Touch-Tone Collection Requirements

Before the DEFINITY System can accept the touch-tone digits entered by a caller, the switch must be equipped with a “collection resource.” The resource used for collecting and interpreting touch-tone digits is a unit of hardware called a Touch-Tone Receiver (TTR). These TTRs are provided on the TN744 call classifier and TN2182 tone detector, one of which is required for Call Prompting.

For new systems, the number of required TTRs is configured according to two sources, as follows:

- Customer input to the Lucent Technologies Account Team
- Account team input to the DOSS/ATTOMS configuration

For existing systems that are adding a Call Prompting application, the Account Team recommends the appropriate number of TTRs based on two factors, as follows:

- Account team input to the DOSS/ATTOMS configuration
- Application review by the Lucent Technologies Design Center

Collecting of CINFO digits does not require TTRs.

Outside callers must have a touch-tone phone to enter the digits requested via the *collect digits* command. For callers using rotary dialing, the Call Prompting timeout takes effect, the *collect digits* command times out, and vector processing continues at the next step. As a precaution, always provide a default treatment (for example, *route-to* attendant command, *queue-to split* command) in the vector script unless the script is created exclusively for users of touch-tone telephones.

NOTE:
The Call Prompting inter-digit timeout can be administered for any number of seconds from 4 to 10. This value is administered on the Feature-Related System Parameters form. See *DEFINITY Enterprise Communications Server Release 5 Implementation*, 555-230-302.

Provisions for users of rotary phones are illustrated in the vector scripts in this chapter.

Call Prompting Digit Entry

The touch-tone digits entered by a Call Prompting user are collected via the *collect digits* command. This command allows the system to collect up to 24 digits from a touch-tone phone. Sixteen of these digits may be collected immediately, while any remaining digits are stored as dial-ahead digits (explained later in this chapter).

Call Prompting allows some flexibility in entering digits. Specifically, the caller can do the following:

- Remove incorrect digits strings
- Enter variable-length digit strings
- Enter dial-ahead digits

The following sections explain these processes.

Removing Incorrect Digit Strings

You can (and probably should) include an announcement that requests the caller to enter digits. As an option, the announcement can instruct the caller to enter an asterisk (*) if he or she enters incorrect data. When the caller enters a "*," the following happens:

1. Digits collected for the current *collect digits* command are deleted.

   NOTE:
   Also deleted are any dial-ahead digits that are entered and that do not exceed the maximum digit count of 24. (Dial-ahead digits are explained later in this chapter.)

2. Digit collection is restarted.
3. Announcement is not replayed.

Once the caller enters "*," the caller can re-enter digits for processing.

Entering Variable-Length Digit Strings

The maximum number of digits requested from the caller must be specified in the administration of the *collect digits* command. In some cases, the caller might be permitted to enter fewer digits than the maximum specified. In fact, the number of digits entered by the caller can vary for several variations of one *collect digits* command. Each such grouping of digits is called a *variable-length digit string*.

Call Prompting allows for variable-length digit strings by providing an end-of-dialing indicator in the form of the pound sign (#). "#" is used to end any digit string entered by the caller, and it does the following:

- Tells the system that the caller has finished entering digits
- Causes the next vector step to be processed immediately

Whenever the caller is permitted to enter a variable-length digit string, the announcement portion of the *collect digits* command should specify the largest possible number of digits that can be entered. Accordingly, you should administer each *collect digits* command to collect no more than the intended maximum number of digits. You can have the caller enter "#" as part of a variable digit string entry either at the end of each variable digit string entered or at the end of each such string that, not counting "#," contains *fewer* characters than the maximum number of allowable digits. In the first case, "#" should be included in the count of the number of maximum digits that can be entered; in the second case, "#" should *not* be included in this count.

If the caller enters more digits than the maximum number specified, the additional digits are saved as "dial-ahead" digits for subsequent *collect digits* commands. (Dial-ahead digits are explained later in this chapter.) If the vector, or vectors chained to it, do not contain another *collect digits* command, the extra digits are discarded.

If the caller enters fewer digits than the maximum number specified *and* does not complete the entry with "#," a Call Prompting timeout occurs. The timeout terminates the command, and any digits collected prior to the timeout are available for subsequent vector processing.

A common application involving the entering of variable-length digit strings allows the user to dial either the number for the attendant or an extension (to reach the desired destination.) Let's say the maximum number of digits that can be entered is 3. In such a case, if the user wishes to reach the attendant, the user should dial "0#." However, if the user chooses to dial a 3-digit extension, the user should dial, for example, "748" and not "748#." Since the maximum number of digits that can be dialed in this case is three, dialing "748#" would cause "#" to be saved as a dial-digit (explained later in this chapter). On the other hand, if the caller dials "748#," and if the maximum number of digits that can be entered is 4, "#" is not saved as a dial-ahead digit since it is the fourth of four digits that can be entered in this case.

Entering Dial-Ahead Digits

When digit collection for the current *collect digits* command completes, vector processing continues at the next vector step. However, the switch continues to collect any digits that the caller subsequently dials until the TTR disconnects. See "Collecting Digits on the DEFINITY System" on page A-34 for more information. These "dialed-ahead" digits are saved for processing by subsequent *collect digits* commands. Dial-Ahead Digits are explained fully on page 4-14.

Functions and Examples

Call Prompting uses some of the functions found in Basic Call Vectoring. This becomes evident when you compare the command set table for Basic Call Vectoring in Chapter 3, Table 3-1 with Table 4-1, Call Prompting, found at the beginning of this chapter.

Call Prompting also provides some additional functions that involve digit processing. These functions include the following:

- Treating digits as a destination
- Using digits to collect branching information (including Vector Routing Tables)
- Using digits to select options

- Displaying digits on the agent's set
- Passing digits to an adjunct
- Creating Service Observing vectors

These functions are illustrated in the following sections.

Treating Digits as a Destination

Call Prompting allows you to route calls according to the digits collected from the caller. Once the digits are collected via the *collect digits* command, the *route-to digits* command attempts to route the call to the destination that the digits represent. The command always routes the call to the destination that is indicated by the digits processed by the most recent collect digits command.

The digits can represent any of the following destinations:

- Internal (local) extension (for example, split/hunt group, station, announcement, etc.)
- VDN extension
- Attendant
- Remote access extension
- External number, such as a trunk access code (TAC) or an Automatic Alternate Route/Automatic Route Selection (AAR/ARS) feature access code (FAC) followed by a public network number (for example, 7 digit ETN, 10 digit DDD, etc.)

Let's take a look at the vector Screen 4-1 that illustrates how a call is routed via digits that are collected from a caller:

```
1. wait-time 0 seconds hearing ringback
2. collect 5 digits after announcement 300
   ("You have reached Redux Electric in Glenrock.
   Please dial a 5-digit extension or wait for the
   attendant.'')
3. route-to digits with coverage y
4. route-to number 0 with cov n if unconditionally
5. stop
```

Screen 4-1. Treating Digits as a Destination

In this vector, the caller is prompted to enter the destination extension of the party he or she would like to reach (step 2). (The extension in this vector may contain up to 5 digits.) The vector collects the digits, then routes to the destination via the *route to digits* command in step 3.

If the *route-to digits* command fails (because the caller fails to enter any digits, or because the extension number entered is invalid), the *route-to number command* in step 4 routes the call to the attendant (default). However, as long as the destination is a valid extension, the *route-to digits* command succeeds, coverage applies, and vector processing terminates. (Even if the destination is busy, vector processing terminates because coverage call processing takes effect.)

NOTE:
From time to time, all of the system's TTRs might be in use. As a result, when you are collecting digits from a caller, you should avoid starting your main vector with a *collect digits* command, since the caller in this case receives no audible feedback if he or she has to wait for a TTR to become available. Accordingly, it is a good practice to include some treatment (for example, *wait-time 0 seconds hearing ringback*) before the initial *collect digits* step.

In addition, if calls are likely to be transferred to this vector, a wait-time step of sufficient length is recommended before the collect step to allow the transferring party enough time to complete the transfer.

Using Digits to Collect Branching Information

Call Prompting allows you to direct a call to another step or vector based on the digits entered by the caller. This branching is accomplished with a *goto* step. For example, in vector Screen 4-2 digits are used to route calls to different vectors based on an assigned customer number.

```
1. wait-time 0 seconds hearing ringback
2. collect 5 digits after announcement 200
   (“Please enter your customer number)
3. goto vector 8 if digits = 10+
4. goto vector 9 if digits = 11+
5. goto vector 10 if digits = 12+
6. route-to number 0 with cov n if unconditionally
7. stop
```

Screen 4-2. Using Digits to Collect Branching Information

The wildcard “+” indicates that the two digits can be followed by any number of additional digits (zero or more digits). So, callers with a number that begins with the digits 10 are routed to vector 8, callers with a number that begins with the digits 11 are routed to vector 9, and callers with a number that begins with the digits 12 are routed to vector 10.

Vector Routing Tables

You also can test digits against entries in a Vector Routing Table.

Vector Routing Tables contain lists of numbers that can be used to test a *goto...if digits* command. Digits collected with the collect digits step can be tested to see if they are either *in* or *not-in* the specified table. Entries in the tables can include either the "+" or "?" wildcard.

- The "+" represents a group of digits and can only be used as the first or last character of the string.
- The "?" represents a single digit. Any number of them can be used at any position in the digit string.

Tables are entered on the Vector Routing Table form. See *DEFINITY Enterprise Communications Server Release 6 Administration and Feature Description*, 555-230-522 for complete instructions for creating Vector Routing Tables.

Screen 4-3 gives an example of a Vector Routing Table.

```
                    VECTOR ROUTING TABLE
Number: 10           Name: Premier Accts          Sort? n


       1: 5734020                          17: 2679038
       2: 8910573                          18: 1345+
       3: 8738494                          19: 2345+
       4: 4385702                          20: ________
       5: 8768995                          21: ________
       6: 7867387                          22: ________
       7: 7802452                          23: ________
       8: 7074589                          24: ________
       9: 5674902                          25: ________
      10: 8789689                          26: ________
      11: 4870985                          27: ________
      12: 8093182                          28: ________
      13: 7809130                          29: ________
      14: 7890301                          30: ________
      15: 7893213                          31: ________
      16: 8743180                          32: ________
```

Screen 4-3. Vector Routing Table

The following vector Screen 4-4 could be used to test against numbers in the example in Screen 4-3.

```
      1. wait-time 0 seconds hearing ringback
      2. collect 7 digits after announcement 200
         (“Please enter your account number)
      3. goto vector 8 if digits in table 10
      4. queue-to split 5 pri l
      5. wait-time 10 seconds hearing ringback
      6. announcement 2771
      7. wait-time 10 seconds hearing music
      8. goto step 6 if unconditionally
```

Screen 4-4. Testing for Digits In Vector Routing Table

If the caller enters an account number that is listed in the Vector Routing Table, the call is routed to vector 8. Likewise, if the caller enters an account number that matches the wildcard entry (for example 1345987), the call is routed to vector 8.

If the caller enters an account number that is not listed in the Vector Routing Table, or if the caller does not enter an account number, the call is queued to split 5.

Suppose that instead of containing a list of premier accounts the Vector Routing Table contained a list of accounts with a poor payment record. The following vector Screen 4-5 only queues calls with account numbers that are not in the table. Calls in the table route to the collection department.

```
      1. wait-time 0 seconds hearing ringback
      2. collect 7 digits after announcement 200
         (“Please enter your account number)
      3. goto step 11 if digits = none
      4. goto step 6 if digits not-in table 10
      5. route-to number 83456 with cov y if unconditionally
(collections)
      6. queue-to split 5 pri l
      7. wait-time 10 seconds hearing ringback
      8. announcement 2771
      9. wait-time 10 seconds hearing music
     10. goto step 8 if unconditionally
     11. route-to number 0 with cov n if unconditionally
     12. stop
```

Screen 4-5. Testing for Digits Not In Vector Routing Table

If no digits are collected, the call routes to the operator.

Entries in Vector Routing Tables also can be tested against the telephone number of the caller (ANI). See “ANI /II-Digits Routing and Caller Information Forwarding (CINFO)” on page 6-1 for more information.

Using Digits to Select Options

Call Prompting allows you to provide a menu of options that the caller can use to satisfy his or her information needs. The caller selects the desired option by entering the appropriate requested digit. Once the digit is entered, a conditional branch to the appropriate treatment is made. The treatment is usually provided via the *route-to number* command.

The following vector Screen 4-6 illustrates how digits are used to select options.

```
1.  wait-time 0 seconds hearing ringback
2.  collect 1 digits after announcement 3531
    (Thank you for calling Bug Out Exterminators.  If you
    wish to learn about the services we provide, please
    dial 1.  If you'd like to set up an appointment for
    one of our representatives to visit your home or
    place of business, please dial 2.)
3.  route-to number 4101 with cov y if digit = 1
4.  route-to number 4102 with cov y if digit = 2
5.  route-to number 0 with cov n if unconditionally
6.  disconnect after announcement none
```

Screen 4-6. Using Digits to Select Options

In step 2 of this vector, the user is asked to enter either 1 or 2, depending upon the service he or she desires. If one of these digits is entered, the appropriate one of the next two steps (3 through 4) routes the call to the relevant extension (that is, either 4101 or 4102). If one of the digits is not entered, the call is routed to the attendant (step 5).

Displaying Digits on the Agent's Set

You may include the CALLR-INFO button at the agents' display stations to help process calls that are serviced by the Call Prompting feature. However, if the agent has a 2-line display set, such as a 7407 or Callmaster, and the display is in normal or inspect mode, the collected digits are automatically displayed on the second line. These digits remain on this line until they are overwritten, even after the call is released by the agent. On the other hand, for other display sets, the agent must press the CALLR-INFO button to display the collected digits.

You might find it beneficial to install the CALLR-INFO button if you want to expedite calls by reducing the amount of time agents spend on the telephone. For example, the button could be set up to collect specific information (such as a customer account number) before the call is answered by the agent, thus eliminating the need for the agent to ask for this information.

The CALLR-INFO button displays information in the following format:

x=Info: 1234567890

where

- *x* is a call appearance letter (for example, *a, b, c,* etc.)
- *1234567890* represents the digits collected from the caller

The digits entered by the caller are collected by the most recent collect digits command. Any digits that were "dialed ahead" and not explicitly requested by the most recently executed *collect digits* command are not displayed.

Let's assume that digits have been collected via Call Prompting. If the agent presses the CALLR-INFO button when the call is alerting at the agent's station or when the station is active on a call appearance, the following events occur:

- Ten second timer for display interval is set.
- Status lamp (if available) associated with the button is lit.
- Display is updated. Specifically, the incoming call identification (calling party ICI) is replaced with the collected digits in the format presented earlier in this section. Only those digits collected for the last *collect digits* command are displayed.

If all the conditions to use the button (except for the collection of digits) are set, and the agent presses the button, the status lamp (if available) associated with the button flashes denial.

One or more events may occur during a successful execution after the button is pushed. These events include the following:

- Ten second timer times out.
- Incoming call arrives (at any call appearance).
- Active call changes status (for example, another caller is added to the conference).

If any of these events occur, the following takes place:

- Status lamp (if available) associated with the button is turned off.
- Display is updated (as previously described).

NOTE:
If the agent needs to display the collected digits again, the CALLR-INFO button can be depressed again to repeat the operation described in this section (provided the agent is active on the call or the call is still alerting). Also, the agent can flip between the collected digits and the ICI by alternately pressing the CALLR-INFO and NORMAL buttons.

Passing Digits to an Adjunct

Call Prompting allows for the passing of information (in the form of collected digits) to an adjunct for further processing. Digits are passed to the adjunct via the ASAI Adjunct Routing capability.

An adjunct is any processor that is connected to a switch via the ASAI link. The adjunct makes a routing decision via the *adjunct routing* command according to caller information and/or agent availability, and it returns the routing response to the switch. For example, the adjunct can indicate the call be routed to a specific ACD agent (direct-agent option).

A maximum of 16 Call Prompting digits from the last *collect digits* command can be passed to the adjunct via use of the *adjunct routing* command.

The following vector, Screen 4-7, illustrates how Call Prompting digits are passed to an adjunct:

```
1. wait-time 0 seconds hearing ringback
2. collect 10 digits after announcement 300
   (''Please enter your 10-digit account number.'')
3. adjunct routing link 50000
4. wait-time 10 seconds hearing music
5. route-to number 52000 with cov y if unconditionally
6. stop
```

Screen 4-7. Passing Digits to an Adjunct

In step 2 of this vector, the caller is asked to enter a 10-digit account number. Once the account number is entered, the adjunct receives this information via the *adjunct routing* command in step 3. This command then makes the appropriate routing decision if it is able to do so. If the command succeeds within the specified wait time, the command routes the call to the appropriate destination, and the call leaves vector processing. If the command fails, vector processing continues at the next step.

Besides the Adjunct Routing capability, collected digits also can be passed by way of ASAI to an adjunct by prompting for the digits in one vector then routing the call to a VDN that is monitored by an Event Notification (VDN) association. The collected digits (up to 16) will be sent to the adjunct in a Call Offered to Domain Event Report. See the *DEFINITY Enterprise Communications Server CallVisor ASAI Technical Reference*, 555-230-220 for detailed information.

NOTE:
Adjunct Routing is fully discussed in Chapter 10.

Creating Service Observing Vectors

As mentioned in Chapter 3, Service Observing vectors can be constructed to allow users to observe calls from a remote location or local station. When combined with Call Prompting, Service Observing vectors can route calls to:

- A Remote Access extension
- A Service Observing Feature Access Code (FAC) and extension entered by the user
- A Preprogrammed FAC and extension

Remote Access Service Observing Vector

The following vector Screen 4-8 connects a user to Remote Access. Once connected, the user can dial either a listen-only or listen/talk Service Observing FAC followed by the extension number to be observed. Although not required, Call Prompting increases security by providing passcode protection.

```
1. wait-time 0 secs hearing ringback
2. collect 5 digits after announcement 2300
   (“Please enter your 5-digit security code.”)
3. goto step 5 if digits = 12345 (security code)
4. disconnect after announcement 2000
5. route-to number 5000 with cov n if unconditionally
6. stop
```

Screen 4-8. Remote Access Service Observing Vector

User-Entered FAC and Extension

The following vector Screen 4-9 connects a user directly to the Service Observing FAC and extension based on digits collected by Call Prompting.

```
1. wait-time 0 secs hearing ringback
2. collect 5 digits after announcement 2300
   (‘‘Please enter your 5-digit security code.’’)
3. goto step 5 if digits = 12345 (security code)
4. disconnect after announcement 2000
5. wait-time 0 seconds hearing ringback
6. collect 6 digits after announcement 3245 (“Please enter the
   number 11 for listen-only observing or the number 12 for
   listen/talk observing followed by the number of the extension you
   would like to observe”)
7. route-to digits with coverage n
8. stop
```

Screen 4-9. Service Observing Vector with User-Entered FAC and Extension

Preprogrammed FAC and Extension

The following vector Screen 4-10 connects a user to a preprogrammed FAC and extension using Call Prompting to allow the observer to select the extension they would like to observe. In this example, the observer will be Service Observing a VDN.

```
1. wait-time 0 secs hearing ringback
2. collect 5 digits after announcement 2300
   (‘‘Please enter your 5-digit security code.’’)
3. goto step 5 if digits = 12345 (security code)
4. disconnect after announcement 2000
5. wait-time 0 seconds hearing ringback
6. collect 1 digits after announcement 2310 (“Enter 1 to observe
   sales, 2 to observe billing”)
7. route-to number 113001 with cov n if digit = 1 (11 = listen-only
   observe, 3001 = “Sales” VDN)
8. route-to number 113002 with cov n if digit = 2 (11 = listen-only
   observe, 3002 = “Billing” VDN)
9. goto step 6 if unconditionally
```

Screen 4-10. Service Observing Vector with Preprogrammed FAC and Extension

Dial-Ahead Digits

Dial-ahead digits provide the caller with a means of bypassing unwanted announcement prompts on the way to acquiring the information or servicing he or she desires. These digits are available for use only by subsequent *collect digits* commands. The digits are never used by other vector commands that operate on digits (for example, *route-to digits, goto...if digits,* etc.) until they are collected. These digits are not forwarded with interflowed calls. In addition, these digits are not displayed as part of the CALLR-INFO button operation until they are collected by a *collect digits* command.

The vectors shown in Screen 4-11 and Screen 4-12 illustrate a situation where a caller can enter dial-ahead digits. Note that, in this case, we are requiring the caller to have a touch-tone telephone. Typically an alternative handling sequence should be programmed in case the caller does not dial a touch tone digit before the timeout period.

```
VDN (extension=1030   name=''Coastal''   vector=30)
Vector 30:
      1. wait-time 0 seconds hearing ringback
      2. collect 1 digits after announcement 3000
         (''Thank you for calling Coastal League Baseball Hotline.
         You must have a touch-tone telephone to use this service.
         If you wish to hear the scores of yesterday's games,
         please press 1.  If you wish to hear today's schedule
         of games, please press 2.'')
      3. route-to number 1031 with cov y if digit = 1
      4. route to number 1032 with cov y if digit = 2
      5. announcement 301 (''Entry not understood.  Please
         try again.'')
      6. goto step 2 if unconditionally
VDN (extension=1031   name=''Scores''   vector=31)
Vector 31:
      1. collect 1 digits after announcement 4000
         (''If you wish to hear scores of games in both divisions,
         please press 3.  If you wish to hear scores for Northern
         Division games only, please press 4.  If you wish to hear
         scores for Southern Division games only, please press 5.'')
      2. goto step 7 if digits = 3
      3. goto step 7 if digits = 4
      4. goto step 9 if digits = 5
      5. announcement 301 (''Entry not understood.  Please
         try again.'')
      6. goto step 1 if unconditionally
      7. announcement 4002 (Northern Division scores)
      8. goto step 10 if digits = 4
      9. announcement 4003 (Southern Division scores)
     10. collect 1 digits after announcement 4004
         (''If you wish to return to the main menu,
         please press 9.  Otherwise, press 0.)
     11. route-to number 1030 with cov n if digit = 9
     12. goto step 15 if digit = 0
     13. announcement 301 (''Entry not understood.  Please
         try again.'')
     14. goto step 10 if unconditionally
     15. disconnect after announcement none
```

Screen 4-11. Dial-Ahead Digits

```
VDN (extension=1032   name=Schedule   vector=32)
Vector 32
      1. collect 1 digits after announcement 5000
         (''If you wish to hear today's schedule of games in
         both divisions, please press 6.  If you wish to hear
         today's schedule of games in the Northern Division
         only, please press 7.  If you wish to hear today's
         schedule of games in the Southern Division only,
         please press 8.'')
      2. goto step 7 if digits = 6
      3. goto step 7 if digits = 7
      4. goto step 9 if digits = 8
      5. announcement 301 (''Entry not understood.  Please
         try again.'')
      6. goto step 1 if unconditionally
      7. announcement 5002 (Northern Division schedule)
      8. goto step 10 if digits = 7
      9. announcement 5003 (Southern Division schedule)
     10. collect 1 digits after announcement 4004
         (''If you wish to return to the main menu,
         please press 9.  Otherwise, press 0.)
     11. route-to number 1030 with cov n if digit = 9
     12. goto step 15 if digits = 0
     13. announcement 301 (''Entry not understood.  Please
         try again.'')
     14. goto step 10 if unconditionally
     15. disconnect after announcement none
```

Screen 4-12. Dial-Ahead Digits

Step 2 in the first vector Screen 4-11 gives the caller two options, each of which provides different information. The caller is prompted to enter either 1 or 2, depending on what information he or she wishes to hear. Once the caller enters a digit, the digit is collected by the *collect digits* command. Thereafter, an attempt is made by the *route-to number* command to route the call to the appropriate vector (step 3 or 4). If the caller enters a digit other than *1* or *2*, the appropriate announcement is provided (step 5), and the digit entry cycle is repeated (step 6).

Let's suppose that the caller, when prompted, enters *1*. In such a case, the second vector is accessed.

In step 1 of this vector, the caller is given three options that supplement the original option provided in the first vector. The caller is prompted to enter either 3, 4, or 5, depending on what information he or she wishes to hear. If the caller enters an incorrect digit, the customary digit correction routine is implemented (steps 5 and 6). Once an appropriate digit is entered, the call is routed—this time via use of a *goto step* command (step 2, 3, or 4)—to the appropriate announcement (step 7 or step 9).

In step 10 of the second vector, the caller is once again prompted. Specifically, the caller is given the choice of returning to the main menu provided in the first vector or of terminating the phone call. If the caller selects the former option (by

entering 9), the call is routed to the first vector, and the entire process is repeated.

Note the third vector Screen 4-12 is similar in design to the second vector. The major difference is the information provided and the requested digit entries.

In our example, we have just seen that the caller has to go through at least two sets of options to get the information he or she wants. Each option set is introduced by an announcement. However, because of the “dial-ahead” digit capability, the caller can bypass the announcements if he or she so chooses. Thus, in our example, the caller could enter 1 and 5 within a matter of seconds to hear yesterday’s Southern Division scores.

The caller may enter digits while he or she is being queued for an announcement or while the announcement is playing. If digits are entered during an announcement, the announcement is disconnected. If digits are entered while a call is queued for an announcement, the call is removed from the announcement queue.

Collection of dial-ahead digits continues until one of the following occurs:

- Vector processing stops or is terminated.
- Sum of the digits collected for the current *collect digits* command plus the dial-ahead digits exceeds the switch storage limit of 24. Any additional digits are discarded until storage is freed up by a subsequent *collect digits* command.

  NOTE:
  Any asterisk (*) and pound sign (#) digits dialed ahead count toward the 24 digit limit, as do any dial-ahead digits entered after the asterisk or pound sign digit.

- The TTR required by the user to collect digits has been disconnected. This happens whenever one of the following conditions is true:
  - Successful or unsuccessful *route-to number* step is encountered during vector processing, except where the number routed to is a VDN extension.
  - Successful or unsuccessful *route-to digits* step is encountered during vector processing, except where the number routed to is a VDN extension.
  - Successful or unsuccessful *adjunct routing* step is encountered during vector processing.
  - Successful or unsuccessful *converse-on* step is encountered during vector processing.

— Call Prompting timeout occurs, during which time the caller has not dialed any additional digits, asterisks (*) or pound signs (#).

— Vector processing stops or is terminated.

— A successful or unsuccessful collect ced/cdpd step is encountered.

NOTE:
When the TTR is disconnected due to a *route-to number, route-to digits, converse-on*, *adjunct routing,* or *collect ced/cdpd* step, all dial-ahead digits will be discarded. This means that following a failed *route-to, converse* or *adjunct routing* step, a subsequent *collect digits* step always requires the user to enter digits.

The caller who enters dial-ahead digits no doubt knows which digits to enter ahead of time due to his or her familiarity with the service provided. Once the caller masters the digit sequence relevant to a particular service, the dial-ahead digit capability saves time and also eliminates much of the redundancy associated with automatic telephone servicing.

ASAI-Requested Digit Collection

The ASAI-requested digit collection feature gives an adjunct the ability to request that a DTMF tone detector (TN744 or TN 2182) be connected for the purpose of detecting user-entered digits. The digits collected as a result of this feature are passed to ASAI monitoring and/or controlling adjuncts for action. The switch handles these digits like dial-ahead digits. This feature allows the caller to request Sequence Dialing after the call has been routed to the final destination and has resulted in an unanswered call (busy, no answer, etc).

Note that these digits are not necessarily collected while the call is in vector processing. They are sent to an ASAI adjunct, and/or they may be used by Call Prompting features.

ASAI Adjunct Routing and Call Prompting features must be enabled on the switch for this feature to work.

ASAI-Provided Dial-Ahead Digits

The ASAI-provided digits feature allows an adjunct to include digits in a *Route Select* capability. These digits are treated as dial-ahead digits for the call. Dial-ahead digits are stored in a dial-ahead digit buffer and can be collected (one at a time or in groups) using the *collect digits* command(s). Although the adjunct may send more than 24 digits in a Route Select, only the first 24 (or 24-x, where x is the number of digits collected by vector processing prior to executing the *adjunct routing* vector command) digits are retained as dial-ahead digits. An application can use this capability to specify the digits that the switch should pass to the VRU as part of the *converse-on* vector step.

NOTE:
The maximum number of dial-ahead digits that can be stored in the buffer is dependent on the number of digits already collected for the call by a previous “collect digits” vector command. If ’x’ digits were collected by vector processing prior to executing an “adjunct routing” vector command, the ’x’ digits collected reduces the maximum number of digits that can be stored as dial-ahead digits as a result of a Route Select. The rest are discarded.

Advanced Vector Routing

5

Introduction

Advanced Vector Routing adds significantly to the conditional routing capabilities of Basic Call Vectoring. Specifically, it adds the following conditions for routing calls.

- Expected Wait Time (expected-wait)
- Rolling Average Speed of Answer (rolling-asa)
- VDN Calls (counted-calls)

Command Set

Table 5-1 illustrates the commands used in Advanced Vector Routing.

Table 5-1. Advanced Vector Routing Command Set

Command Category	Action Taken	Command
ROUTING	Queue the call to a backup ACD split.	*check split*
BRANCHING/ PROGRAMMING	Go to a vector step.	*goto step*
	Go to another vector.	*goto vector*

Expected Wait Time (EWT)

EWT Routing allows you to make routing decisions based on the time that a caller can expect to wait in queue. This wait time can be predicted for a split or for a call. When predicted for a split, the wait time indicates the amount of time the caller can expect to wait if the call is queued to the specified split. When predicted for a call, the wait time indicates the time remaining that the caller can expect to wait in queue until the call is serviced from the queue. The expected wait time can also be passed to a VRU so that a caller can be notified of his or her expected time in queue. The *expected-wait* conditional can be used with either the *goto* or *check* commands.

Call vectoring offers several conditionals that can be used to estimate the time a caller will be delayed waiting in queue, for example, EWT, rolling ASA and Oldest Call Waiting (OCW). EWT is the most accurate of these conditionals. It takes into account more real-time and historical information than the other predictors. For example, priority level, position in queue, number of working agents, etc.

EWT is very responsive to changing call center conditions. For example, it adjusts instantly to any staffing changes in the split; if an agent moves into or out of auxiliary work mode, the wait time predictions adjust immediately.

EWT does not include the time in a call vector before the call enters a queue. It also does not include the time the call rings at a voice terminal after it is removed from the queue.

See "When to Use Wait Time Predictions" later in this chapter for a description of when the predictions are most accurate and the circumstances that will limit their accuracy.

EWT for a Split

The EWT for a split is the time that a new call would be expected to remain in queue if it were queued to the split at the specified priority level. It is generally used to determine if a call should be queued to the split.

For example, the vector in Screen 5-1 uses EWT for a split to determine if a call should be queued to that split.

```
1. goto step 3 if expected-wait for split 1 pri l < 600
2. busy
3. queue-to split 1 pri l
4. announcement 3001
5. wait-time 998 secs hearing music
```

Screen 5-1. EWT for a Split

If there are agents available, EWT is zero.

EWT is infinite if:

- There are no logged-in agents
- All logged-in agents are in AUX work mode
- The split queue is full
- There is no split queue and all agents are busy
- The split queue is locked

EWT for a Call

EWT for a call is the remaining time a caller can expect to wait before his or her call is serviced from queue. If the call is queued to multiple splits, the remaining queue time for each of the splits is calculated, and the shortest of these is taken as the call's EWT.

For a call to have an expected wait time it must be queued to at least one split. If it is not queued, or if it is queued to splits that are not staffed, the EWT value is infinite.

The example in Screen 5-2 uses EWT for a call to determine the treatment the call will receive.

```
1. queue-to split 1 pri m
2. check split 2 pri m if expected-wait < 30
3. goto step 5 if expected-wait for call < 9999
4. busy
5. announcement 3001
6. wait-time 998 secs hearing music
```

Screen 5-2. EWT for a Call

Passing EWT to a VRU

As stated, the Expected Wait Time for a call can be passed to a VRU so that a caller can be notified of his or her expected time in queue. EWT is passed to the VRU with the *converse-on* command as "wait" data. The value outpulsed to the VRU is the expected wait time of the call in seconds. The VRU can then convert the seconds to a spoken message probably rounding up to minutes or converting to minutes and seconds. The expected wait is calculated after the VRU port answers the call, so queuing to a converse split does not adversely impact the EWT value passed to the VRU.

No zero padding is added to the wait time passed to the VRU. If the EWT for the call is 128 seconds, the digits 1, 2, and 8 are outpulsed. If the EWT is 5 seconds, the digit 5 is outpulsed.

The wait time passed to the VRU is the most accurate prediction possible. On the average 50% of the time the actual wait time will be shorter and 50% of the time it will be longer. It is recommended that VRU applications make an upwards adjustment of the prediction so that the majority of callers receive a predicted wait time that is equal to or greater than their actual wait time.

The VRU can also announce expected wait time to a caller periodically throughout the time that a call is in queue. In this way, the caller can observe his or her progress up the queue. However, this approach should be used with caution. Circumstances such as a reduction in the number of agents or a sudden influx of higher priority calls could cause the caller's expected wait time to increase from one announcement to the next.

If the call is not queued or if it is queued only to splits that are unstaffed or splits where all agents are in AUX work mode, the end-of-string character "#" is the only data item outpulsed.

The EWT Algorithm

EWT is calculated using an algorithm that is based on the number of calls in a queue at a particular priority level and the rate of service of calls from the queue at that priority level. It adjusts for many other factors such as multiple split queuing, call handling times, and the impact of direct agent calls on the wait time of other calls to the split. The algorithm adjusts EWT immediately for changes in staffing, such as agents logging in or taking breaks in AUX work mode.

Since changes occur constantly in a call center, and since EWT cannot predict the future, the accuracy of the EWT predictions will be in proportion to the rate at which calls are serviced from the queue and the level of stability achieved in the call center between the time that the prediction is made and the time that the call is serviced from queue.

When to Use Wait Time Predictions

This section contains a number of situations that can have an adverse impact on the accuracy of wait time predictions. These factors have an adverse impact on all predictors, not just on EWT. The EWT algorithm still should be more accurate than other predictors, even when these situations are present.

Wait time predictions are best suited for medium or high volume call scenarios. In general, the potential accuracy of a wait time predictor increases as the rate of removal from queue increases. It is recommended that EWT be used when the rate of removal from queue at a given split priority level is at least one call every 30 seconds.

Predictions can be made for a split with multiple priority levels in use as long as the majority of calls are delivered to the lower priority levels. If the majority of calls

are queued at the higher priority levels, any predictions made for the lower priority levels may not be accurate.

The following list describes circumstances that will limit the accuracy of the wait time predictions.

- Immediately after a system restart or when a new split is administered.

  The EWT algorithm uses a combination of historical and real-time information to make predictions. When no historical information exists, such as when a new split is added or a reset system 3 or 4 is completed, there is the potential for inaccuracies.

  To prevent inaccurate predictions when there is no historical information, administer the Expected Call Handling Time field on the Hunt Group form. The value in this field is then used in place of the missing historical data.If the value of this field does not accurately reflect the call handling times of the split, EWT predictions may be inaccurate until some call history is generated. The algorithm normally requires about 30 queued calls to be answered from a split priority level before it reaches its maximum accuracy.

  You can change the value in the “Expected Call Handling Time” field by executing a change hunt group command. Changing the value will not disrupt EWT predictions by overwriting EWT history. The value is stored and used the next time a reset system 3 or 4 is executed.

- Low call volume applications.

  Split priority levels where the rate of removal from queue is very low can only be predicted with limited accuracy.

- Sites with frequent staffing changes.

  Although EWT immediately adjusts for all types of staffing changes, since predictions may have already been made for calls waiting in queue, those past predictions will have been based on staffing information which is now out of date. Therefore, scenarios where large staffing changes are continually happening can only be predicted with limited accuracy.

- Staffed agents who rarely answer calls to a split.

  The EWT algorithm takes account of agents in multiple splits in its calculation. However, suppose there are many agents who are assigned to a split but spend most of their time answering calls in their other splits. If a large number of these agents are moved to or from the split, then EWT for this split may be temporarily inaccurate until it adjusts to those changes.

- Applications with widely varying call handling times.

  If the majority of calls to a split are handled within a narrow range of times the accuracy of any predictor will be much greater than that for a split where call handling times are widely different.

Examples

Example 1 — EWT Routing and Passing Wait to a VRU

The following vector, Screen 5-3, illustrates routing based on the wait time of a split, as well as passing wait data to the VRU. Wait time is only given to the caller if the caller is expected to wait a total of more than 60 seconds in queue. Callers who would wait more than 10 minutes are told to call back later.

```
1. goto step 3 if expected-wait for split 32 pri l < 600
2. disconnect after announcement 13976
3. queue-to split 32 pri l
4. wait-time 20 secs hearing ringback
5. goto step 7 if expected-wait for call < 40
6. converse-on split 80 pri l passing wait and none
7. announcement 11000
8. wait-time 60 secs hearing music
9. goto step 7 if unconditionally
```

Screen 5-3. EWT Routing and Passing VRU Wait

Calls with more than 10 minutes to wait fail step 1 and are disconnected after an announcement asking them to call back later. If the expected wait time is less than 10 minutes step one routes the call to step 3 where it is queued to split 32 and waits 20 seconds hearing ringback. After 20 seconds if the expected wait time for the call is less than 40 seconds, step 5 routes the call to an announcement followed by a wait with music. If the expected wait time for the call is equal to or greater than 40 seconds, step 6 informs the caller of the amount of time he or she can expect to wait before the call is answered.

Example 2 — Notifying Callers of Wait Time Without a VRU

You can still use EWT to notify calls of their expected wait time even without a VRU. This can be done using the DEFINITY System recorded announcements and by associating each recorded announcement with a time band as illustrated in Screen 5-4.

```
VECTOR 101
       1. queue-to split 3 pri h
       2. goto step 4 if expected-wait for call <= 600
       3. busy
       4. wait-time 12 seconds hearing ringback
       5. announcement 3001 (“Thank you for calling ABC Inc. All agents
          are busy, please wait and we will get to your call as soon as
          possible”)
       6. goto vector 202 if unconditionally

VECTOR 202
       1. goto step 13 if expected-wait for call > 280
       2. goto step 11 if expected-wait for call > 165
       3. goto step 9 if expected-wait for call > 110
       4. goto step 7 if expected-wait for call > 55
       5. announcement 3501 (“Thank you for waiting.
          Your call should be answered within the next minute”)
       6. goto step 14 if unconditionally
       7. announcement 3502 (“Thank you for waiting.
          Your call should be answered within approximately one to
          two minutes”)
       8. goto step 14 if unconditionally
       9. announcement 3503 (“Thank you for waiting.
          Your call should be answered within approximately two to
          three minutes”)
      10. goto step 14 if unconditionally
      11. announcement 3504 (“Thank you for waiting.
          Your call should be answered within approximately three to
          five minutes”)
      12. goto step 14 if unconditionally
      13. announcement 3505 (“We apologize for the delay. Due to heavy
          call volume, you may have to wait longer than five minutes
          to speak to a representative. If possible, we suggest that you
          call between the hours of 8am and 10am for the fastest service”)
      14. wait-time 120 secs hearing music
      15. goto step 1 if unconditionally
```

Screen 5-4. Notifying Callers of Wait-Time Without a VRU

In step 1 the call is queued to split 3 at high priority. If the call fails to get a queue slot in split 3, if split 3 has no working agents, or if the wait time in split 3 at high priority exceeds 10 minutes, step 2 fails and the caller receives busy tone. If step 2 succeeds, the caller hears ringback and an announcement and is then sent to vector 202. Steps 1 through 4 of vector 202 determine which of five time bands the caller’s remaining queuing time is estimated to be within. One of five recorded announcements is then played to the caller to inform him or her of the expected wait time in queue.

Notice that the EWT thresholds are set lower than the times quoted in the recorded announcements. Callers may become upset if their actual wait time exceeds the time stated in the announcement. Therefore, you may want to program your vectors such that few callers ever experience wait times that exceed the wait time of the announcement.

Notice also that vector 202 can be used for any application requiring that the caller be notified of their remaining time in queue.

Example 3 — Using EWT to Route to the Best Split

With EWT, you may wish to change your normal queuing strategy of queuing calls to multiple splits in order to insure the call is answered in the shortest possible time. This strategy uses additional system resources and can make it more difficult to read and analyze split reports.

Instead, you may wish to use EWT to determine up-front which split is best for each call and avoid multiple split queuing.

In Screen 5-5, there are two splits, a main split (1) and a backup split (2). Either split can service a particular type of call. It is preferable that an agent from the main split service the call. However, a 30-second maximum wait time is also desirable. The strategy in this vector is to use the backup split only if the backup split can answer the call within 30 seconds and the main split cannot.

```
 1. goto step 5 if expected-wait for split 1 pri m <= 30
 2. goto step 5 if expected-wait for split 2 pri m > 30
 3. check split 2 pri m if unconditionally
 4. goto step 6 if unconditionally
 5. queue-to split 1 pri m
 6. wait-time 12 secs hearing ringback
 7. announcement 3501
 8. converse-on split 18 pri m passing wait and none
 9. wait-time 120 secs hearing music
10. goto step 8 if unconditionally
```

Screen 5-5. EWT Routing—Routing to the Best Split

Step 1 branches to step 5 to queue to the main split if the main split can answer the call within 30 seconds. If the main split cannot answer the call within 30 seconds, step 2 checks to see if the backup split can answer the call within 30 seconds. If it cannot, the call branches to step 5 and is queued to the main split. If it can, the call is queued to the backup split in step 3. At this point, the call is queued either to the main or the backup split but not to both.

Steps 6 through 10 provide audible feedback to the caller while the call is in queue. Note that in step 8, which is executed every two minutes, a VRU is used to provide the caller with his or her remaining wait time in queue.

Factors that Affect the Value of EWT

Factors that Cause EWT for a Split Priority Level to Increase

Most common:

- Number of calls in queue increases
- Agents logout
- Agents go on break (AUX work mode)
- Agents are moved to another split
- Agents with multiple splits answer an increasing number of calls in other splits

Other possibilities:

- Average talk time increases
- Number of calls at higher priority increases
- Number of DAC calls increases
- Number of RONA calls increases
- Number of abandoned calls decreases
- Number of calls queued in this split but answered in another decreases

Factors that Cause EWT for a Split Priority Level to Decrease

Most common:

- Number of calls in queue decreases
- Agents login (and start answering calls)
- Agents return from break (leave AUX work mode)
- Agents are moved from another split
- Agents with multiple splits answer fewer calls in other splits

Other possibilities:

- Average talk time decreases
- Number of calls at higher priority decreases
- Number of DAC calls decreases
- Number of RONA calls decreases
- Number of abandoned calls increases
- Number of calls queued in this split but answered in another increases

Simple Troubleshooting for EWT

To verify that your EWT is operating as intended, use the **list trace ewt** command to observe processing events of alll calls. Refer to "Tracing Expected Wait Time" on page E-33 for more information.

Rolling Average Speed of Answer (ASA)

Rolling ASA Routing allows you to make routing decisions based on the current average time that it takes for a call to be answered in a split or VDN. In this way, a vector can route a call to the VDN or split where it is likely to be answered most quickly.

The Average Speed of Answer used for vector routing is called "rolling" ASA to differentiate it from the "interval" ASA that is recorded in BCMS and CMS reports. Rolling ASA is a running calculation that does not take into account the 15-minute, half-hour, or hour BCMS/CMS reporting intervals. It does not reflect interval boundaries. On the other hand, the "interval" ASA used for BCMS/CMS reporting is calculated on reporting interval boundaries and clears to zero at the start of each reporting interval.

The Rolling Average Speed of Answer for a split or VDN is calculated based on the speed of answer for all calls recorded since system start-up. When rolling ASA is calculated, each call is given a weighted value that is greater than the call that preceded it. In this way, the most recent calls contribute the most to the average. Approximately 95% of the value of rolling ASA is obtained from the last ten calls.

The rolling ASA for a split or VDN is recalculated every time a call is answered so that it always reflects the most recently available data. Calls that are not answered, for example calls that receive a forced busy, are not considered for the rolling ASA calculation.

The rolling ASA is calculated for an entire split or VDN. The calculation does not consider the priority levels of answered calls.

The following sections explain what is included in the rolling ASA calculation for a split or VDN.

Rolling ASA Split Calculation

The rolling ASA for a split is the average time it takes for a call to be answered from the time the call attempts termination to the split until it is answered in that split. Rolling ASA includes the time the call is waiting in queue and the time it is ringing at a voice terminal.

If the call is answered in another split or the call is abandoned by the caller before it is answered, rolling ASA is not recorded for the call. If a call flows into a split from another split, the time queued and ring time for the previous split are not included. If a call is queued in multiple splits, only the rolling ASA for the split in which the call is answered is impacted.

Rolling ASA VDN Calculation

The rolling ASA for a VDN is the average time it takes for a call to be answered from the time it starts processing within the specified VDN until it is answered. It includes any time spent in vector processing including time spent in announcements administered as vector steps. If the call is answered by an agent, it includes the time the call is waiting in queue and the time it is ringing at the agent's voice terminal.

The rolling ASA for a VDN only includes data from calls answered in that VDN. If a call flows between VDNs, only the time spent within the answering VDN is used in the calculation. For example, if a call is placed to VDN1 and after ten seconds routes to VDN2 and is then answered in VDN2 after five seconds, the ASA for the call is recorded in VDN2 as five seconds. Nothing is recorded for VDN1 since the call was not answered there.

The VDN for a vector step can be specified in three ways: a VDN number, the value "latest," or the value "active." The "latest" VDN is the VDN that is currently processing the call. The value is not affected by VDN override. The "active" VDN is the VDN of record. That is, it is the called VDN as modified by override rules. For example, if a call routes from a VDN with override set to *yes* then the new VDN is the active VDN. If a call routes from a VDN with override set to *no* then the previous VDN is the active VDN.

Rolling ASA Considerations

Because of its greater accuracy and greater flexibility, EWT is recommended over rolling ASA as a predictor of split/skill waiting time. However, rolling ASA is provided for those who may have a special requirement or wish to use the more traditional ASA measurement.

Normally, rolling ASA conditionals should not be used to prevent calls queuing to the main split/skill or being answered in the principal VDN. Rather, rolling ASA should be used to see whether vector processing should attempt to queue the call to additional splits/skills if the main split/skill does not currently meet the targeted threshold. If no calls are being answered in the main split/skill or VDN, the value of rolling ASA will not change. This could result in all future calls being locked out of the main split/skill or VDN unless there are other call vectors in the system directing calls to them.

If you wish to implement a call flow that decides whether or not to queue a call to a main split/skill, use the EWT feature.

Example

The following example combines VDN and split ASA routing.

```
1. queue-to split 10 pri h
2. goto step 6 if rolling-asa for split 10 <= 30
3. check split 11 pri h if rolling-asa <= 30
4. check split 12 pri h if rolling-asa <= 30
5. check split 13 pri h if rolling-asa <= 30
6. announcement 10000
7. wait-time 40 secs hearing music
8. goto step 3 if unconditionally
```

Screen 5-6. Rolling ASA Routing

Step 1 queues the call to the main split. If the main split is currently answering calls within the target time of 30 seconds, step 2 bypasses all of the backup splits and goes directly to the announcement in step 6. The assumption is that the call will be handled by split 10 within the time constraints. However, if the call is not answered by the time vector processing reaches step 8, the backup splits are checked at that time.

If the rolling ASA for the main split is greater than 30 seconds, steps 3, 4, and 5 check backup splits. The call is queued to any of these splits that have a rolling ASA of 30 seconds or less. If the call still is not answered by the time vector processing reaches step 8, then the backup splits are checked again.

VDN Calls

VDN Calls routing allows you to make routing decisions based on the number of incoming trunk calls that are currently active in a VDN. With the VDN Calls conditional, a vector can be used to limit the number of simultaneous calls made to a particular VDN. For example, if a service agency is contracted to handle 100 simultaneous calls for a client, calls in excess of that number can be routed to a *busy* step.

When Advanced Vector Routing is enabled, a count of active incoming trunk calls is kept for each VDN. The VDN counter is incremented each time an incoming call is placed to the VDN. It is decremented each time an incoming call is released. A call is considered active in a VDN from the time the call routes to the VDN until all parties on the call have been dropped and the call is released.

NOTE:
The call is counted for the originally called VDN only. When a call is routed to another VDN, the call counter for the subsequent VDN is not incremented. And, the call counter for the original VDN is not decremented.

As with other Advanced Vector Routing conditionals, the VDN for a *goto* step can be specified in three ways: a VDN number, the value “latest,” or the value “active.”

The following section describes which calls are included in the VDN Calls counts and which are not.

Counted Calls

The VDN call count includes:

- Incoming trunk calls that route directly to the VDN.
- Incoming trunk night service calls where the VDN is the night service destination.
- Calls that cover or forward to the VDN if it is the first VDN routed to and the call is an incoming trunk call.
- Already counted calls that are conferenced with counted or not counted calls from the same VDN.

The VDN call count does not include:

- Internal calls to the VDN.
- Calls that are transferred to the VDN.
- Calls redirected to their VDN return destination.
- Conferenced calls previously counted on different VDNs.

Example

The following example shows how the *counted-calls* conditional can be used to route calls.

```
1. goto step 3 if counted-calls to vdn 1234 <= 100
2. busy
3. queue-to split 60 pri l
4. wait-time 20 seconds hearing ringback
5. announcement 27000
6. wait-time 60 seconds hearing music
7. goto step 5 unconditionally
```

Screen 5-7. VDN Calls Routing

If more than 100 calls are active in VDN 1234, the caller will hear busy tone and vector processing is terminated. If 100 or fewer calls are active, the call queues to split 60.

ANI /II-Digits Routing and Caller Information Forwarding (CINFO)

6

Introduction

ANI and II-digits allow you to make vector routing decisions based on the caller identity and the type of the originating line. Caller Information Forwarding (CINFO) allows you to collect caller entered digits (ced) and customer database provided digits (cdpd) for a call from the network.

ANI and II-digits when provided with an incoming call to a VDN are sent to CentreVu® CMS when vector processing starts. ANI, II, and CINFO digits are forwarded with interflowed calls. In addition, ANI and II-digits are passed over ASAI in event reports.

Command Sets

ANI and II-digits are both used for conditional branching with the *goto* step. Table 6-1 illustrates the commands used in ANI/II-Digits Routing.

Table 6-1. ANI/II-Digits Routing Command Set

Command Category	Action Taken	Command
BRANCHING/ PROGRAMMING	Go to a vector step.	*goto step*
	Go to another vector.	*goto vector*
INFORMATION COLLECTION	Pass ANI to a Voice Response Unit (VRU).	*converse-on*

Continued on next page

Table 6-2 illustrates the commands that can use CINFO digits.

Table 6-2. CINFO Command Set

Command Category	Action Taken	Command
INFORMATION COLLECTION	Collect ced and cdpd from a network ISDN SETUP message.	*collect digits*
	Pass ced and cdpd digits to a Voice Response Unit (VRU).	*converse-on*
ROUTING	Route the call to a number programmed in the vector based on ced/cdpd digits.	*route-to number*
	Route the call to digits supplied by the network.	*route-to digits*
	Request routing information from an ASAI adjunct based on ced or cdpd.	*adjunct routing*
BRANCHING/ PROGRAMMING	Go to a vector step based on ced/cdpd digits.	*goto step*
	Go to another vector based on ced/cdpd digits.	*goto vector*

ANI Routing

ANI routing allows you to make routing decisions based on incoming or internal caller identity. In this way, calls from a particular customer can receive unique routing, local calls can be routed differently from long distance calls, or calls from different geographical areas can receive different routing. See "ANI Routing Example" later in this section for more information. ANI also can be compared against entries in a Vector Routing Table. See "Vector Routing Tables with ANI" later in this section for more information.

ANI (Automatic Number Identification) is based on the Calling Party Number (CPN). It is not always identical to the Billing Number. For example, if the call is placed by a user from a Private Branch Exchange (PBX), the CPN can be either the PBX billing number or the Station Identification Number.

The ANI routing digit string can contain up to 16 digits. This supports international applications. However, ANI information in North America only contains 10 digits.

The following calls will have ANI values associated with them:

- Incoming ISDN-PRI calls that send ANI
- Incoming R2MFC Signaling calls that send ANI
- DCS calls
- Internal calls

If ANI is not provided by the network for a call, then it will not be available for vector processing on that call.

When an EAS agent makes a call to a VDN, the agent's login ID is used as the ANI, not the number of the physical terminal.

When a call is transferred internally to a VDN the following is true:

- If the transfer is completed before the call reaches the ANI conditional, the ANI value of the originator of the call is used.
- If the transfer is completed after the call reaches the ANI conditional, the ANI value of the terminal executing the transfer is used.

To ensure that the originator's ANI is preserved during a transfer, add a filler step (such as wait with silence) to the beginning of the vector. In this way a transfer can complete before the ANI conditional is encountered.

The ANI value specified for a goto step can include the "+" and/or "?" wildcard. The "+" represents a group of zero or more digits and can only be used as the first or last character of the string. The "?" represents a single digit. Any number of them can be used at any position in the digit string.

ANI Routing Example

The example in Screen 6-1 demonstrates several applications of ANI Routing.

```
 1. wait-time 4 secs hearing silence
 2. goto step 13 if ani = none
 3. goto step 12 if ani = 3035367326
 4. goto vector 74920 if ani <= 9999999
 5. goto vector 43902 if ani = 212+
 6. goto vector 43902 if ani = 202+
 7. wait-time 0 seconds hearing ringback
 8. queue-to split 16 pri m
 9. wait-time 120 seconds hearing 32567 then continue
10. announcement 32456
11. goto step 9 if unconditionally
12. route-to number 34527 with cov y if unconditionally
13. route-to number 0 with cov n if unconditionally
14. busy
```

Screen 6-1. ANI Routing Example

In step 2 calls that do not have ANI associated with them are routed to an operator. Step 3 routes calls from a specific phone to a specified extension. Step 4 routes local calls (those with 7 or fewer digits) to a different vector. Steps 5 and 6 route calls from area codes 212 and 202 to a different vector. Calls that are not rerouted by the previous steps are then queued.

Vector Routing Tables with ANI

You can also test ANI against entries in a Vector Routing Table.

Vector Routing Tables contain a list of numbers that can be used to test a *goto...if ani* command. ANI can be tested to see if it is either *in* or *not-in* the specified table. Entries in the tables can also include the "+" and/or "?" wildcard.

Screen 6-2 gives an example of a Vector Routing Table with ANI values. The table contains most of the area codes for the state of California.

```
                    VECTOR ROUTING TABLE
Number: 6             Name: California          Sort? n


       1: 714+                              17: ________
       2: 805+                              18: ________
       3: 619+                              19: ________
       4: 707+                              20: ________
       5: 209+                              21: ________
       6: 310+                              22: ________
       7: 213+                              23: ________
       8: 408+                              24: ________
       9: 510+                              25: ________
      10: 818+                              26: ________
      11: 909+                              27: ________
      12: 916+                              28: ________
      13: 415+                              29: ________
      14: ________                          30: ________
      15: ________                          31: ________
      16: ________                          32: ________
```

Screen 6-2. Vector Routing Table for ANI Routing

The following vector, Screen 6-3, could be used to route these California area code calls to a separate vector.

```
1. announcement 45673
2. goto step 9 if ani = none
3. goto vector 8 if ani in table 6
4. queue-to split 5 pri l
5. wait-time 10 seconds hearing ringback
6. announcement 2771
7. wait-time 10 seconds hearing music
8. goto step 6 if unconditionally
9. route-to number 0 with cov y if unconditionally
```

Screen 6-3. Testing for ANI In Vector Routing Table

In this example, if no ANI is available for the call, it is routed to an operator. If the first three numbers match an area code from table 6, the call is routed to vector 8. All other calls are queued.

II-Digits Routing

II-digits routing allows you to make routing decision based on the type of the originating line. In this way, calls from pay phones, cellular phones, or motel phones, for example, can receive unique routing. See "II-Digits Routing Example" later in this section for more information.

II-digits (Information Indicator Digits) is a 2-digit string provided for an incoming call by ISDN PRI. II-digits delivery is a generally available ISDN PRI AT&T Network service. This service is bundled with ANI delivery and tariffed under the MEGACOM 800® and MultiQuest 800® INFO-2 feature to provide information about the call's origination. II-digits indicate the type of originating line.

II-digits routing can be used for example to:

- Help detect fraudulent orders for catalog sales, travel reservations, money transfers, traveler's checks, and so on.
- Assign priority or special treatment to calls placed from pay phones, cellular phones, or other types of lines. For example, special priority could be given by an automobile emergency road service to calls placed from pay phones.
- Detect calls placed from pay phones when it is the intention of the caller to avoid being tracked by collection agencies or dispatching services.
- Convey the type of originating line on the agent display by routing different type calls to different VDNs.

The II-digits routing string can only contain 2 characters. The string can contain either the "+" or "?" wildcard. Leading zeros are significant. The II-digits value "02" associated with a call will not match the digit string "2" in a vector step.

As with ANI routing and collected digit routing, II-routing digits can be compared against entries in a Vector Routing Table.

The following calls will have II-digits values associated with them.

- Incoming ISDN PRI calls that include II-digits.
- Incoming ISDN PRI Tie Trunk DCS or non-DCS calls that include II-digits.

  Note that since tandeming of II-digits is only supported if the trunk facilities used are ISDN PRI, traditional DCS will not support II-digits transport but DCS Plus (DCS over PRI) will.

When a call is returned to vector processing as a result of the VDN Return Destination feature, the ii-digits are preserved.

When a call is transferred internally to a VDN the following is true:

- If the transfer is completed before the call reaches the II-digits conditional, the ii-digits value of the originator of the call is used.
- If the transfer is completed after the call reaches the II-digits conditional, the II-digits value of the terminal executing the transfer is used. Under normal circumstances, there will be no II-digits for a terminal executing a transfer.

To ensure that the originator's II-digits is preserved, add a filler step (such as wait with silence) to the beginning of the vector. In this way a transfer can complete before the II-digits conditional is encountered.

Table 6-3 is a paraphrased summary of currently available ii-digits. A complete and more descriptive list of II-digits is published quarterly in Section 1 of the "Local Exchange Routing Guide" published by Bellcore.

Table 6-3. II-digits Summary

Code	Use
00	Identified line - no special treatment
01	Multiparty - ANI cannot be provided
02	ANI failure
06	Hotel/Motel - DN not accompanied by automatic room ID
07	Special operator handling required
20	AIOD - Listed DN of PBX sent
23	Coin or Non-Coin - line status unknown
24	800 Service

Continued on next page

Table 6-3. II-digits Summary — *Continued*

Code	Use
27	Coin Call
29	Prison/Inmate Service
30-32	Intercept
34	Telco Operator Handled Call
40-49	Locally determined by carrier
52	OutWATS
60	Telecommunication Relay Service (TRS) - Station Paid
61	Type 1 Cellular
62	Type 2 Cellular
63	Romer Cellular
66	TRS - From Hotel/Motel
67	TRS - From restricted line
70	Private paystation
93	Private Virtual Network call

II-Digits Routing Example

The example in Screen 6-4 demonstrates branching calls with different II-digits to different VDNs. The VDN override is set to “yes” on the called VDN. In this way, the VDN name or VDN of Origin Announcement can be used to convey to the agent the type of II-digits associated with the call.

```
 1. goto step 9 if ii-digits = none
 2. goto step 10 if ii-digits = 00
 3. goto step 11 if ii-digits = 01
 4. goto step 12 if ii-digits = 06
 5. goto step 13 if ii-digits = 07
 6. goto step 13 if ii-digits = 29
 7. goto step 14 if ii-digits = 27
 8. goto step 15 if ii-digits = 61
 9. route-to number 1232 with cov n if unconditionally
10. route-to number 1246 with cov n if unconditionally
11. route-to number 1267 with cov n if unconditionally
12. route-to number 1298 with cov n if unconditionally
13. route-to number 1255 with cov n if unconditionally
14. route-to number 1298 with cov n if unconditionally
15. route-to number 1254 with cov n if unconditionally
```

Screen 6-4. II-Digits Routing Example

In this vector, step 1 routes calls with no associated II-digits to extension 1232. Steps 2 through 8 route calls with different II-digits to different extensions.

Caller Information Forwarding

Caller Information Forwarding (CINFO) allows you to use *collect digits* steps to retrieve caller entered digits (ced) and customer database provided digits (cdpd) supplied by the network in an incoming call's ISDN PRI SETUP message. These network provided digits are available with AT&T Network Intelligent Call Processing (ICP) service. ISDN-PRI is required.

For example, a caller could dial a number that resulted in ICP routing at the network switch. The network switch could request information from the caller (ced) and/or it could request information from the call center customer host database (cdpd). These digits are sent in the call ISDN message to the DEFINITY System and are then available for a *collect digits* vector steps.

Up to 30 ced and/or up to 30 cdpd digits can be stored for a call. These digits also are forwarded with a tandemed or interflowed call.

Detailed Operation

When an ISDN call is received from either the AT&T network or a tandemed PRI call, the system stores the Codeset 6 User Entered Code (UEC) Information Element when it contains the caller entered digits and/or customer database provided digits. If more than one ced UEC IE is received only the first one is stored or tandemed with the call. If more than one cdpd UEC IE is received only the first one is stored or tandemed with the call.

When a *collect ced digits* or *collect cdpd digits* step is processed, the system retrieves the ced or cdpd digits and places them in the collected digits buffer. Any digits that were in the collected digits buffer, such as dial-ahead digits, are erased. If a TTR was connected to the call from a previous *collect digits* step, the TTR is disconnected.

If the ced or cdpd digits contain invalid digits (not 0-9, *, #) the system does not store the UEC IE. When the *collect digits* step is reached, the collected digits buffer is still cleared and if a TTR is attached it is still disconnected. A vector event is generated indicating that no digits were collected.

If no ced or cdpd digits were received from the network, when the *collect ced digits* or *collect cdpd digits* step is reached, the step is skipped. However, the collected digits buffer is still cleared and if a TTR is attached it is still disconnected.

If a * is included in the collected digits, it is treated as a delete character. Only the digits to the right of the * are collected. If a # is included in the collected digits it is treated as a terminating character. Only the # and the digits to the left of the # are collected. If a single # is sent, it is placed in the collected digits buffer.

The number of ced or cdpd digits to collect cannot be specified in the *collect digits* step. If there are 16 or fewer digits, all the digits are collected. If there are more than 16 digits, the first 16 digits are collected and a vector event is generated.

The CINFO ced and cdpd digits can be used with any vector step that uses the digits in the collected digits buffer. These steps are:

- *adjunct routing* (digits passed in an event report as collected digits)
- *converse-on...passing digits*
- *goto...if digits...*
- *goto...if digits in table...*
- *route-to digits*
- *route-to number ... if digit...*

ced or cdpd digits can be displayed using the callr-info button in the same way as other collected digits.

When a call is transferred internally to a VDN the following is true:

- If the transfer is completed before the call reaches the CINFO conditional, the CINFO value of the originator of the call is used.
- If the transfer is completed after the call reaches the CINFO conditional, the CINFO value of the terminal executing the transfer is used.

To ensure that the originator's CINFO is preserved during a transfer, add a filler step (such as wait with silence) to the beginning of the vector. In this way a transfer can complete before the CINFO conditional is encountered.

To retrieve both the ced and cdpd for a call, you must use two *collect digits* steps. Because the *collect digits* command for ced or cdpd clears the collected digits buffer, the ced or cdpd that is collected first must be used before the second set is requested. The following sample vector shows an application where both ced and cdpd digits are used.

CINFO Vector Example

In the example in Screen 6-5, ced and cdpd digits are both used to determine routing for the call.

```
 1. wait-time 2 secs hearing silence
 2. collect ced digits
 3. goto step 7 if digits = 1
 4. goto step 11 if digits = 2
 5. route-to number 0 with cov n if unconditionally
 6. stop
 7. collect cdpd digits
 8. route-to digits with coverage n
 9. route-to number 0 with cov n if unconditionally
10. stop
11. queue-to split 6 pri m
12. wait-time 10 secs hearing ringback
13. announcement 2564
14. wait-time 20 secs hearing music
15. goto step 13 if unconditionally
16. route-to number 0 with cov n if unconditionally
```

Screen 6-5. CINFO Example

In this vector, step 1 provides a wait-time step in case calls will be transferred to this vector. Step 2 collects the ced digits. Steps 3 and 4 branch the call to a different vector step depending upon the ced digit that was received. If no ced digits were received, or if the digit received was not 1 or 2, step 5 routes the call to the attendant. If the ced digit collected was 1, the call routes to a second collect step where cdpd digits are collected. The vector then routes the call to the cdpd digits. If the ced digit collected was 2, the call queues to split 6.

CINFO Interactions

- ASAI

  ced and cdpd digits can be passed to an ASAI adjunct as collected digits with the adjunct routing command and other event reports. ASAI will pass a maximum of 16 digits.

If a TTR is connected to a call as a result of ASAI-Requested Digit Collection, and the call encounters a collect ced or cdpd digits step, the TTR is disconnected from the call. In addition, any ASAI-requested digits stored in the collected digit buffer are discarded and no entered digits event report is sent.

ASAI does not distinguish between CINFO digits and user-entered digits that are collected as a result of a *collect digits* step. In other words, CINFO digits are provided to an ASAI adjunct but without any indication that they are anything other than collected digits from a vector.

The Call Offered to (VDN) Domain Event Report will contain the digits from the most recent *collect ced or cdpd* vector step.

- Best Service Routing™

  CINFO digits are included with the call if a multi-site BSR application routes the call to another switch.

- CentreVu® Call Management System (CMS)

  To administer ced or cdpd digits for the *collect digits* step via CMS, R3V5 or later CMS is required. The Vectoring (CINFO) customer option is not required for ced or cdpd digits to be passed to CMS. In addition, there is no CMS version requirement for passing ced or cdpd digits to CMS.

- Conference

  When a conference is established, the CINFO digits are merged into the call record of the conference. However, there is no indication of which party the digits originally belonged with. Therefore, for security reasons, when the first ISDN call drops out of the conference, the CINFO digits are erased.

- Look-Ahead Interflow

  CINFO digits are included with the call if Look-Ahead Interflow routes the call to another switch. The collect ced/cdpd step is neutral for Look-Ahead Interflow.

- Transfer

  If a call is transferred off the DEFINITY System, the CINFO digits are lost. If a call is transferred to an internal extension, CINFO digits are retained.

  If a call is transferred to a VDN, the CINFO digits should not be collected until the transferring party has had time to complete the transfer. If transfers are likely, a wait-time step of sufficient length is recommended before the collect step.

Look-Ahead Interflow

7

Look-Ahead Interflow (LAI) enhances Call Vectoring for call centers with multiple ACD locations. Look-Ahead Interflow allows these centers to improve call-handling capability and agent productivity by intelligently routing calls among call centers to achieve an improved ACD load balance. This service is provided via ISDN D-channel messaging over QSIG or non-QSIG private networks, virtual private networks, or public networks. The receiving switch is able to accept or deny interflowed calls sent by the sending switch.

DEFINITY 6.3 release includes an enhanced Look-Ahead Interflow that

- produces First in First Out (FIFO) or near FIFO call processing
- uses fewer computer processing resources during Look-Ahead Interflow than pre-DEFINITY 6.3 software
- includes enhanced information forwarding (codeset 0 user information transport).

Before You Start

The following items are criteria for basic LAI call control operation over virtual private or public switched networks:

- Sending and receiving call center locations must have ISDN (PRI or BRI) trunk facilities.
- DEFINITY must support the ISDN country protocol.
- LAI has been tested with several major carriers. To find out if these capabilities work with your carrier, check with your account team for the most current information. If testing has not been done to verify operation

over the public networks involved with the preferred specific configuration, use of private ISDN trunking between the nodes should be assumed until successful testing has been completed.

- The ISDN SETUP and DISCONNECT messages are transported between sending and receiving locations (for example, SS7 or equivalent public network connectivity).
- A receiving-end generated DISCONNECT message must transmit back to the sending DEFINITY call center without changing the cause value.

  Conversion of the DISCONNECT message to a progress message (with a Progress Indicator Description set to 1 and a Cause Value other than 127 included) is a valid reject message and compatible with LAI.
- Progress messages generated towards the sending end by intervening network switches must have the Progress Indicator Description set to 8 so that DEFINITY will not consider the call accepted or rejected.
- ISDN codeset 0 user information transport supports LAI information forwarding for DEFINITY 6.3 software. For passing pre-DEFINITY 6.3 information, LAI can use ISDN transport of a codeset 6/7 LAI IE. As another alternative, LAI can use dedicated VDNs at the receiving location to provide an equivalent display of the forwarding application identity and set trunk group options to not send either the codeset 6/7 LAI IE or codeset 0 information transport.

  NOTE:
  Best Service Routing cannot use these LAI alternatives. BSR must use ISDN codeset 0 user information transport.

A Quick Example

Look-Ahead Interflow is enabled through the use of call vectors and their associated commands. For a two switch configuration, these vectors are included in both the sending switch, which processes vector outflow, and the receiving switch, which processes vector inflow. The Two Switch Configuration for LAI is illustrated in Figure 7-1.

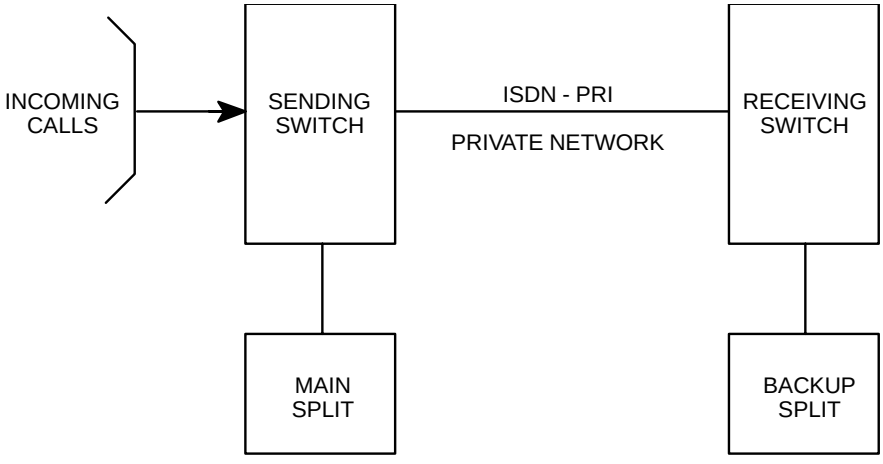


Figure 7-1. Two Switch LAI Connections

What is Look-Ahead Interflow?

Look-Ahead Interflow enhances call vectoring so that calls will interflow only to those remote locations that can accept the calls.

Look-Ahead Interflow is achieved through a set of vector commands. Table 7-1 lists the call-acceptance vector commands used in LAI.

Table 7-1. Call Acceptance Vector Commands

Call Acceptance Vector Command	Qualification
announcement	■ Announcement available ■ Queued for announcement ■ Retrying announcement
check split	■ Call terminates to agent ■ Call queued to split
collect digits	■ Always (except for ced and cdpd digits which are neutral)

Continued on next page

Table 7-1. Call Acceptance Vector Commands

Call Acceptance Vector Command	Qualification
converse-on split	■ VRU answers the call ■ Call queued to converse split
disconnect	■ With *announcement* and announcement available ■ With *announcement* and queued for announcement ■ With *announcement* and retrying announcement
messaging split	■ Command successful ■ Call queued
queue-to split	■ Call terminates to agent ■ Call queued to split
route-to	■ Terminates to valid local destination ■ Successfully seizes a non-PRI trunk ■ Results in a Look-Ahead Interflow call attempt, and the call is accepted by the far end switch
wait-time	■ Always (except *wait-time hearing i-silent* which is neutral)

If the receiving switch decides it is unable to accept the Look-Ahead Interflow call, call denial is accomplished by executing one of the vector commands listed in Table 7-2. One recommendation: use *busy* instead of *disconnect* to allow for compatibility with similar network services such as Alternate Destination Redirection (ADR).

Table 7-2. Call Denial Vector Commands

Call Denial Vector Command	Qualification
busy	■ Always
disconnect	■ With no announcement ■ With announcement but announcement unavailable
reply-best	■ Always—used with Best Service Routing™

The vector commands presented in Table 7-3 are considered neutral because they do not generate either call acceptance or denial messages.

Table 7-3. Neutral Vector Commands

Neutral Vector Command	Qualification
adjunct routing	■ Always
announcement	■ Announcement unavailable
check split	■ Call neither terminates nor queues
collect ced/cdpd digits	■ Always
consider	■ Always—used with Best Service Routing™
converse-on split	■ Call neither terminates nor queues
goto step	■ Always
goto vector	■ Always
messaging split	■ Command failure
queue-to split	■ Call neither terminates nor queues
route-to	■ Unsuccessful termination ■ Trunk not seized ■ Look-Ahead Interflow call denied by far end switch
stop	■ Always
wait-time hearing i-silent	■ Always **NOTE:** This command is used following an *adjunct routing* command in applications where the adjunct decides whether to accept or reject the Look-Ahead calls.

How Traditional Look-Ahead Interflow Works

Traditional Look-Ahead Interflow is recommended when the preferred call flow performs LAI attempts before queuing the call.

Look-Ahead Interflow feature uses the commands included within the Basic Call Vectoring and Call Prompting features:

- *Route-to number with coverage n* or *route-to digits with coverage n* command on a switch that has Look-Ahead Interflow optioned and that successfully seizes an ISDN trunk automatically results in a normal Look-Ahead Interflow call attempt being placed. The call attempt can be rejected or accepted by the remote end.
- *Route-to number with coverage y or route-to digits with coverage y* command never results in a Look-Ahead Interflow call attempt. The sending end assumes the call is always going to be accepted. This command always completes the call. Moreover, the command should not be used when the vector at the receiving location ends up denying the call, since the caller in this case would be given a busy signal, or the call will be disconnected. This command with coverage y should only be used for those cases when an unconditional interflow is wanted (with Look-Ahead Interflow active) and the terminating switch is set up accordingly.

When a Look-Ahead Interflow call attempt is made, Call Vectoring at the sending location checks a potential receiving location to determine whether to hold or send the call. While this is done, the call remains in queue at the sending location. As such, the call can still be connected to the sending-location agent (if one becomes available) before the receiving location accepts the call.

Call Vectoring at the receiving location decides whether to accept the call from the sending location or to instruct the sending location to keep the call. In the latter case, the sending location can then either keep the call, check other locations, or provide some other treatment for the call. Conditions for sending, refusing, or receiving a Look-Ahead Interflow call attempt can include a combination of any of the following: expected wait time for a split, number of staffed or available agents, number of calls in queue, average speed of answer or the number of calls active in a VDN, time of day and day of week, or any other conditional.

If the call is accepted by the receiving switch, the call is removed from any queues at the sending switch, and call control is passed to the receiving switch. If the call is denied by the receiving switch, vector processing simply continues at the next step at the sending switch. Until the call is accepted by either switch, the caller continues to hear any tones applied by the sending switch. Should the call be denied, the call vector may then apply alternate treatment, such as placing another Look-Ahead Interflow call to an alternate backup switch.

NOTE:
The Look-Ahead Interflow operation is completely transparent to the caller. While a Look-Ahead Interflow call attempt is being made, the caller continues to hear any audible feedback provided by the sending switch vector. The caller also maintains his or her position in any split queues until the call is accepted at the receiving switch.

In DEFINITY 6.3, Look-Ahead interflow passes Call Prompting digits collected in the sending switch to the receiving switch via codeset 0 user information transport. For more information, see Chapter 9, "Information Forwarding".

Example 1: Traditional LAI

The vector(s) in the sending switch use the *goto* command to determine whether the call should be sent to the receiving switch. Recall that the *goto* command tests various outflow threshold conditions (such as expected wait time). If the expressed condition is met, a branch is made to the appropriate *route to* command. This command sends the call to the receiving switch, which, as already noted, can accept or deny the call.

Screen 7-1 shows an example of an outflow vector that might be included in a sending switch.

```
1. wait-time 0 secs hearing ringback
2. goto step 5 if expected-wait for split 3 pri m < 30
3. route-to number 5000 with cov n if unconditionally
4. route-to number 95016781234 with cov n if unconditionally
5. queue-to split 3 pri m
6. announcement 3001
7. wait-time 30 secs hearing music
8. goto step 6 if unconditionally
```

Screen 7-1. Sending Switch Outflow Vector

If split 3 has an expected wait time of less than 30 seconds (step 2), step 5 queues the call to the split's queue at a medium priority.

If the expected wait time is 30 seconds or more, Look-Ahead Interflow attempts are made in steps 3 and 4. If the call is accepted by one of the receiving switches call control passes to the receiving switch.

If the receiving switches deny the call, the call queues to split 3 and announcement 3001 plays. The caller then hears music (interrupted by announcement 3001 every 30 seconds).

Receiving Switch Operation

When the receiving switch receives the Look-Ahead Interflow request, the call first routes to a VDN. The VDN then maps the call to the receiving switch's inflow vector, and vector processing begins, starting with inflow checking. Inflow checking is enabled via conditional *goto* commands in the inflow vector. The decision to accept or deny a call can be based on checks such as any of the following:

- Expected Wait Time
- Number of staffed agents
- Number of available agents
- Time-of-day/day of the week
- Number of calls in split's queue
- Average Speed of Answer
- Active VDN Calls
- ANI
- II-Digits
- CINFO ced and/or cdpd digits
- Collected digits forwarded from the sending switch (DEFINITY R6.3 only)

Once inflow checking is complete, acceptance of the Look-Ahead Interflow call is accomplished by executing any of the vector commands presented in Table 7-1.

NOTE:
For each command in Table 7-1, Table 7-2 and Table 7-3, only one of the corresponding qualifications need be true for the command to effect the desired result (call acceptance, call denial, or no effect on such acceptance or denial).

Screen 7-2 shows an example of an inflow vector that might be used by a receiving switch.

```
1. goto step 6 if expected-wait in split 1 pri h > 30
2. queue-to split 1 pri h
3. announcement 4000
4. wait-time 2 seconds hearing music
5. stop
6. busy
```

Screen 7-2. Receiving Switch Inflow Vector

Step 1 of this inflow vector checks the inflow thresholds. The *goto step* command in step 1 checks the expected wait time in split 1. If the expected wait time is greater than 30 seconds, a branch is made to the *busy* command in step 6. If executed, the *busy* command denies the call, and the receiving switch returns a call denial message to the sending switch. The sending switch, in turn, drops the Look-Ahead Interflow call attempt and then continues vector processing at the next vector step.

On the other hand, if the expected wait time in split 1 is less than or equal to 30 seconds, the receiving switch returns a call acceptance message to the sending

switch, and call control is passed to the receiving switch. Thereafter, the call is queued to split 1 in the receiving switch (step 2). Once queued, the caller receives the appropriate announcement in step 3 and is then provided with music until the call is answered by an agent or abandoned by the caller (steps 4 and 5). (Remember, the *stop* command halts vector processing but does not drop the call.)

If the sending switch does not receive a call acceptance or call denial message within 120 seconds (G3 switches) or 10 seconds (G2 or network switches) after the Look-Ahead Interflow call request, the Look-Ahead Interflow attempt is dropped, and the sending switch continues vector processing at the next step.

How Enhanced Look-Ahead Interflow Works

Enhanced Look-Ahead Interflow (DEFINITY 6.3 or later) uses the same basic vectoring commands as traditional Look-Ahead Interflow, but adds the new conditional *interflow-qpos*. Enhanced LAI is recommended when the preferred call flow performs LAI attempts after queuing the call.

Using this conditional

- produces First in First Out (FIFO) or near FIFO call processing
- uses less processing during Look-Ahead Interflow than pre-DEFINITY 6.3 software.

The simple way to achieve FIFO

You can use the *interflow-qpos* conditional in a *route-to* or *goto* command to achieve FIFO results.

For example, you can use the following *route-to* command with the conditional to achieve FIFO results:

```
route-to number 9581234 with cov n if interflow-qpos=1
```

If you have a lot of remote agents, you may want to set the *route-to* command as follows:

```
route-to number 9581234 with cov n if interflow-qpos<=2
```

For more information about using the *interflow-qpos* conditional, see the following section Detailed information about the new conditional.

Detailed information about the new conditional

Before you read this section, please note that you can use this feature without understanding the differences between “split queues” and eligible queues” or

between “interflow-qpos” and “queue position.” There are features built into enhanced LAI so that when you write a step such as “`route-to number 9581234 with cov n if interflow-qpos=1`,” the system will operate smoothly under all conditions.

The new conditional

The *interflow-qpos* conditional only applies interflow processes to a dynamic “eligible queue” and to calls that are queued locally before the *route-to* is attempted.

The “eligible queue” is that portion of the split/skill queue that

- **includes** only calls that are not expected to be answered locally during the interflow process at that moment relative to the call being processed
- **does not include** direct agent calls because these calls are excluded from any interflow process.

Here is the *interflow-qpos* conditional in a *route-to* command:

```
route-to number _____ with cov _ if interflow-qpos CM x
```

where

- CM is the comparator. It is one of three symbols: [=, <, <=].
  - With “if interflow-qpos = x”, the call is interflowed if it is at the x position from the top of the eligible queue.
  - With “if interflow-qpos < x”, the call is interflowed if it is among the top x-1 of the eligible queue.
  - With “if interflow-qpos <= x”, the call is interflowed if it is among the top x eligible calls.
- x indicates the call’s position in the eligible queue. Valid queue positions are 1 through 9. 1 is the top queue position. The eligible queue is made up of calls from the first local split/skill the call has been queued to due to previous steps in the vector.

NOTE:
Calls which are likely to be serviced locally before an LAI can be completed will not be eligible for interflow since they are excluded from the eligible queue. Calls that are likely to be answered are identified based on conditions of the split/skill to which the call is queued and an administered minimum EWT threshold value (under certain conditions).

Here is the *interflow-qpos* conditional in a *goto* command:

```
goto step/vector ____ if interflow-qpos CM x
```

where

- CM is the comparator. It is one of six symbols [=, <>, <, <=, >, >=]
- x indicates the call's position in the eligible queue. Valid queue positions are 1 through 9. 1 is the top queue position.

  Calls which are likely to be serviced locally before an LAI can be completed will not be eligible for interflow since they are excluded from the eligible queue.

When does a call not interflow?

A call does not interflow under the following circumstances:

- If the conditional is not met.

  As with other conditionals, the *route-to number... if interflow-qpos* step or the *goto step/vector* branch will only be executed if the conditional is met, otherwise vector processing will go to the next step.

- If the call is not in a split/skill queue or not in the eligible portion of the queue when the conditional step is executed.

  If the call is not in queue when the *route-to number... if interflow-qpos* step is executed, a vector event is logged and vector processing continues at the next step.

  If the call is not in queue when a *goto... if interflow-qpos* step is executed, the position in queue of the call is considered to be infinite in determination of the conditional.

  NOTE:
  A vector event is not logged if the call is in queue, but not in the eligible portion of the queue.

- Interflow failure or LAI rejection

  Interflow failure or LAI rejection will also go to the next step. Route-to operation and feature interactions will be the same as other configurations of the route to number command (e.g., *route to number ___ with cov _ if digit CM x*).

Table 7-4 outlines what action is taken for different cases of interflow eligibility.

Table 7-4. Actions taken for cases of interflow eligibility

Case	Action at *route-to* step	Action at *goto* step
Call not eligible for interflow	Never routed	Treat as if interflow queue position is infinite
Call is not in any split queue	Treat as if interflow queue position is infinite	Treat as if interflow queue position is infinite
Call is eligible for interflow	Act according to conditional	Act according to conditional

How is the minimum EWT set?

The minimum EWT threshold used to help determine which calls are more likely to be answered locally is administered via a new field for the Feature-Related System Parameters form. Minimum EWT is used when the local agents (in the first split/skill to which the call is queued) are handling a significant number of the calls. If not, the call is eligible for look-ahead interflow even if its EWT is lower than the threshold.

NOTE:
When enhanced lookahead interflow vectors or the lookahead EWT threshold are administered inappropriately, remote agents may experience phantom calls or a delay between becoming available and receiving an ACD call.

The instructions below assume that you use a SAT terminal or terminal emulator to administer the DEFINITY switch.

To set the minimum EWT threshold:

1. In the command line, enter **change system-parameters feature** and press RETURN.

   The Feature-Related System Parameters form comes up.

2. Go to page 7.

   If Lookahead Interflow is active, the `Interflow-Qpos EWT Threshold` field is administerable.

3. In the `Interflow-Qpos EWT Threshold` field, enter the number of seconds from **0** to **9** that you want for the EWT threshold (the default of 2 seconds is recommended).

NOTE:
When the lookahead EWT threshold field is set too low, remote agents may experience phantom calls.

4. Press ENTER to save your changes.

Example 2: Single-Queue Multi-Site Operation

In this scenario, all new calls for a given customer application are routed (by the public network) to only one of the switches in the network, where the calls are put in queue.

Local agents service the calls from the queue in the normal fashion; however, remote agents service calls by means of enhanced lookahead.

The switch with the call queue does rapid enhanced lookahead attempts to all other switches in the network which can service this call type, looking for an available agent.

Normally, the lookahead attempts are placed only on behalf of the call at the head of the queue ("`interflow-qpos = 1`"). However, in scenarios where there are large numbers of agents at a remote switch, it may be necessary to do interflows on behalf of more than one call in order to outflow a sufficient volume of calls to keep all agents busy ("`interflow-qpos <= 2`").

```
1. announcement 3501
2. wait-time 0 secs hearing music
3. queue-to skill 1 pri m
4. route-to number 93031234567 with cov n if interflow-qpos = 1
5. route-to number 99089876543 with cov n if interflow-qpos = 1
6. wait-time 5 secs hearing music
7. goto step 4 if unconditionally
```

Screen 7-3. Vector to Back up Split

In the above vector, interflow call attempts are placed every 5 seconds to the two other switches in the network, on behalf of the call at the head of the queue.

If queuing times are very long, 5 minutes for example, it is wasteful to go round the vector loop from steps 4 to 7 every 5 seconds when the call is nowhere near the head of the queue. For this reason, the Example 3 vector would be more efficient.

Example 3: Maintaining FIFO Processing with LAI

One of the advantages of enhanced LAI is the ability to provide FIFO or near-FIFO call processing. Screen 7-4 illustrates a vector used to achieve such call processing.

```
1. announcement 3501
2. wait-time 0 secs hearing music
3. queue-to skill 1 pri m
4. goto step 7 if interflow-qpos < 9
5. wait-time 30 secs hearing music
6. goto step 5 if interflow-qpos >= 9
7. route-to number 93031234567 with cov n if interflow-qpos = 1
8. route-to number 99089876543 with cov n if interflow-qpos = 1
9. wait-time 5 secs hearing music
10. goto step 7 if unconditionally
```

Screen 7-4. FIFO Processing Vector

In this vector:

- the rapid lookahead loop is only entered when the call has reached one of the top 8 positions in queue.
- the number of executed vector steps is reduced dramatically (whenever call waiting times are long).

It is important to write vectors so that calls at the head of the queue have advanced to the rapid lookahead loop by the time their turn to interflow has been reached. In the above example, if 8 calls can be serviced from queue in less than 30 seconds (the loop time on step 5), there could be a delay in outflowing calls to available agents at the remote sites.

Single-Queue FIFO Considerations

— When there are available agents, calls will always be delivered to available agents at the queuing switch before available agents at the remote switches.

— When there are calls in queue and agents serve calls from multiple applications, the agents will always service calls from the applications queued locally before calls from applications queued at another switch.

— Backup VDNs and vectors are recommended in order to provide continuous operation in the event of a failure at a queuing switch.

— EWT predictions cannot be made if the split/skill in which the calls are queued has no working agents.

— EWT predictions may be temporarily inaccurate if there are sudden, major changes in the number of working agents in the split/skill in which the calls are queued.

Example 4: LAI in a Tandem Switch Configuration

Tandem Look-Ahead Interflow is implemented by using *route-to* commands that contain external destinations that route over ISDN facilities. This configuration is illustrated in Figure 7-2.

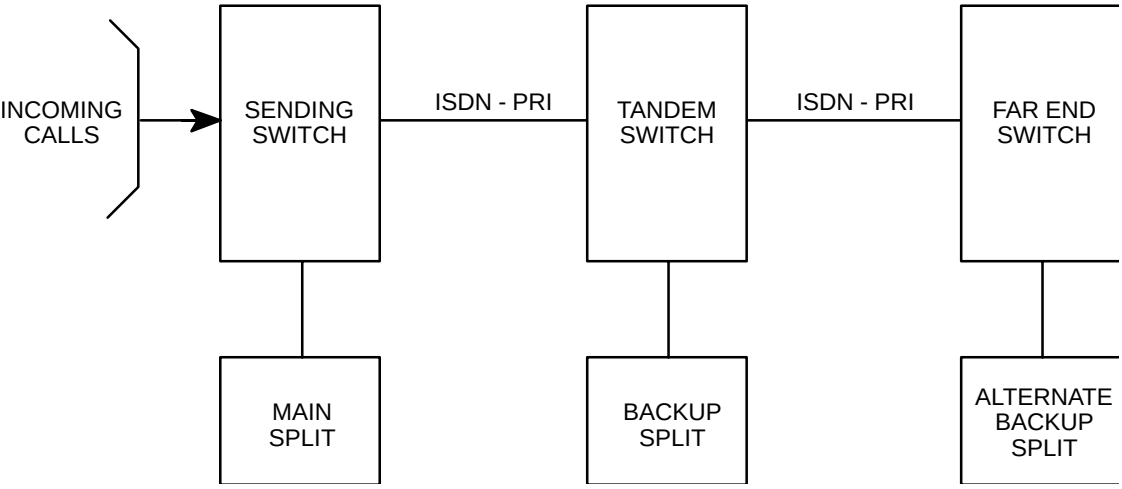


Figure 7-2. LAI Using a Tandem Switch

Sending Switch Operation

The sending switch is unaware that its Look-Ahead Interflow call is being tandemed to an alternate switch. The operation of the sending switch in the tandem switch configuration is the same as that in the two switch configuration.

Tandem Switch Operation

If the receiving switch executes a *route-to* command that routes the call over an ISDN facility before call acceptance, the *route-to* command is performed on a “look ahead” basis in the same manner as a sending switch. If the call is accepted at the far end switch, acceptance is passed to the sending switch, and

call control is passed to the far end switch, along with tandeming of the original calling party information and the original DNIS name. If the call is denied, the next step of the tandem switch vector is executed.

Screen 7-5 shows an example of a tandem switch vector.

```
1. goto step 6 if expected-wait in split 30 pri h > 30
2. queue-to split 30 pri h
3. announcement 200
4. wait-time 2 seconds hearing silence
5. stop
6. route-to number 4000 with cov n if unconditionally
7. busy
```

Screen 7-5. Tandem Switch Vector

Step 1 of this vector checks the inflow threshold. If the inflow criteria are acceptable, the vector flow drops to step 2, where the *queue-to split* command provides acceptance to the sending switch. Thereafter, steps 3 through 5 provide a typical queuing-wait scheme.

If, however, the inflow criteria are not acceptable, a branch is made to step 6. The *route-to* command in this step checks another switch enabled with Look-Ahead Interflow on a “look-ahead” basis. If this “far end” switch rejects the call, a denial message is relayed back to the tandem switch, which then drops the Look-Ahead Interflow call attempt. On the other hand, if the far end switch accepts the call, an acceptance message is relayed all the way back to the sending switch.

Note that no ringback is provided in this tandem switch vector. This is necessary so that an acceptance message is not returned to the sending switch. This operation is appropriate for the caller because the sending switch has already returned an announcement before a Look-Ahead Interflow attempt is made to the receiving switch.

Be sure the sending switch is not used as a backup location for the tandem switch or for any of the far end switches. Should the sending switch be administered in this manner, all trunk facilities could be tied up by a single call.

Far End Switch Operation

The far end switch is also unaware that tandeming has taken place. The far end switch functions in the same manner as the receiving switch within the two switch configuration.

DNIS and VDN Override in an LAI Environment

Look-Ahead Interflow handles Dialed Number Identification Service (DNIS) and VDN Override in various ways, depending on a number of different characteristics relevant to the call. Recall that DNIS (described in Chapter 2) allows any agent with a display-equipped voice terminal to receive visual displays that specify the name of the called VDN. Also recall that VDN Override in its basic form (as described in Chapter 2) allows the name of a subsequently routed to VDN to be displayed to the answering agent instead of the name of the originally called VDN.

The following sections discuss how Look-Ahead Interflow handles DNIS and VDN Override.

Answering Agent's Display

For Look-Ahead Interflow, the DNIS name (the "called" VDN name from the sending switch) is presented on the answering agent's display on the receiving switch if all of the following are true:

- Look-Ahead Interflow option is enabled
- Call routes to a VDN
- DNIS name field is not blank

The type of DNIS information that is displayed depends upon a number of different scenarios. This information is presented in Table 7-5.

Table 7-5. DNIS Information Displayed for LAI Scenarios

Scenario	Information Displayed
Tandemed Look-Ahead Interflow call	■ Look-Ahead Interflow DNIS information from the original Look-Ahead Interflow call
No redirection at the sending switch	■ VDN name according to Override rules at the sending switch (active VDN)
Redirection at the sending switch (VDN in coverage path)	■ Original VDN name, or ■ If multiple VDNs are accessed, the name of the VDN last accessed via a *route-to* command

Table 7-5. DNIS Information Displayed for LAI Scenarios

Scenario	Information Displayed
Sending switch sends a blank DNIS Name field (that is, a name is not assigned to the sending switch "called" VDN) or the trunk group is administered to not send the LAI name (see Chapter 9, "Information Forwarding").	■ Name associated with the receiving VDN. (This name can be changed according to the rules of VDN Override at the receiving switch.)

NOTE:
VDNs that map to vectors that place Look-Ahead Interflow calls must have their ISDN Calling Party Number (CPN) prefixes administered. If an ISDN CPN prefix is not administered, the assigned VDN name is not sent. Instead, a DNIS of all blank space characters is sent and displayed on the answering agent's terminal.

Originator's Display

For internal calls, the originator's display contains the same information as for Basic Call Vectoring. However, it is possible the originator might receive undesirable display updates during Look-Ahead Interflow call attempts. To avoid this scenario, ensure that the Look-Ahead Interflow calls are going out over trunk groups with the "Outgoing Display" field set to *no*. When the display field is set to no, internal callers calling that trunk group will see the digits that they dialed on their display.

LAI with Network ADR

Call Vectoring and Look-Ahead Interflow are compatible with and supplement the network services Alternate Destination Redirection (ADR) rerouting feature or equivalent service from other network providers. ADR uses ISDN-PRI connectivity with the DEFINITY System in the same manner as Look-Ahead Interflow to allow the receiving system to indicate whether a call is to be accepted or rejected. The same type of vector used as a receiving ACD for Look-Ahead Interflow is used at the ADR-receiving ACD. If the call is accepted, it is connected to the system. If the call is rejected, the network routing number is translated to another number, which routes the call to the alternate location within dialing-plan constraints. ADR allows for only one alternate location. Look-Ahead Interflow can be used at the alternate location to test other locations for less busy conditions.

Figure 7-3 shows configuration for a multilocation application.

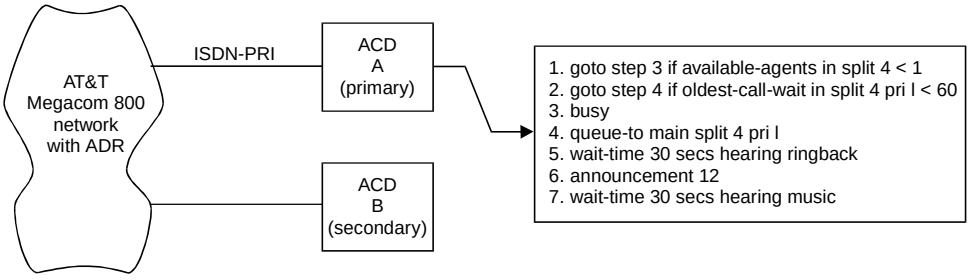


Figure 7-3. ADR Example

The network requires ISDN-PRI connectivity to primary location A. Connection to secondary location B may or may not be ISDN-PRI. ADR attempts to route the call to location A over the ISDN-PRI link using a routing number that selects a VDN assigned to the receiving vector shown. When the routing attempt is made, Call Vectoring starts processing the vector. The example then proceeds at location A as follows:

1. Step 1 checks for staffing of the ACD split, and branches to step 3 if it is not staffed.
2. If the ACD split is staffed, step 2 checks the oldest call waiting time in the split, and branches to step 4 if it is less than 60 seconds.
3. If the ACD split is unstaffed or if the oldest call waiting time is 60 seconds or more, step 3 rejects the call (returns a busy indication to the network).
4. If the oldest call waiting time is less than 60 seconds, step 4 accepts the call and queues it. ADR then connects the call through to the receiving system.
5. Steps 5-7 provide ringback, announcement, and music to the caller.

If the vector at location A rejects the call by sending a busy indication back to the network over the ISDN-PRI link, ADR reroutes the call to location B which must accept the call. If location B is closed or too busy to take the call, location B can use Call Vectoring and Look-Ahead Interflow to check other locations. If other locations exist and can take the call, location B can forward the call. If other locations do not exist or cannot take the call, location B can use Call Vectoring to route the call to location A. If location A is not open, location B can use Call Vectoring to provide an announcement or busy tone to the caller.

Multisite Applications for Enhanced LAI

Enhanced Look-Ahead Interflow has two principal applications in a multisite environment.

- It is possible to implement single-queue FIFO operation for any application. However, in many cases, Lucent recommends the use of Best Service Routing™ instead of LAI for maximum efficiency and flexibility. For more information, see Chapter 8, "Best Service Routing™".
- LAI can be used in combination with Best Service Routing for those switches in the network with extremely low call volumes.

  For more information about using BSR and LAI together, see Appendix D, "Advanced Multi-Site Routing".

Look-Ahead Interflow Considerations

The following are considerations you should keep in mind when working with Look-Ahead Interflow:

- Never interflow to a remote vector that in turn might interflow back to the same local vector. This could cause a single call to use up all available trunks.
- The *oldest-call-wait* test condition should not be used in LAI vectors. OCW corresponds to the very next call to be answered and, as such, this test condition gives no information on the current state of call overload (for example, if OCW = 30 seconds, all we know from this is that the queue was overloaded 30 seconds ago). In place of *oldest-call-wait*, use the EWT conditional. See "Expected Wait Time (EWT)" on page 5-2.
- If an LAI call attempt is accepted by a step that contains a *queue-to, check split,* or *route-to* command, there is a small but finite interval during which the call could be answered by an agent at the sending switch before notification of "acceptance" is received by the sending switch. In this case, the caller would be connected to the agent at the sending switch, while the agent at the receiving switch might receive a "phantom" call. For this reason, for traditional LAI operations you should consider using a short *wait-time* or *announcement* step at the receiving switch to allow the call to be accepted and taken out of queue at the sending switch. If call acceptance is to be based on available agents, use of a *wait-time* > 0 seconds or an *announcement* is not recommended. A *wait-time* with 0 seconds of silence might be useful in this case.

  NOTE:
  For enhanced LAI operation, there are capabilities built into the feature to eliminate or reduce the occurrence of phantom calls. If phantom calls are a problem in an enhanced LAI operation, the `Interflow-Qpos EWT Threshold` field has been set too low.

- When an LAI call attempt is made, the TTR (if attached) is disconnected, and any dial-ahead digits are discarded. This implies that a subsequent *collect digits* command would require that the TTR be connected.
- Be sure the feedback provided by the receiving switch after a successful LAI attempt is consistent with what the caller has already received.
- It is perfectly acceptable for a vector to route a call over an ISDN-PRI facility to a destination that is not a VDN. In such a case, the sending switch treats the call like a Look-Ahead Interflow call. Generic ISDN processing at the receiving switch causes the call to be accepted. The DNIS name is ignored.
- If a Look-Ahead Interflow call terminates to a VDN on a receiving switch where the Look-Ahead Interflow option is not enabled, intelligent interflow still results. However, any relevant DNIS information is ignored, and intelligent interflow to far-end switches is not possible.
- The LAI time-out in the sending switch occurs after two minutes.
- T-1 equipment might modify the ISDN D-channel that is used for Look-Ahead Interflow. If multiplexors are introduced into the ISDN-PRI circuit, bit compression and echo cancellation must be turned off for the D-channel.

Simple Troubleshooting for LAI

The following are troubleshooting suggestions when working with LAI:

- If remote agents are experiencing a high volume of phantom calls, the `Interflow-Qpos EWT Threshold` may be set too low or too high.
- If remote agents are experiencing a delay between becoming available and receiving a call the following may be the cause:
  - `Interflow-Qpos EWT Threshold` may be set too low
  - Insufficient LAI attempts from the sending switch. In this case, change *interflow-qpos* conditional at the sending switch. For example, change `interflow-qpos=1` to `interflow-qpos <= 2`.
  - Insufficient number of tie trunks.
- If remote agents are receiving no calls, the maximum number of vector steps executed at the sending switch vector may have been reached (before calls reached the head of the queue). In this case, rewrite the sending switch vector.

Best Service Routing™

8

An Overview of Best Service Routing

Best Service Routing (BSR) allows DEFINITY® ECS to compare specified splits or skills, determine which will provide the best service to a call, and deliver the call to that resource. If no agents are currently available in that split or skill, the call is queued. To respond to changing conditions and operate more efficiently, BSR monitors the status of the specified resources and adjusts call processing appropriately.

BSR can be configured for either single-site or multi-site operation. Single-site BSR compares splits or skills on the DEFINITY ECS where it resides to find the best resource to service a call. Multi-site BSR extends this capability across a network of DEFINITY switches, comparing local splits or skills, remote splits or skills, or both, and routing calls to the resource that provides the best service.

What's in This Chapter?

If you're not familiar with Best Service Routing (BSR), we suggest you read this chapter from the beginning. If you already understand BSR and want information about a specific aspect of the feature, use the table below.

Just want to…	Go to…
read a general overview of the benefits of BSR?	page 8-2
learn how single-site BSR works?	page 8-10
learn how multi-site BSR works?	page 8-21

read how to plan and administer single-site BSR on DEFINITY ECS? page 8-46

read how to plan and administer multi-site BSR on DEFINITY ECS? page 8-47

study advanced topics on multi-site BSR? page D-1

Benefits of Best Service Routing

Both single- and multi-site BSR intelligently compare specific resources to find the one that can best service a call. In addition, multi-site BSR allows you to integrate a network of call centers for better load balancing and optimal agent utilization. Depending on your specific application, BSR can yield a variety of other benefits as shown below.

Table 8-1. Best Service Routing Benefits

You can benefit from…	As a result of…
Increased revenue	■ Better agent utilization, thus allowing more calls to be handled with a given staff level. ■ Lower abandonment rates. By balancing the load between resources, BSR reduces extremes in wait times across local resources or across an entire network. ■ The ability to deliver calls to the best qualified or highest revenue generating agents (for centers with Expert Agent Selection)
Lower costs	■ Better agent utilization ■ Shorter trunk holding times ■ Reductions of ineffective interflows. ■ Operation over ISDN-BRI trunks and public networks.

Continued on next page

Table 8-1. Best Service Routing Benefits — *Continued*

You can benefit from…	As a result of…
Improved customer satisfaction	■ Interflowing calls from centers with a surplus of calls to centers with a surplus of agents. You can achieve uniform service levels across your network: all callers for a given application will experience approximately equivalent waiting times. ■ Shorter wait times. ■ The ability to deliver calls to the best qualified or highest revenue generating agents (for centers with Expert Agent Selection) ■ Lower abandonment rates. By balancing the load between resources, BSR reduces extremes in wait times across local resources or across an entire network. ■ Robust information forwarding capabilities. Multi-site BSR can forward original service requirements and any caller-entered digits with each call, and can use both QSIG and non-QSIG information transport methods over private or public networks.
Increased performance and more efficient trunk usage	■ Less messaging and processing required per call than in traditional LAI scenarios. ■ Eliminates phantom calls to remote agents ■ Intelligent interflows that only route calls to centers with available agents.
BSR's easy configuration	■ Simple vector commands. You don't need to learn complex programming languages or design comparison steps. Simply list the local and remote resources to be considered for calls and instruct DEFINITY ECS to queue or deliver the call to the best resource on the list.

Continued on next page

Table 8-1. Best Service Routing Benefits — *Continued*

You can benefit from…	As a result of…
Improved agent productivity	■ Increased efficiency Improve your service without adding staff, or reduce staff while maintaining your current level of service. Network-wide load balancing means that agents at one location are less likely to sit idle while calls wait in queue at another location. ■ No call delivery delays In contrast to approaches that queue calls at all remote centers simultaneously, with BSR there is no delay in delivering a call when an agent becomes available.
Increased operating flexibility, easier staffing and scheduling	■ Larger pool of agents available to take calls in a split or skill. Through its network-wide call distribution and information forwarding, BSR effectively converts distributed locations into a virtual call center. Thus, staffing problems do not need to be solved on a center-by-center basis. BSR can automatically react to staff shortages at one center by routing more calls to other locations. ■ Automatic management of sudden and unexpected increases in call volume Spikes in call volume for a single split or skill can be distributed across other splits or skills. Spikes in call volume at a single call center can be distributed across all centers (provided that sufficient trunk capacity is available between switches).
Improved service levels	■ Lower average speed of answer (ASA)[1]

1. If a call center network is heavily overloaded and a significant number of calls are being blocked or abandoned, shorter wait times may not result when BSR is used. Rather than reducing wait times, any productivity gains will allow more calls to gain access to the network.

Before You Start

For single-site BSR applications, your switch must meet the requirements shown below (except for ISDN trunks and LAI). To use multi-site BSR applications, all switches involved, and the network connecting them, must meet all the requirements described in this section.

CAUTION:

To ensure your network meets the requirements for BSR support presented below, contact your Account Executive about BSR network certification.

Switch Requirements

Your switch has to meet the requirements shown below to support Best Service Routing. Check these settings before you try to use BSR.

Table 8-2. Requirements to Use Best Service Routing

Form	Page	Field	Must be set to...
Software Version	1	Memory Resident	G3V6i.03 or G3V6r.03
System-Parameters Customer-Options	1	G3 Version	V6 or higher
	2	ISDN-BRI Trunks[1]	Y
		ISDN-PRI Trunks[1]	Y
	3	Vectoring (G3V4 Advanced Routing)	Y
		Vectoring (Best Service Routing)	Y
		Lookahead Interflow (LAI)[2]	Y
Feature-Related System Parameters	8	Adjunct CMS Release	R3V6 or higher, or left blank

1. Multi-site BSR operates over both BRI and PRI trunks. ISDN connectivity is only necessary if you want to use multi-site BSR, in which case one or both of these fields must be set to "Y."
2. Look-Ahead Interflow is only necessary if you want to use multi-site BSR.

NOTE:
If you begin using BSR and then decide to turn it off, you will not be able to set `Vectoring (Best Service Routing)` to **N** until you remove all BSR commands from vectors. If you're using multi-site BSR with Look-Ahead Interflow and want to turn LAI off, you will not be able to set `Lookahead Interflow (LAI)` to **N** until you remove all *consider location*, *reply-best,* and *interflow q-pos* commands from vectors.

Network Requirements

To support multi-site BSR, networks must meet both the criteria for LAI call control operation over switched networks (see Chapter 7, "Look-Ahead Interflow") and the following criteria:

- The network must support end-to-end transport of codeset 0 user data, either as a User-to-User Information Element (UUI IE) or by QSIG Manufacturer Specific Information (MSI IE), in the ISDN SETUP and DISCONNECT messages. (For more information, see "Determining User Information Needs" on page 9-7.)

  Private networks can be configured for either QSIG (using MSI packaged in codeset 0 Facility IEs) or non-QSIG (using a codeset 0 UUI IE) transport. Currently, public networks do not support QSIG and user data can only be transported via the UUI IE when supported by the network. Future public network offerings may support QSIG, possibly by Virtual Private Network.

- DEFINITY ECS must support the ISDN country protocol.

- The network byte limit for user information contents (the user data portion) must be large enough to carry the data needed for the customer application.

  NOTE:
  Some public network providers may require service activation and/or fees for user information transport.

BSR, LAI, enhanced information forwarding, and UCID have been tested with several major carriers. To find out if these capabilities work with your carrier, check with your account team for the most current information.

If testing has not been done to verify operation over the public networks involved with the preferred specific configuration, use of private ISDN trunking between the nodes should be assumed until successful testing has been completed.

Terms to Know

Some of the following terms are familiar to current users of DEFINITY ECS and call vectoring, while others are new and apply only to BSR. Understanding these terms will be helpful as you read through the material in this chapter. Table 8-3 contains terms pertaining to both single-site and multi-site BSR.

Table 8-3. Definitions of BSR Terminology

Term	Definition
adjusted EWT	Expected Wait Time plus a user adjustment set by a *consider* command.
agent selection method	The method DEFINITY ECS uses to select an agent in a hunt group when more than one agent is available to receive the next call: ■ UCD-MIA ■ UCD-LOA[1] ■ EAD-MIA ■ EAD-LOA The agent selection method is a property of hunt groups and is set in the Group-Type field on the Hunt Group form.
application	A general term for a system in any call center that handles calls of a particular type. In relation to BSR, any specific implementation of multi-site Best Service Routing.
application plan	Used only in multi-site applications, the application plan identifies the remote switches that may be compared in consider series. The plan also specifies the information used to contact each switch and to interflow calls to it.

Continued on next page

Table 8-3. Definitions of BSR Terminology — *Continued*

Term	Definition
best	**No agents available** When no agents are available in any of the specified splits or skills, the “best” resource is the one with the lowest adjusted EWT. **Agent available in one resource** When an agent is available in one and only one of the splits/skills specified in a consider series, that agent is the “best’ and the call will be delivered to that agent. If the BSR Available Agent Strategy is **1st-found**, BSR will ignore all subsequent steps in the consider series. If any other available agent strategy is used, all remaining resources will still be considered before the call is delivered. **Agents available in two or more resources** When agents are available in two or more splits or skills, the “best” agent is the one that best meets the criteria specified in the BSR Available Agent Strategy. For example, if the available agent strategy is UCD-MIA, the best agent out of those available will be the agent with the longest idle time.
Best Service Routing (BSR)	A DEFINITY ECS feature, based on call vectoring, that routes ACD calls to the resource best able to service each call. Best Service Routing can be used on a single switch, or it can be used to integrate resources across a network of DEFINITY Enterprise Communication Servers.

Continued on next page

Table 8-3. Definitions of BSR Terminology — *Continued*

Term	Definition
BSR Available Agent Strategy	A field that appears on the VDN form when either version of BSR is enabled. The entry in this field is a property of the VDN and its assigned vector. Possible entries are: ■ 1st-found ■ UCD-MIA ■ UCD-LOA ■ EAD-MIA ■ EAD-LOA When the VDN is the active VDN for a call, as determined by VDN Override, this field determines how BSR commands in the vector identify the best split or skill when several have available agents.
consider series	*Consider* commands are typically written in a set of two or more.[2] This set of *consider* commands is called a consider series.
consider sequence	A consider series plus a *queue-to best, check-best, or reply-best* step is called a consider sequence.
Expected Wait Time (EWT)	Expected Wait Time is an estimate of how long a call in queue will have to wait before it is connected to an agent.
Intelligent polling	An automatic feature of BSR that significantly reduces the number of status polls executed. When a remote location cannot be the best resource at a given moment in time, the intelligent polling feature temporarily suppresses polls to that location.
interflow	To route an incoming call to an external switch without answering it at the origin switch.

Continued on next page

Table 8-3. Definitions of BSR Terminology — *Continued*

Term	Definition
poll suppression	A component of BSR intelligent polling that eliminates wasteful polling of remote locations which have returned poor adjusted EWTs.
resources	An agent, split, skill, or location
status poll	A call placed by a *consider location* vector command to obtain status data from a remote location in a multi-site BSR application.

1. LOA is a new agent selection method available as an option on DEFINITY ECS R6.3 and later switches. You must have CentreVu® Advocate in order to use LOA. For more information, please see the *CentreVu® Advocate User Guide* (585-215-855).
2. A consider series in a status poll vector might have just one *consider* step.

Single-Site BSR

Single-site BSR is a simple, logical extension of call vectoring. Like any other vector, vectors with BSR commands are assigned to one or more VDNs. Using new vector commands and command elements, you tell DEFINITY ECS to compare, or “consider,” specific splits or skills for each call processed in that particular vector. Throughout the comparison, DEFINITY ECS can remember which resource is the best based on how you’ve defined “best.” BSR vectors can deliver a call to the first available agent found, or they can consider all of the specified resources and deliver the call to the best split or skill. If no agents are available in any split or skill, the call is queued to the split or skill with the shortest adjusted EWT.

The Elements of Single-Site BSR

Table 8-4 shows the forms, the vectors, and the vector commands and command elements needed to use single-site BSR. The table also briefly describes the purpose of each component.

Table 8-4. The Elements of Single-Site BSR

Components		Use this…
Forms	Vector Directory Number form	■ to link a VDN to a BSR vector ■ to set the agent selection strategy that will be used for all calls to that VDN
	Call Vector form	■ to confirm that BSR is optioned ■ to write vectors that use BSR commands
Commands	*consider split/skill*	to obtain the EWT and agent data needed to identify the best local split or skill. One *consider* step must be written for each split or skill you want to check.[1]
	queue-to	with the *best* keyword to queue calls to the best resource identified by the consider sequence.
	check	with the *best* keyword to queue calls to the best resource identified by the consider sequence if the resource meets certain conditions.
Keyword	*best*	in *queue-to, check,* and *goto* commands that refer to the resource identified as best by a series of consider steps
Conditional	*wait-improved*	in *check* and *goto* commands to prevent calls from being queued to backup splits or skills unless the reduction in EWT is significant.
User adjustment	*adjust-by*	to prefer one split, skill, or location over another by adding a set value to the EWT for that resource.[2]

1. Since the *consider* command is designed to compare two or more resources, *consider* commands are typically written in a series of two or more with the sequence terminating in a *queue-to best* step. This set of *consider* commands and a *queue-to best* step is called a consider sequence.
2. Of course, DEFINITY ECS doesn't increase the actual wait time in a split or skill. It only adjusts EWT in the calculations used to identify the "best" split or skill to which to queue a call.

How BSR Determines the Best Resource

BSR determines the best resource to service a call by examining one or all of the following variables:

- the EWT of the resource
- any user adjustments
- the availability of agents
- the selection strategy for the active VDN[1]

Call Surplus Situations

Any BSR application compares a set of predetermined resources (local splits or skills, remote splits or skills, or both) and selects the “best” resource to service the call. In a call surplus situation (no agents available), the best resource is the split or skill with the lowest adjusted Expected Wait Time (EWT). For purposes of calculating the best resource in a call surplus situation, BSR allows you to adjust the EWT value for any split or skill. The actual EWT for calls in queue isn’t changed, of course; only the value used in the calculations performed by the BSR feature is changed. You don’t have to enter adjustments, but the ability to adjust the EWT for splits or skills allows you to program preferences in vectors. Because of agent expertise, for example, you might prefer that some resources *not* service a call unless doing so significantly decreases the call’s time in queue.

When agents are available in one or more of the specified resources, BSR does not consider EWT adjustments in selecting an agent for a call.

Agent surplus situations

In an agent surplus situation (one or more agents available to take incoming calls), BSR will deliver a new call according to the BSR Available Agent Strategy specified on the VDN form for the active VDN. The “best” resource will be the split or skill that meets the criteria defined by the strategy you’ve chosen. BSR can use any of the five strategies shown in the table below to select an agent when agents are available.

1. The BSR Available Agent Strategy that applies to a given call is the strategy assigned to the active VDN for that call, as determined by VDN override.

Table 8-5. BSR Available Agent Strategies

If BSR Available Agent Strategy is set to…	The call will be delivered to…
1st-found	the first available agent. BSR will not consider any other resources as soon as it finds an available agent.
UCD-MIA	the agent who has been idle the longest. BSR will compare all the splits or skills specified in the vector before delivering the call.
EAD-MIA	the agent with the highest skill level who has been idle the longest. BSR will compare all the splits or skills specified in the vector before delivering the call.
UCD-LOA	the least-occupied agent.[1] BSR will compare all the splits or skills specified in the vector before delivering the call.
EAD-LOA	the agent with the highest skill level who is the least occupied. BSR will compare all the splits or skills specified in the vector before delivering the call.

1. LOA is a new agent selection method available as an option on DEFINITY ECS R6.3 and later switches. For more information, please see the *CentreVu® Advocate User Guide* (585-215-855).

When agents are available in one or more of the specified resources, BSR does not consider EWT adjustments in selecting an agent for a call.

NOTE:
For greatest efficiency, the agent selection method used in the splits or skills considered by a BSR vector should match the BSR Available Agent Strategy assigned to the active VDN.

Example 1: Basic Single-Site BSR

BSR Available Agent Strategy	1st-found
BSR Commands & Command Elements	consider split, queue-to best
Splits/Skills Considered	2
User Adjustments?	No

This example shows the simplest use of Best Service Routing. The central element of all single-site and multi-site BSR is a VDN/vector pair. The vector contains the commands that actually process the call, but the active VDN for the call contains information used by some vector steps. For single-site BSR, the active VDN for a call sets the available agent strategy used by the vector.

```
                                                  Page  1 of   1
                    VECTOR DIRECTORY NUMBER

                      Extension: 5000
                           Name: Single-site BSR
              Allow VDN Override? n
                            COR: 59
                             TN: 1
                   Vector Number: 234
                      AUDIX Name:
           Messaging Server Name:
                       Measured: none
  Acceptable Service Level (sec):
   VDN of Origin Annc. Extension: 301
                       1st Skill:
                       2nd Skill:
                       3rd Skill:

                  Return Destination:
          VDN Timed ACW Interval:

    BSR Available Agent Strategy: 1st-found
```

Screen 8-1. BSR Example 1: VDN Form

The VDN form above shows VDN 5000, the VDN we'll use in this example. Note the **1st-found** entry in the `BSR Available Agent Strategy` field: if vector 234 uses BSR commands, as soon as a *consider* step locates a resource with an available agent any subsequent *consider* steps will be skipped and the call will be delivered to that resource. Resources specified in any subsequent *consider* commands won't be checked. If no split has an available agent, the call will be queued to the split with the lowest adjusted EWT.

Also note that `Allow VDN Override?` is set to **n**. If a second VDN and vector are used to process this call, the 1st-found strategy specified in VDN 5000 will still be used.

The screen below shows vector 234, which compares two splits. No adjustment is assigned to either resource, indicating that both splits are equally suited to service calls since neither is preferred to the other. In reality, such a vector would probably have additional steps after step 4 (such as *announcement* or *wait-time* commands). We've omitted such steps in this example for purposes of clarity.

```
1. wait time 0 secs hearing ringback
2. consider split 1 pri l adjust-by 0
3. consider split 2 pri l adjust-by 0
4. queue-to best
```

Screen 8-2. BSR Example 1: Vector

Notice that the *consider* commands follow each other in unbroken sequence and that the *queue-to best* command immediately follows the last *consider* command. This structure is called a "consider series," and Lucent recommends that you typically write such series in uninterrupted order. A few commands, such as the *goto* command, which cause little if any delay in the execution of the *consider* steps, may be used. In general, however, you shouldn't put other commands between *consider* steps, or between a *consider* and a *queue-to best* step. Even if Best Service Routing still works in that situation, you might seriously impair the performance of the vector.

Consider commands collect and compare information. When a call is processed in the vector above, the first consider step collects and temporarily saves the following information about split 1:

- The fact that split 1 is a local split
- The queue priority specified in the *consider* step
- The user adjustment specified in the *consider* step
- The split's
  - — split number
  - — Expected Wait Time

If EWT=0 (one or more agents are available), the step also collects all the agent information that might be needed by the BSR Available Agent Strategy:

- Agent Idle Time (AIT)
- Agent Occupancy (AOC)
- Skill level of the agent in the split or skill who will receive the next call

In this example, neither split has an available agent when the consider series executes. If one did, the call would be delivered to that split by the *queue-to best* step. Since there are no available agents in either split, the complete set of saved data now defines the "best" resource—for the moment. The second consider step collects the same data and compares it to the current "best" data. For this example, let's say that split 1's EWT is 40 sec. and split 2's is 20 sec. When the second *consider* step executes, its data will replace the "best" data from step 1 because its adjusted EWT is lower. The "best" data is essentially a placeholder. When a *queue-to best* step executes, it simply reads the data that's saved as the "best" at that moment and queues the call to that split. In this case, the best data was collected from split 2, so the call will be queued to split 2 at the specified priority.

Questions.

What if there were available agents in both splits?

Since the BSR Available Agent Strategy in this example is 1st-found, the consider series would have skipped any *consider* steps after step 2 and the *queue-to best* step would have delivered the call to split 1—the first split with an available agent found by the vector.

In any BSR vector, the order of the *consider* steps should reflect your preferences for the resources considered. Put the step that considers the most preferred split or skill first, the step for your second preference second, and so on in the consider series.

What if there were several available agents in split 1? Which agent would get the call?

This is a very important point: when more than one agent is available in a split, the BSR *consider* command collects agent data only for the agent who will receive the next call to that split. This agent is identified according to the agent selection method specified in the `Group-Type` field on the Hunt Group form.

NOTE:
For greatest efficiency, the agent selection method used in the splits or skills considered by a BSR vector should match the BSR Available Agent Strategy assigned to the active VDN.

User Adjustments in Single-site BSR

You may have preferences as to which splits or skills should answer certain types of calls. In both single- and multi-site BSR, the *adjust-by* portion of the *consider* command allows you to program these preferences into your vectors.

You can assign a value of 0–100 in user adjustments. The units of this value are supplied by the switch depending on the conditions whenever that consider step executes. For example, in the command *consider split 1 pri h adjust-by 20,*

DEFINITY ECS interprets *adjust-by 20* to mean "add 20% to the EWT, but add at least 20 seconds."[2] For Expected Wait Times of 1–100 seconds, an adjustment of 20 will therefore add 20 seconds. Above 100 seconds, the same adjustment will add 20% to the EWT for the split or skill specified in the *consider* step. Table 8-6 shows the results of applying a constant adjustment to a range of Expected Wait Times.

Table 8-6. User Adjustments in BSR

EWT of resource (sec.)	User adjustment	Adjustment applied by DEFINITY ECS (sec.)	Adjusted EWT used to select resource
10	20	20	30
60		20	80
120		24	144
300		60	360

Example 2: Single-Site BSR with Adjustments

BSR Available Agent Strategy	EAD-MIA
BSR Commands & Command Elements	consider skill, queue-to best
Splits/Skills Considered	4
User Adjustments?	Yes

This example shows a more complex implementation of single-site BSR. Four skills in an Expert Agent Selection environment are compared. The Expected Wait Time (EWT) for some skills is adjusted to reflect the administrator's preferences.

2. If the user adjustment were defined as a number of seconds, BSR would not be efficient when EWT was high. If the user adjustment were defined as a percentage, BSR would not be efficient when EWT was low. Such efficiencies, while always important, become critical in multi-site BSR applications where issues of trunk cost and capacity are involved.

```
                                                    Page  1 of   1
                      VECTOR DIRECTORY NUMBER

                         Extension: 5001
                              Name: Single-site BSR
                Allow VDN Override? n
                               COR: 59
                                TN: 1
                      Vector Number: 11
                         AUDIX Name:
             Messaging Server Name:
                          Measured: none
   Acceptable Service Level (sec):
    VDN of Origin Annc. Extension: 501
                         1st Skill:
                         2nd Skill:
                         3rd Skill:

                Return Destination:
           VDN Timed ACW Interval:

     BSR Available Agent Strategy: EAD-MIA
```

Screen 8-3. BSR Example 2: VDN Form

The VDN form above shows VDN 5001, the VDN we'll use in this example. Note the **EAD-MIA** entry in the BSR `Available Agent Strategy` field: if vector 11 uses BSR commands, calls will not automatically be delivered to the first resource found with an available agent. All consider steps in vector 11 will be executed, and one of the following things will happen:

If …	Then…
no skill has an available agent	the call will be queued to the skill with the lowest adjusted EWT.
only one skill has an available agent	the call will be delivered to that skill.
two or more skills have available agents	the call will be delivered to the skill with the most expert agent.
two or more skills have available agents with the same skill level	the call will be delivered to whichever of these agents has been idle the longest.

Also note that `Allow VDN Override?` is set to **n**. If a second VDN and vector are used to process this call, the, the EAD-MIA strategy specified in VDN 5001 will be used. Were `Allow VDN Override?` set to **y** and vector 11 routed some calls to another VDN, the subsequent VDN's available agent strategy would govern the operation of *consider* steps in its vector.

The screen below shows vector 11, which compares four skills.

```
1.  wait-time  0 secs hearing ringback
2.  consider skill  1 pri l adjust-by 0
3.  consider skill  2 pri l adjust-by 30
4.  consider skill 11 pri l adjust-by 30
5.  consider skill 12 pri l adjust-by 30
6.  queue-to best
7.  wait-time 10 secs hearing ringback
8.  announcement 1001
9.  wait-time 30 secs hearing music
10. goto step 8 unconditionally
```

Screen 8-4. BSR Example 2: Vector

For this example, let's say that the Expected Wait Times of the four skills are 95, 60, 180, and 50 seconds, respectively. Notice that all consider steps except the first adjust the EWT returned by the specified skill. Skill 1 is the preferred skill to handle calls to VDN 5001, so its EWT is not adjusted. Skills 2, 11, and 12 can handle this call type, but they are not preferred. The adjustment of 30 means that, in call surplus situations, these skills will not handle calls to VDN 5001 unless their EWT is at least 30 seconds better than the EWT in skill 1.

Table 8-7 shows the adjustments that would be applied to each skill given its EWT and the user adjustment specified in the *consider* step. The last column shows the adjusted EWT the switch will use to select a skill for the call.

Table 8-7. BSR Example 2: User Adjustments

Skill #	User adjustment in consider step	Actual EWT (sec.)	Adjustment applied by DEFINITY ECS (sec.)	Adjusted EWT used in BSR calculations (sec.)
1	0	95	0	95
2	30	60	30	90
11	30	180	54	234
12	30	50	30	80

Since the available agent strategy is not 1st-found, all four consider steps are executed each time the vector processes a call. In this example there are no available agents in any of the skills. In fact, EWT is high enough in the first three skills for DEFINITY ECS to queue the call to skill 12.

When the *queue-to-best* step executes, the data in the best data placeholder is the data from skill 12 and so the call is queued to that skill. From this point on, if the call is not answered during the execution of step 7, a common vector loop regularly repeats an announcement for the caller while he or she waits in queue.

Questions

What if there were an available agent in one skill? Would user adjustments be applied?

Since the BSR Available Agent Strategy in this example is EAD-MIA, the entire consider series will always be executed to check all the skills for available agents. If only one skill has available agents, the call is delivered to that skill and user adjustments are not applied.

What if there were available agents in two skills, say, skills 1 and 2? Which skill would get the call? Would user adjustments be applied?

Since the BSR Available Agent Strategy for VDN 5001 (the active VDN) is EAD-MIA, the call would be delivered to the skill with the most expert agent. If there were available agents in both skills with the same skill level, their idle times would be compared and the call would go to the skill with the agent who's been idle the longest.

If a split or skill has more than one available agent, remember that it is the split or skill's agent selection method that determines which agent's data will be used in BSR's selection of the best resource.

What if no agents were staffed in a skill? Will DEFINITY ECS recognize this?

Yes. Under any of the following conditions, the EWT returned from a skill (or split) will be infinite:

- no agents logged in
- no queue slots available
- all agents in AUX work mode

DEFINITY ECS will log a vector event and go to the next vector step without changing the data in the best placeholder. A resource with an infinite EWT will never be selected as the best resource.

Can VDN skills be used in consider steps?

Yes. For example, *consider skill 1st [2nd, 3rd] pri m adjust-by 0* will collect data on the 1st [2nd, 3rd] skill as defined for the active VDN.

Simple Troubleshooting for Single-Site BSR

You should regularly execute a **display events** command for the appropriate vectors—especially if you've just implemented a new BSR application. Vector events will identify and indicate the source of common malfunctions and administration errors.

For a list of BSR vector events and definitions, see "Tracking Unexpected Vector Events" on page E-16.

NOTE:
Only the most recent events are displayed when a **display events** command is executed. Periodic display of vector events will help you quickly identify problems.

To verify that your BSR vectors are operating as intended, use a **list trace vdn** or **list trace vec** command to observe processing of an individual call. Refer to "Tracing Call Flow" on page E-32 for more information.

Multi-site BSR

Multi-site BSR includes all the capabilities of single-site BSR and extends these across a network of DEFINITY switches. Multi-site BSR can compare local splits or skills, remote splits or skills, or both, and route calls to the resource that provides the best service. In addition, multi-site BSR has special features that work to ensure efficient use of processor power and network resources in your BSR applications.

Throughout the rest of this chapter, the words "local," "origin," and "remote" are used to label different switches in multi-site applications. These words may seem to suggest that only one switch (the "local" or "origin" switch) in a network is receiving calls, polling other ("remote") switches, and interflowing calls. While such a centralized system may sometimes be useful, in most networks with BSR every switch in the network will be interflowing calls to other switches and receiving interflowed calls from other switches. For clarity in the following discussions, "local" or "origin" simply means a switch that is considering whether to interflow a call. "Remote" means any switch that may be polled by this first switch and thus might receive the interflowed call. More generally, these terms are relative to the BSR applications you design. In terms of a given application, the "local" or "origin" switch is the switch on which the Application Plan form for this application resides, and the "remote" switches are the switches identified at the locations listed on the form. See "Multi-site BSR Applications" on page 8-26 for an explanation of multi-site BSR applications.

When each switch in a network may interflow calls to other switches and receive interflows, this is called a distributed system. A centralized system, by contrast, is one in which all calls are initially delivered to a single call center (the "hub")

and distributed from this site to queues at remote switches. A centralized system requires greater inter-switch trunking, since a greater percentage of calls need to be redirected. However, it may be an appropriate configuration if your organization has a significant investment in VRU and CTI technology at the hub.

NOTE:
The following material on multi-site BSR assumes you already understand the operation of single-site BSR.

The Elements of Multi-site BSR Applications

Table 8-8 shows the forms, the vectors, and the special vector commands and command elements that you use to administer multi-site BSR applications. The table also briefly describes the purpose of each component. Since all the elements of single-site BSR may be used in multi-site applications, these are included in the table for convenience.

Table 8-8. The Elements of Multi-Site BSR Applications

Components		Use this to…
Forms	Best Service Routing Application Plan form	■ define the group of remote sites that will be polled by a specific application ■ assign a unique name and number to each application ■ assign routing numbers for the status poll and interflow VDNs
	Vector Directory Number form	■ link a VDN to a BSR application via its application number ■ link the VDN to a BSR vector ■ set the agent selection strategy that will be used for all calls to that VDN
	Call Vector form	confirm that BSR is optioned and program the vector steps for BSR
	ISDN Trunk forms	tell DEFINITY ECS whether to forward user information via Shared UUI or QSIG MSI
	List Best Service Routing Applications form	display a list of all the BSR applications by name and number
	System Capacity	monitor the number of BSR application- location pairs assigned in your system

Continued on next page

Table 8-8. The Elements of Multi-Site BSR Applications — *Continued*

Components		Use this to…
VDNs and Vectors	Primary VDN (the active VDN for the call at the origin, as determined by VDN override)	define the application plan and available agent strategy that are used by the vector assigned to this VDN
	Primary vector	control call processing at the original switch and compare local and remote resources
	Status poll VDN/vector	respond to status poll calls from another switch. The status poll vector considers a set of local splits or skills and returns data on the best resource to the original switch.
	Interflow VDN/vector	accept BSR calls from another switch and queue them to the best of the local resources considered

Continued on next page

Table 8-8. The Elements of Multi-Site BSR Applications — *Continued*

Components		Use this to…
Commands	*consider split/skill*	obtain the EWT or agent data needed to identify the best resource on the local switch. One *consider* step must be written for each split or skill you want to check.
	consider location	obtain the EWT and agent data from a remote location needed to identify its best resource. One *consider* step must be written for each location you want to check.
	reply-best	return data to another switch in response to a status poll
	queue-to	with the *best* keyword to queue or route calls to the best resource identified by the consider sequence.
	check	with the *best* keyword to queue or route calls to the best resource identified by the consider sequence if the resource meets certain conditions.
Keyword	*best*	write *queue-to, check,* and *goto* commands that refer to the resource identified as best by a series of consider steps. *Goto best* is for special applications and not used in all BSR vectors.

Continued on next page

Table 8-8. The Elements of Multi-Site BSR Applications — *Continued*

Components		Use this to…
Conditional	*wait-improved*	prevent calls from being queued remotely when the reduction in EWT isn't large enough to justify the cost and trunk usage. This conditional is for special applications and not used in all BSR vectors.
User adjustment	*adjust-by*	control long-distance costs and limit trunk usage

Multi-site BSR Applications

You can implement BSR at a single location just by using the new BSR commands in vectors. Using BSR across a network is more complex and requires additional administration.

Since a series of *consider location* steps in a multi-site BSR vector will contact one or more remote locations, you need to define these locations, tell DEFINITY ECS how to contact each one, and set up VDNs and vectors to handle communications between the origin switch and the remote (or receiving) switches. The BSR application should support some larger application in your call center that handles calls of a particular type.

NOTE:
Any mixture of split or skill numbers, VDN numbers, and vector numbers can be used to support a single customer application or call type across a network. But Lucent recommends that the BSR Application Plan number and the location numbers for a given application be the same on all switches for clarity and simplicity.

You also need to set up ISDN trunk groups, set the parameters for information forwarding (UUI Transport), and administer numbering plans and AAR/ARS tables.

Multi-site BSR starts with the active VDN for a call, as determined by VDN override. If you want any specific VDN/vector pair to interflow calls via multi-site BSR, you create a specific *application* for it. A multi-site application must contain the following elements:

Table 8-9. The Necessary Elements of Multi-site BSR Applications

A BSR application consists of…	Which serves this purpose…
the Primary VDN	The Primary VDN is the active VDN for a call at the origin switch, as defined by VDN override. Therefore, the Primary VDN in a BSR application need not be the VDN that originally received the incoming call. The primary VDN links its assigned vector to a BSR application plan and sets the BSR Available Agent Strategy.
the Primary vector that handles the incoming call on the origin switch	The Primary vector contacts the specified remote switches, collects information, compares the information, and delivers or queues the call to the resource that is likely to provide the best service.
an application plan	The application plan identifies the remote switches you may compare and specifies the information that will be used to contact each switch and to route calls to it.
two VDN/vector pairs on each remote switch	**Status poll VDN/vector** The status poll vector compares splits at its location and replies to the origin switch with information on the best of these splits. Each remote switch in a given application has to have a dedicated status poll VDN/vector.
	Interflow VDN/vector When a given remote switch is the best available, the origin switch interflows the call to this VDN/vector on the remote switch. Each remote switch in a given application has to have a dedicated interflow VDN/ vector. The steps in this vector deliver or queue the call, as appropriate, to the best resource found by the status poll vector.

To create a multi-site BSR application, you start by creating an application plan on the origin switch.

NOTE:
Remember that the terms “local,” “origin,” and “remote” are relative terms. In most networks using multi-site BSR, every switch may interflow calls to other switches and receive interflowed calls from other switches. Therefore, every switch in the network may have all the elements described above. For clarity in the following discussions, “local” or “origin” simply means a

switch that is considering or might consider whether to interflow a call. “Remote” means any switch that is polled or might be polled by this first switch.

Application plans

The application plan identifies the remote switches you may compare and specifies the information that will be used to contact each switch and to route calls to it.

The plan for each application is identified by a number (the *application number) and* a name. It specifies the remote switches that might be polled by the application and identifies each with a number called the *location number*. The plan also specifies the numbers for the status poll and interflow VDNs for each remote switch. Whatever you would dial to reach these VDNs is what should be entered in these fields: full length numbers as well as AAR, ARS, UDP, or public network numbers will work.

You create application plans on the Best Service Routing Application form. A plan for an application with three remote switches might look like Screen 8-5.

```
add best-service-routing                                    15 Page 1 of 16

BEST SERVICE ROUTING APPLICATION PLAN

 Number: 15     Name: Customer Service   Maximum Suppression Time: 60    Lock? y

Num  Location Name        Switch Node    Status Poll VDN     Interflow VDN

1    New Jersey          320            84015               84115
2    Denver              18             913031234015        913031234115
4    New York            12345          912121234015        912121234115
___ ____________________  _____  ____________  ____________
___ ____________________  _____  ____________  ____________
___ ____________________  _____  ____________  ____________
___ ____________________  _____  ____________  ____________
___ ____________________  _____  ____________  ____________
___ ____________________  _____  ____________  ____________
___ ____________________  _____  ____________  ____________
___ ____________________  _____  ____________  ____________
___ ____________________  _____  ____________  ____________
___ ____________________  _____  ____________  ____________
___ ____________________  _____  ____________  ____________
___ ____________________  _____  ____________  ____________
___ ____________________  _____  ____________  ____________
___ ____________________  _____  ____________  ____________
___ ____________________  _____  ____________  ____________
___ ____________________  _____  ____________  ____________
___ ____________________  _____  ____________  ____________
___ ____________________  _____  ____________  ____________
```

Screen 8-5. A Sample Multi-site BSR Application Plan

A single DEFINITY ECS can have 1–255 application plans. Each application plan can contain 1–255 locations. The limit on the number and size of BSR multi-site

applications is lower than these numbers suggest, however: one switch can have no more than 1,000 application-location pairs. For example, on a single switch you can't have more than 100 applications with 10 locations each, or 50 applications with 20 locations each, and so on.

By entering the application number from this plan on a VDN form, you can link a given VDN on your local switch to this list of locations. This VDN becomes the primary VDN for the application. Then, for example, if the primary vector contains instructions to consider locations 1 and 2, DEFINITY ECS knows that the instructions mean to place a status poll call to the status poll VDN at the New Jersey and Denver switches and compare the results. If, say, location 2 is better than either location 1 or any splits considered on the originating switch, the call will be interflowed to the interflow VDN specified in the plan for location 2.

Example 3: Multi-Site BSR with Two Switches

BSR Available Agent Strategy	UCD-MIA
BSR Commands & Command Elements	consider location, consider split, queue-to best, reply-best
Locations Considered	2
User Adjustments?	No

To see how the basic elements of multi-site BSR work, let's look at a simple application in a two-switch network. Multi-site BSR compares local and remote splits or skills and queues calls to the resource that provides the best service. Remember that each BSR application has two main parts:

- An application plan. This plan identifies the remote switches you want to check in the comparison. For this example, we'll use the application plan in Screen 8-5.
- A set of three VDN/vector pairs:

The *primary VDN/vector*	This vector on the origin switch contacts the specified remote switches, collects information, compares the information, and routes the call to the switch that is likely to provide the best service.
The *status poll VDN/vector*	The status poll vector on the remote switch compares resources on that switch and replies to the origin switch with information on the best of these. Each remote switch in a given application has to have a dedicated status poll vector.
The *interflow VDN/vector*	When a given remote switch is the best available, the origin switch interflows the call to this vector on the remote switch. Each remote switch in a given application has to have a dedicated interflow vector.

Screen 8-6 shows the primary VDN for this application.

```
                                                  Page  1 of   1
                     VECTOR DIRECTORY NUMBER

                       Extension: 52222
                            Name: Multi-site BSR
              Allow VDN Override? n
                             COR: 59
                              TN: 1
                    Vector Number: 222
                       AUDIX Name:
           Messaging Server Name:
                        Measured: none
  Acceptable Service Level (sec):
   VDN of Origin Annc. Extension: 201




               Return Destination:
          VDN Timed ACW Interval:

                  BSR Application: 15
    BSR Available Agent Strategy: UCD-MIA
```

Screen 8-6. BSR Example 3: Primary VDN

The VDN form above shows VDN 52222, the VDN we'll use in this example. The entry in the BSR `Application` field links this VDN to BSR Application Plan 15. Also note the **UCD-MIA** entry in the BSR `Available Agent Strategy` field: if vector 222 uses BSR commands, calls will not automatically be delivered to the first resource found with an available agent. All consider steps in vector 222 will be executed, and one of the following things will happen:

If …	Then…
there is no available agent in the local or the remote splits	the call will be queued to the split with the lowest adjusted EWT.
only one split has an available agent	the call will be delivered to that split.
two or more splits have available agents	the call will be delivered to the split with the most idle agent.

Also note that `Allow VDN Override?` is set to **n**. If a second VDN and vector are used to process this call, the, the UCD-MIA strategy and the application plan specified in VDN 52222 will be used.

Application plan 15 on the origin switch identifies the remote switch and provides the digit strings to dial into the VDNs for both the status poll vector and the interflow vector. (See Screen 8-5 on page 8-28 to see what plan 15 looks like.)

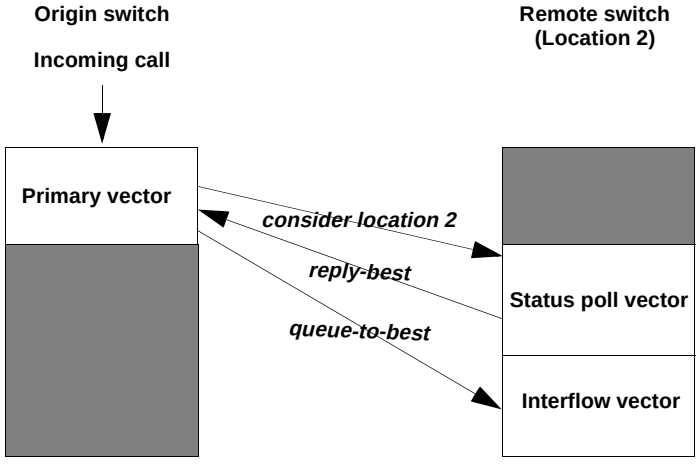


Figure 8-1. BSR Example 3: Origin and Remote Switches

Primary Vector

When a call arrives at the origin switch, it's processed by the *primary vector*. This vector begins the BSR process by considering resources you've specified. It might look like this:

```
1.   wait time 0 secs hearing ringback
2.   consider split 1 pri m   adjust-by 0
3.   consider location 2      adjust-by 30
4.   queue-to-best
```

Screen 8-7. BSR Example 3: Primary Vector on Origin Switch

In this example, the *consider* commands in steps 2 and 3 collect information to compare local split 1 with one or more splits at location 2. (Location 2 is the Denver switch identified on the BSR Application Plan form.) Step 4 queues the call to the best split found. As in single-site BSR, the *adjust-by* portion of the consider command allows you to set preferences for each resource—whether it's a remote location or a split or skill on the origin switch. In multi-site BSR, this user adjustment enables you to control the frequency of interflows by adjusting the EWT that's returned by a particular resource on a remote switch. In this example, the switch administrator has chosen to adjust the EWT value for location 2 by 30.

Status Poll Vector

To collect information from the remote switch, the command *consider location 2 adjust-by 30* in the primary vector places an ISDN call—a *status poll*—to the *status poll vector* on the switch at location 2. The status poll vector on the remote switch might look like this:

```
1.   consider split 2    pri m     adjust-by 0
2.   consider split 11   pri m     adjust-by 0
3.   reply-best
```

Screen 8-8. BSR Example 3: Status Poll Vector on Remote Switch

The status poll only obtains information and returns it to the origin switch; the call is not connected to the status poll VDN.

This vector compares splits 2 and 11, identifies the better of the two, and sends this information back to switch 1 with the *reply-best* command. Notice that the *adjust-by* command could be used on the remote switch to adjust the EWT returned by either of the splits. When EWT adjustments are applied at both the origin and remote switches, the two adjustments are added at the origin switch. See "User Adjustments in Multi-site BSR" on page 8-35 for more detail on user adjustments in multi-site applications.

The *consider* command is ISDN-neutral and does not return answer supervision. The status poll call is dropped when the *reply-best* step executes, but the ISDN DISCONNECT message returned to switch 1 contains the information from the best split considered at location 2. Once the remote switch has returned the necessary information, the consider series in the primary vector on switch 1 can continue at the next vector step.

CAUTION:

Lucent recommends that status poll vectors not be used to poll other switches. Status poll vectors should only consider resources on the switch where the vector resides.

NOTE:

Multi-site BSR includes mechanisms that automatically limit the number of status poll calls placed over the network when such calls are unlikely to yield better service for the caller. For a detailed explanation of these mechanisms, see Appendix D, "Advanced Multi-Site Routing".

Interflow Vector

In this example, let's suppose that no agents are available and that split 11 (location 2) has the lowest adjusted EWT. The *queue-to best* command in the primary vector will interflow the call to the *interflow vector* at location 2. The interflow vector looks like this:

```
1.   consider split 2    pri m     adjust-by 0
2.   consider split 11   pri m     adjust-by 0
3.   queue-to best
```

Screen 8-9. BSR Example 3: Interflow Vector on Remote Switch

The interflow vector reconsiders the status of both splits to get the most current information and queues or delivers the call to the best split. Notice that the consider sequences in the interflow vector and the status poll vector are identical aside from their last step. As happens today when a call is interflowed, it is removed from any queues at the origin switch and any audible feedback at the origin switch is terminated.

CAUTION:

Best Service Routing will not operate correctly unless the consider series in the status poll vector and the interflow vector use the same splits or skills with the same queue priorities.

Questions

What if the interflow attempt fails? What happens to the call?

If the interflow attempt fails (for example, because there are no available trunks), the call will be queued to the best local split. The call will not be disconnected. The call will not be dropped from vector processing on the origin switch. For the call to be queued to a local split, however, that split must have been the "best" resource at some previous point in the consider series. In writing primary vectors, always consider local splits or skills before considering remote resources.

I can adjust the EWT returned by a split or skill when no agents are available. When agents are available in two or more splits or skills, can I adjust Agent Idle Time (AIT) returned by a resource? Can I adjust the agent skill level returned by a resource?

No. EWT for a resource is the only data BSR lets you adjust.

BSR Available Agent Strategies

In multi-site BSR applications, the 1st-found available agent strategy results in fewer interflows and thus minimizes the load on inter-switch trunking. DEFINITY ECS also has less processing to perform for each call in BSR vectors, since it may not need to compare as many resources to identify the best. If processing power and tie trunk capacity are issues in your multi-site applications, you may want to use the 1st-found strategy.

The other strategies typically result in a much greater percentage of calls being interflowed, thus optimizing load balancing across locations. For a strategy that greatly increases agent fairness across the network while limiting the number of trunks used, see "Example 4: Multi-site BSR with Limited Trunking" on page 8-36.

More on Status Poll and Interflow Vectors

- Since status poll vectors don't return answer supervision, call charges are not normally incurred for the status poll portion of the call flow.
- When a *consider location* step performs a status poll, it also checks for the availability of a B-channel. If no B-channel is available, the remote resource will never be considered the best since the call cannot be redirected to it.
- If only one split or skill on a remote switch can service the call type handled in a BSR application, you need not write a consider series in the interflow vector: simply queue the call to the appropriate resource.
- If status poll and interflow vectors consider more than one split or skill, the VDNs for these vectors must be administered with the appropriate BSR Available Agent Strategy.

User Adjustments in Multi-site BSR

User adjustments are especially important in multi-site applications, where unnecessary interflows may be costly and use precious trunk capacity inefficiently.

User adjustments in multi-site applications function in the same way they do in single-site BSR, with one important difference: user adjustments may be applied at the remote switches in an application as well as at the origin switch. Since a status poll vector uses *consider* steps to evaluate resources on the switch where it resides, the *adjust-by* portion of each *consider* command allows the administrator at each switch to set preferences for the splits or skills at that switch. In BSR applications, any such adjustment for a split or skill is considered by the status poll vector in selecting the best resource on its switch. The adjustment is then returned to the origin switch along with the other data for that resource. When DEFINITY ECS receives this adjustment from the remote switch, it adds it to any adjustment that was assigned to that location in the *consider location* step. Let's look at an example. The following example assumes, of course, that no agents become available during the time these vectors are processing the call.

The primary vector below considers one remote location, to which it assigns an adjustment of 30.

```
1.   wait time 0 secs hearing ringback
2.   consider split 1 pri m   adjust-by  0
3.   consider location 2      adjust-by 30
4.   queue-to-best
```

Suppose the status poll vector at location 2 looks like this:

```
1.   consider split 2    pri m     adjust-by 0
2.   consider split 11   pri m     adjust-by 20
3.   reply-best
```

Consider split/skill commands in status poll vectors work just like they do in single-site BSR vectors. The user adjustments are applied to a single split or skill—not to the entire location. In this case, the two splits are assigned different adjustments. Let's say that split 11, despite having the larger adjustment, returns the lower adjusted EWT for a call. The *reply-best* command in step 3 returns the user adjustment of 20 to the primary vector on the origin switch, along with the rest of the data for split 11.

In saving the data returned by location 2, the origin switch will add the remote adjustment of 20 to the adjustment of 30 specified in step 3 of the primary vector. As a result, the call will not interflow to location 2 in this example unless the EWT for location 2 is more than 50 seconds better than the EWT in split 1 on the origin switch.

Example 4: Multi-site BSR with Limited Trunking

BSR Available Agent Strategy	EAD-MIA
BSR Commands & Command Elements	consider location, consider skill, queue-to best, reply-best, goto
Locations Considered	5
User Adjustments?	Yes

Multi-site BSR applications must balance improvements in wait times and agent utilization with the cost of interflows and the availability of inter-switch trunking for status polls and interflows. This example illustrates an application recommended for balancing agent workload across the network while still limiting tie-trunk usage.

Screen 8-10 shows the application plan.

```
add best-service-routing                                    15 Page 1 of 16

BEST SERVICE ROUTING APPLICATION PLAN

 Number: 10     Name: International    Maximum Suppression Time: 60    Lock? y

Num  Location Name       Switch Node    Status Poll VDN     Interflow VDN

1    Kansas City         1111           919131234015        919131234115
2    New York            1112           912121234015        912121234115
3    Montreal            1113           915141234015        915141234115
4    London              1114           90114411234015      90114411234115
___ ___________________  ____           ____________        ____________
___ ___________________  ____           ____________        ____________
___ ___________________  ____           ____________        ____________
___ ___________________  ____           ____________        ____________
___ ___________________  ____           ____________        ____________
```

Screen 8-10. BSR Example 4: Application Plan

Screen 8-11 shows the VDN form for VDN 51110, the VDN we'll use in this example. The entry in the BSR `Application` field links this VDN to BSR Application Plan 10. Also note the **EAD-MIA** entry in the BSR `Available Agent Strategy` field: if vector 100 uses BSR commands, calls will not automatically be delivered to the first resource found with an available agent. In

each consider sequence, when the *queue-to best* or *check best* step executes, one of the following things will happen:

If …	Then…
no skill has an available agent	the call will be queued to the skill with the lowest adjusted EWT.
only one skill has an available agent	the call will be delivered to that skill.
two or more skills have available agents	the call will be delivered to the skill with the most expert agent (lowest skill level).
two or more skills have available agents with the same skill level	the call will be delivered to whichever of these skills has the most idle agent.

Also note that `Allow VDN Override?` is set to **n**. If a second VDN and vector are used to process this call, the, the EAD-MIA strategy and the application plan specified for VDN 51110 will still be used.

```
                                                    Page  1 of   1
                     VECTOR DIRECTORY NUMBER

                       Extension: 51110
                            Name: Multi-site BSR
              Allow VDN Override? n
                             COR: 59
                              TN: 1
                    Vector Number: 100
                       AUDIX Name:
           Messaging Server Name:
                        Measured: none
  Acceptable Service Level (sec):
   VDN of Origin Annc. Extension: 1001
                       1st Skill:
                       2nd Skill:
                       3rd Skill:

               Return Destination:
          VDN Timed ACW Interval:

                  BSR Application: 10
    BSR Available Agent Strategy: EAD-MIA
```

Screen 8-11. BSR Example 4: Primary VDN

With four remote switches to be considered, the overall application is represented in Figure 8-2. Application plan 10 on the origin switch identifies the remote switches and provides the digit strings to dial into the VDNs for both the status poll vector and the interflow vector on each switch.

Each *consider location* command in the primary vector will place a status poll call to its specified location. The status poll vector at that location will execute a series of *consider skill* commands and return data on the best resource to the origin switch through a *reply-best* command.

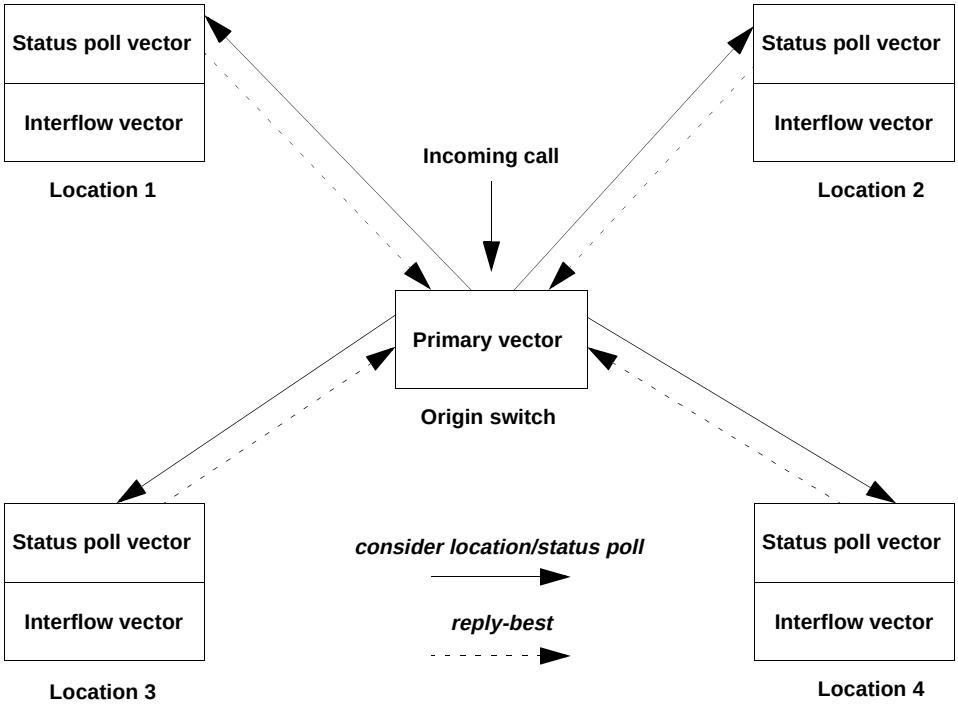


Figure 8-2. BSR Example 4: Multi-site Application with 4 Switches and Limited Tie Trunk Capacity

Let's look at the primary vector for this application, vector 100, shown in Screen 8-12. The first consider series in the primary vector tests two local skills. If either skill has an available agent, step 4 jumps to step 9 and the call is queued locally. No remote locations are polled. If no agents are available in either local skill, though, steps 5–8 test 4 remote locations.[3] If the best remote location's adjusted EWT can reduce the call's current adjusted EWT, step 9 interflows the call to that

3. In general, you shouldn't put other commands between *consider* steps. This use of the *goto* step is one of the few exceptions to that rule.

location. In this vector, a local available agent is always favored over a remote available agent. Whichever location services a call, it will always be directed to the most idle, best skilled agent available.

```
1.   wait time 0 secs hearing ringback
2.   consider skill  1 pri m  adjust-by  0
3.   consider skill  2 pri m  adjust-by 20
4.   goto step 9 if expected-wait for skill best = 0
5.   consider location 1      adjust-by 30
6.   consider location 2      adjust-by 30
7.   consider location 3      adjust-by 50
8.   consider location 4      adjust-by 50
9.   queue-to best
10.  announcement 1001
11.  wait time 60 secs hearing music
12.  goto step 10 if unconditionally
```

Screen 8-12. BSR Example 4: Primary Vector

In the primary vector, note that user adjustments have been entered for local skill 2 as well as for all the remote locations. These indicate the administrator's preferences regarding both local and remote resources. For this example, let's say that neither local resource has an available agent (and therefore an EWT>0).

Status Poll Vector

Each receiving switch in a multi-site application must have a status poll vector. To collect information from these locations, each *consider location* command in the primary vector places a status poll to the status poll vector for the appropriate switch. We'll look at only one of these vectors—the status poll vector on the switch at location 3, which might look like this:

```
1.   consider skill 2    pri m     adjust-by 0
2.   consider skill 11   pri m     adjust-by 20
3.   consider skill 21   pri m     adjust-by 30
4.   reply-best
```

Screen 8-13. BSR Example 4: Status Poll Vector at Location 3

This vector compares skills 2, 11, and 21, identifies the best one, and sends this information back to the origin switch through the *reply-best* command. Notice that user adjustments are applied to skills 11 and 21 to adjust the skill's EWT.

When EWT adjustments are applied at both the origin and remote switches, the two adjustments are added at the origin switch. See "User Adjustments in Multi-site BSR" on page 8-35 for more detail on user adjustments in multi-site applications.

In this example, let's suppose that skill 11 has the best adjusted EWT at location 3. Its data, including a user adjustment of 20, is returned to the origin switch by the *reply-best* command.

Finding the Best Resource

Once the remote switches have returned the best data for each location, the second consider series in the primary vector can be completed. In this example, let's suppose that no agents are available at any remote location. Table 8-10 shows how user adjustments at the origin and remote switches yield the adjusted EWT for each location.

Table 8-10. BSR Example 5: User Adjustments

Location	Actual EWT of remote best (sec.)	User adjustment on origin switch	User adjustment on remote switch	Adjustment applied by origin switch (sec.)	Adjusted EWT used in BSR calculations (sec.)
1	60	30	0	30	90
2	45	30	10	40	85
3	40	50	20	70	110
4	70	50	0	50	120

The second consider series will identify location 2 as the best remote location, with an adjusted EWT of 85, and the *queue-to best* step will interflow this call to location 2.

Interflow Vector

The interflow vector on a remote switch in a multi-site application accepts the interflowed call from the origin switch. It also executes the same consider series as the status poll vector to identify the current best resource, in case conditions have changed since the status poll.

```
1.   consider skill 2    pri m     adjust-by 0
2.   consider skill 11   pri m     adjust-by 20
3.   consider skill 21   pri m     adjust-by 30
4.   queue-to best
```

Screen 8-14. BSR Example 4: Interflow Vector at Location 2

As happens today when a call is interflowed, it is removed from any queues at the origin switch and any audible feedback at the origin switch is terminated.

CAUTION:

Best Service Routing will not operate correctly unless the consider series in the status poll vector and the interflow vector use the same splits or skills with the same queue priorities.

Example 5: Multi-site BSR with Slow Networks

BSR Available Agent Strategy	EAD-MIA
BSR Commands & Command Elements	consider location, consider skill, queue-to best, reply-best, check best, wait-improved
Locations Considered	5
User Adjustments?	Yes

Network response times will not be an issue for most users. This example is intended for those users, if any, who experience such a problem. This example uses the same VDN, application plan, and 4-switch network described in example 4.

The vector in example 4 minimized interflows by using a *goto* step that skips the remote consider series if a local resource has an available agent. Example 5 shows a design that's especially useful if network response times are slow. Calls are always queued once locally before remote locations are considered. Furthermore, both status polls and interflows are conditional. The call can wait in queue for a local resource while BSR looks for a better split or skill at remote locations.

This example also illustrates the function of the *check best* command and the *wait-improved* conditional.

Let's look at the primary vector for this application, vector 100, shown in Screen 8-15. The first consider series in the primary vector tests two local splits and queues the call to the best one. If the EWT for the best split is 30 seconds or less, step 5 jumps to the loop in step 11 and the second consider series isn't executed. If the EWT for the best split is over 30 seconds, though, steps 6–9 test 4 remote locations. If the best remote location can reduce the call's EWT by more than 30 seconds, as compared to its EWT in the best local queue, step 10 interflows the call to that location.

CAUTION:

Be certain to queue calls at least once before using the wait-improved conditional in a vector step. If calls are not already queued when the step with the wait-improved conditional executes, DEFINITY ECS will read the call's EWT as infinite. This could result in a vector that interflows all calls, even if that is not its intended function.

```
1.   wait time 0 secs hearing ringback
2.   consider skill  1 pri m  adjust-by  0
3.   consider skill  2 pri m  adjust-by 20
4.   queue-to-best
5.   goto step 11 if expected-wait for call <= 30
6.   consider location 1      adjust-by 30
7.   consider location 2      adjust-by 30
8.   consider location 3      adjust-by 50
9.   consider location 4      adjust-by 50
10.  check best if wait-improved > 30
11.  announcement 1001
12.  wait time 60 secs hearing music
13.  goto step 11 if unconditionally
```

Screen 8-15. BSR Example 5: Primary Vector

A consider series can end with either a *queue-to best* or a *check best* step. All consider series in the other examples have used a *queue-to best* command to queue the call unconditionally. The *check best* command lets you set conditions that must be met before a call is queued to the best resource. In this example, step 10 in the primary vector is `check best if wait-improved > 30.` In other words, step 10 will only interflow the call to the best location found by the consider series if the EWT for that location is more than 30 seconds better than the call's EWT in the local queue.

You can use up to 3 consider series in one vector.[4] It's also possible to combine single- and multi-site consider series, as this example illustrates. Note that user adjustments have been entered for local skill 2 as well as for locations 3 and 4.

These indicate the administrator's preferences regarding both local and remote resources. In this example, let's say that step 2 queues the call to skill 1, which has an EWT of 65 seconds, before the second consider series is executed.

Status Poll Vector

Each receiving switch in a multi-site application must have a status poll vector. To collect information from these locations, each *consider location* command in the primary vector places a status poll to the status poll vector for the appropriate switch. We'll look at one of these vectors—the status poll vector on the switch at location 3, which is also taken from example 4:

```
1.   consider skill 2    pri m     adjust-by 0
2.   consider skill 11   pri m     adjust-by 20
3.   consider skill 21   pri m     adjust-by 30
4.   reply-best
```

Screen 8-16. BSR Example 5: Status Poll Vector at Location 3

This vector compares skills 2, 11, and 21, identifies the best one, and sends this information back to the origin switch through the *reply-best* command. Notice that user adjustments are applied to skills 11 and 21 to adjust the skill's EWT. When EWT adjustments are applied at both the origin and remote switches, the two adjustments are added at the origin switch. See "User Adjustments in Multi-site BSR" on page 8-35 for more detail on user adjustments in multi-site applications.

In this example, let's suppose that skill 11 has the best adjusted EWT at location 3. Its data, including a user adjustment of 20, is returned to the origin switch by the *reply-best* command.

Remember that the first consider series queued the call to local skill 1. Let's say that the second consider series identifies location 2 as the best remote resource. The *check* command in step 10 recalculates the call's current, unadjusted EWT in skill 1 and compares it to location 2's unadjusted EWT. If the call's actual (unadjusted) EWT can be improved by more than 30 seconds, the call is interflowed.

---

4. It's possible to write more than 3 consider series in a vector, but there's no benefit in doing so. DEFINITY ECS only allows you to queue a call simultaneously to 3 different local resources. Since each consider series ends by queuing a call (assuming no agent is available), using more than 3 series in a vector will not place the calls in additional local queues. If the call interflows to another switch, it's removed from vector processing and any queues it was in on the origin switch.

NOTE:
BSR uses adjusted EWT to determine which of the resources in a consider series is the best. Once the best resource has been identified, subsequent *expected-wait* and *wait-improved* conditionals use the actual EWT values.

Interflow Vector

When a call is interflowed to any of the remote locations, the interflow vector on that switch accepts the interflowed call from the origin switch. It also executes the same consider series as the status poll vector to identify the current best resource, in case conditions have changed since the status poll.

```
1.   consider skill 2    pri m     adjust-by 0
2.   consider skill 11   pri m     adjust-by 20
3.   consider skill 21   pri m     adjust-by 30
4.   queue-to best
```

Screen 8-17. BSR Example 5: Interflow Vector at Location 2

CAUTION:
Best Service Routing will not operate correctly unless the consider series in the status poll vector and the interflow vector use the same splits or skills with the same queue priorities.

Questions

If the call is queued to a remote resource by step 10 in the primary vector, is the call removed from the local queue it entered in step 4?

As happens today when a call is interflowed, the call is removed from any queues at the origin switch and any audible feedback at the origin switch is terminated.

The second consider series could have compared local and remote resources. If it did, and if step 10 queued the call to another local skill, would the call be removed from the local queue it entered in step 4?

No. In general, DEFINITY ECS can queue a call to as many as 3 local splits or skills simultaneously. BSR doesn't change this limit.

Example 6: Handling Excessive Wait Times

This short example illustrates a simple primary vector in a multi-site BSR application. If wait times are sometimes excessive because of high call volumes, step 4 of this vector directs calls to a *disconnect after announcement* step when wait time in the network exceeds 5 minutes.

```
1.   wait 0
2.   consider skill 1    pri m     adjust-by 0
3.   consider location   2    pri m     adjust-by 30
4.   goto step 6 if expected-wait for best ≤ 300
5.   disconnect after announcement 3001
6.   queue-to best
```

Screen 8-18. BSR Example 6: Primary Vector

Announcement 3001 might say something like “We’re sorry: we are currently experiencing heavy call volume and cannot service your call at this time. Please try again later. We are normally least busy between 8 a.m. and 11 a.m. each morning.”

Simple Troubleshooting for Multi-Site BSR

You should regularly execute a **display events** command for the appropriate vectors—especially if you’ve just implemented a new BSR application. Vector events will identify and indicate the source of common malfunctions and administration errors.

When tie-trunks or queue slots become exhausted, BSR cannot effectively balance calls across the network. If such problems are revealed frequently by vector events, review the design of the BSR application involved. If tie-trunks are frequently exhausted, the user adjustments on *consider location* steps may be set too low.

For a list of BSR vector events and definitions, see “Tracking Unexpected Vector Events” on page E-16.

NOTE:
Only the most recent events are displayed when a **display events** command is executed. Periodic display of vector events will help you quickly identify problems.

To verify that your BSR vectors are operating as intended, use a **list trace vdn** or **list trace vec** command to observe processing of an individual call. Refer to “Tracing Call Flow” on page E-32 for more information.

Planning and Administering Single-Site BSR

This section only presents information specific to BSR. Follow existing procedures to add or change other properties of VDNs and vectors not discussed in this section.

First, confirm that your switch meets the requirements for single-site BSR if you haven't already done so. See "Before You Start" on page 8-5 for a listing of requirements.

Planning

To work more efficiently, you may want to record goals, VDN extensions, vector numbers, and other information on paper before you begin your administration session.

1. Select the group of callers for which you want to use single-site BSR, and identify the VDNs and vectors that support this group.
2. Define your goals (for example, faster average speed of answer, or better service by routing calls to the most qualified agents).

   Different VDNs or vectors may have different goals.
3. Decide which agent selection strategy you'll assign to each VDN in order to best achieve the goals relevant to that VDN.
4. Decide whether you'll allow VDN Override for each of the VDNs you've identified.

Administration

These instructions assume you're using the SAT screen or a terminal emulator to access the DEFINITY software.

5. To go to the Vector Directory Number form for the first VDN you identified in step 1, type **add vdn** ##### or **change vdn** ##### at the command line prompt and press RETURN. (In place of #####, type a valid 1–5 digit VDN extension as defined in your system's dial plan.)

   If this is a new application, create the VDN.
6. In the `Allow VDN Override?` field, type **y** or **n**. If the call is directed to another VDN during vector processing:

   — **y** allows the settings on the subsequent VDN—including its BSR Available Agent Strategy—to replace the settings on this VDN.

   — **n** allows the settings on this VDN—including its BSR Available Agent Strategy—to replace, or override, the settings on the subsequent VDN.
7. In the `BSR Available Agent Strategy` field, type the identifier for the agent selection method you want this VDN to use.

When this VDN is the active VDN for a vector that uses BSR, the available agent strategy determines how calls are directed when one or more of the specified resources have available agents. If there is only one split or skill with available agents, calls will be delivered to that resource.

If you enter...	Consider series in vectors will select the resource with...
1st-found	the first available agent. BSR will not consider any other resources as soon as it finds an available agent.
UCD-MIA	the agent who has been idle the longest. BSR will compare all the splits or skills specified in the vector before delivering the call.
EAD-MIA	the agent with the highest skill level who has been idle the longest. BSR will compare all the splits or skills specified in the vector before delivering the call.
UCD-LOA	the least-occupied agent.[1] BSR will compare all the splits or skills specified in the vector before delivering the call.
EAD-LOA	the agent with the highest skill level who is the least occupied. BSR will compare all the splits or skills specified in the vector before delivering the call.

1. LOA is a new agent selection method available as an option on DEFINITY ECS R6.3 and later switches. For more information, please see the *CentreVu® Advocate User Guide* (585-215-855).

8. Press ENTER to save your changes.

You're now ready to write or modify the vector assigned to this VDN. For tips on using BSR commands in vectors, see "Tips for Writing BSR Vectors" on page 8-52.

Planning and Administering Multi-Site BSR

This section only presents information specific to BSR. Follow existing procedures to add or change other properties of VDNs and vectors not discussed in this section.

To create multi-site applications, follow the process below. Record location numbers, Status Poll VDNs, and similar information on paper. If you have all this information in one place, you'll be able to work more efficiently when you begin to create the application on the switch.

Define the Purpose of the Application

1. Select the group of callers for which you want to create the application.
2. Define the goal of the application (for example, faster average speed of answer, better service by routing calls to the most qualified agents).
3. Decide which agent selection strategy (on VDNs) will best achieve your goal.
4. Decide whether you'll implement BSR in a distributed or a centralized system.
   - In a distributed system, all switches receive incoming calls and query other switches to interflow calls when appropriate.
   - In a centralized system, one switch serves as a hub: all incoming calls arrive at this switch and are routed from it to the other switches in the network.

   Since a distributed system is the more complicated of the two, the rest of this procedure is written in terms of implementing a distributed system. The same steps apply to implementing a centralized system, but only one switch will have application plans and primary VDNs/vectors.

Select or Create the Elements of the Application Plan

5. Select the VDNs on each switch that serve the group of callers you've identified.

   On each switch these are the Primary VDNs for your application. You may, of course, want or need to create new VDNs. In either case, record the extensions of each VDN that will point to a vector with a BSR application.
6. Select the locations you want to include in each application plan. To uniquely identify each location, assign a number between 1–255 and a short name of 15 characters or less.
7. Record the node number of the switch at each location.
8. Create Status Poll VDNs on each of the switches in the application plan.

   Record the full numbers you'll need to route calls to these VDNs. You'll enter these numbers on the Best Service Routing Application Plan form when you create the plan.

If you're creating new VDNs on the switches to receive interflowed calls, record these numbers too. You'll need them to complete the BSR Application Plan form. Remember: you can't use the same number for a Status Poll VDN and an Interflow VDN.

Administer the Application on the Switch

These instructions assume you're using the SAT screen or a terminal emulator to access the DEFINITY software.

Define the Application Plan

Follow these steps to create an application plan on each switch.

9. At the command line prompt, type **add best-service-routing** ### and press ENTER. (In place of ###, type the number between 1–255 that you want to assign to this BSR application.)

   The Best Service Routing Application Plan form appears. The number you typed in the command appears in the Application Number field.

10. Assign a name to the plan.

    The best names are short and descriptive. This name can't be longer than 15 characters.

11. Type in the information for the first remote location. Fill in the information for each field as shown below.

    NOTE:
    Each row on the form contains all the information the BSR application needs to identify and communicate with one of the resources in the plan

Table 8-11. Fields on Application Plan form

Field	Type	Description
Num	Required	Type the number you assigned to this location in Step 6.
Location Name	Optional	Type the name you assigned to this location in Step 6.

Continued on next page

Table 8-11. Fields on Application Plan form — *Continued*

Field	Type	Description
Switch Node	Optional	This field is for user reference only and it won't hurt anything to leave it blank. If you're using DEFINITY's Universal Call ID feature, you may want to type each switch's node identity in this field. The switch's node identity is the number entered in the UCID Network Node ID field on page 4 of the Feature-Related System Parameters form.
Status Poll VDN	Required	This is the complete digit string your switch will dial for the status poll call. The string can be up to 16 digits long.
Interflow VDN	Required	This is the complete digit string your switch will dial to interflow a call to this location. The string can be up to 16 digits long.

12. Repeat Step 11 for each of the locations you want to include in the application plan.
13. Press ENTER to save your changes.

NOTE:
You must set up trunk groups to other sites. Refer to Chapter 7, "Look-Ahead Interflow" and Chapter 9, "Information Forwarding" for information on setting up trunk groups.

Link the Application Plan to a Primary VDN and Enter an Agent Selection Strategy

14. Go to the Vector Directory Number form for the first VDN you identified in Step 5.

    If this is a new application, create the VDN.
15. In the `Allow VDN Override?` field, type **y** or **n**. If the call is directed to another VDN during vector processing:
    - **y** allows the settings on the subsequent VDN—including its BSR Available Agent Strategy—to replace the settings on this VDN.
    - **n** does not allow the settings on the subsequent VDN—including its BSR Available Agent Strategy—to replace, or override, the settings on this VDN.

16. In the BSR Application field, type the application number you assigned to the plan.
17. In the BSR Available Agent Strategy field, type the identifier for the agent selection method you want this application to use:

If you enter...	The application will select the resource with...
1st-found	the lowest Expected Wait Time. If the application finds an available agent before it has compared all the locations in the plan, the application will route the call to that agent without contacting any other locations.
ucd-mia	the agent who has been idle the longest. The application will compare all the locations in the plan.
ead-mia	the agent with the highest skill level (lowest number) who has been idle the longest
ucd-loa	the least-occupied agent
ead-loa	the agent with the highest skill level (lowest number) who is the least occupied

18. Press ENTER to save your changes.

Repeat Step 9–Step 18 on each switch that needs an application plan and a Primary VDN/vector pair.

This process covers the administration needed for BSR vector commands to function. Now, of course, you need to write or modify the vectors that will control call processing.

Tips for Writing BSR Vectors

Before you write your first vector using BSR, you should study the sample vectors we've provided and familiarize yourself with the new commands and command elements. Sample vectors are provided in the sections "Single-Site BSR" and "Multi-site BSR". The new commands and command elements are explained in Appendix A.

As you write BSR vectors, Lucent strongly recommends that you follow the guidelines below.

- Arrange your *consider* steps in order of preference.

  The *consider* step that tests the main, or preferred, resource should be the first in the series. The second *consider* step should test the resource that is your second preference for handling the given call type, and so on. To avoid unnecessary interflows, put *consider* steps for local resources before steps that consider remote resources.[5] Arranging *consider* steps in order of preference is recommended for all BSR vectors. It's especially important when the active VDN for the call is using the 1st-found agent strategy: since DEFINITY ECS will deliver the call to the first available agent found, arranging *consider* steps in order of preference will ensure that calls are delivered to the best of the available resources and that unnecessary interflows are avoided.

- Don't put any commands between the steps of a consider series that would cause a delay. *Goto* commands are OK.

- Don't put a consider series in vector loops.

- Confirm that calls queue successfully.

  This check is recommended for all vectors. Since EWT is infinite for a call that hasn't queued, a step that checks EWT after a queue attempt is a good confirmation method. After a *queue-to best* step, for example, a command such as *goto step X if expected-wait for call < 9999*

- Don't use the wait-improved conditional in a vector before you've queued the call at least once.

  The *wait-improved* conditional compares the call's EWT in its current queue to the best resource found by a consider series. If a call hasn't been queued and a vector step such as *check best if wait-improved > 30* is executed, DEFINITY ECS will interpret the call's current EWT as infinite and the *check best* step will always route the call to the best resource. In other words, in this situation the *check best* step functions like an unconditional *goto* or *route-to* command.

5. This arrangement also provides a local "best' as a backup in case the interflow fails.

Information Forwarding

9

Standard information forwarding involves the transport of the following incoming call-related information:

- ANI
- II-Digits
- CINFO
- ASAI-provided user information
- Look-Ahead Interflow (LAI) information (such as VDN name)

DEFINITY R6.3 supports this information forwarding, and also supports information forwarding for

- Universal Call ID (UCID)—Tags a call with a unique identifier used to track each call. For more information, see Universal Call ID in the *DEFINITY® ECS Release 6 Administration and Feature Description Guide* (555-230-522)
- Best Service Routing™ (BSR)—Allows DEFINITY® ECS to compare specified splits or skills, identify the split or skill that will provide the best service to a call, and deliver the call to that resource. For more information, see Chapter 8, "Best Service Routing™".
- New interflowed call data (collected digits, in-VDN time)

NOTE:
This transport takes place via globally-supported ISDN information transport methods over public and/or private networks using ISDN trunks (for example, PRI or BRI). Private networks may be configured for QSIG or non-QSIG protocols.

When a call is interflowed to an alternative switch by the Best Service Routing™ or Lookahead Interflow features, the following new data forwarding is supported:

- Collected Digits—Any digits collected for the call are passed with the interflowed call, and automatically collected when the call enters vector processing at the receiving switch.
- Elapsed in-VDN time—The elapsed time that the call has already spent at the sending switch is passed with the interflowed call and automatically sent to CMS when the call enters vector processing at the receiving switch.
- UCID—Universal Call ID.

Benefits of enhanced information forwarding

Enhanced information forwarding provides the following benefits:

- **Improved agent efficiency and service to call**—forwarding of original caller service requirements and entered prompted digits speeds service to the caller and saves the agent time.
- **Better network-wide call tracking**—forwarding of UCID, In-VDN-Time and collected digits allows tracking as a single call and provides a network-wide view for call statistics.
- **Better CTI integration**—forwarding of UCID, In-VDN-Time, and collected digits provides screen pop and database access applications across sites.
- **Improved global compatibility and viability**—use of codeset 0 supports information transport over ISDN PRI/BRI facilities (QSIG or non-QSIG) as well as supporting operation over public networks.

Table 9-1 outlines the benefits of each function of enhanced information forwarding.

Table 9-1. Benefits of Enhanced Information Forwarding

Function	Benefit
Forwarding of original call service requirements (VDN Name or DNIS)	Faster and more efficient agent handling, better service to caller, and improved CTI integration
Transport of UCID	Improved call tracking as a single call and CTI integration
Collected Digits Transport	Better service to caller (caller doesn't have to repeat input of information), more information for agent, better and faster call handling, improved call tracking (included with call record), and improved CTI integration
Forwarding of In-VDN Time	Improved call tracking as a single call and end-to-end time-before-answer statistics.
Continued support of ASAI user information forwarding	CTI integration
Globally-supported transport	Improved global compatibility
Operation over public networks	improved global compatibility and lower facility/call transport costs
Operation over QSIG trunks	Improved global compatibility and interoperability
Operation over BRI trunks	Lower facility/call transport costs

Network requirements

The network must meet the following conditions for correct information forwarding operation:

- The network (private or public) must support end-to-end transport of codeset 0 user data either as user-to-user information (UUI IE) or QSIG manufacturer specific information (MSI) in the SETUP and DISCONNECT ISDN messages. Private networks can be configured for either non-QSIG (transport via a codeset 0 UUI IE) or QSIG (transport via MSI packaged in a codeset 0 Facility IE). Currently, public networks do not support QSIG,

and user data can only be transported via the UUI IE when supported by the network. Future public network offerings may support QSIG (possibly via Virtual Private Network).

- DEFINITY must support the ISDN country protocol.
- The network byte limit for user information contents (user data portion) must be large enough to carry the data needed for the customer application.

NOTE:
Some public network providers may require service activation and/or fees for user information transport.

Enhanced information forwarding has been tested with several major carriers. To find out if these capabilities work with your carrier, check with your account team for the most current information.

If testing has not been done to verify operation over the public networks involved with the preferred specific configuration, use of private ISDN trunking between the nodes should be assumed until successful testing has been completed.

Enhanced Information Forwarding

DEFINITY R6.3 enhanced information forwarding involves the following capabilities:

- Forwarding of existing call related information: ASAI user data, VDN name, other LAI info (for example, the in-queue timestamp), and network-provided caller information.
- Forwarding of new call-related information: collected digits, UCID, and in-VDN time
- Transporting information via globally-supported methods
- Providing LAI backward compatibility

NOTE:
The switch version must be V6 or later.

For information about administering information transport, see *DEFINITY® ECS Release 6 Administration and Feature Description* (555-230-522). For detailed information about ISDN trunk group setting interactions with information forwarding, UCID, and multi-site routing, see "Advanced Information Forwarding" on page D-14.

Forwarding of call-related information

Forwarding of call-related information via the globally supported transport applies to both BSR and LAI. Depending on administration, information will be sent with an LAI or BSR interflowed call.

Forwarding collected digits with interflowed call

The following summarizes what happens to forwarded collected digits:

- The last set of collected digits (up to 16 digits—not the dial-ahead digits) are forwarded with a call interflowed over ISDN facilities.
- When processing for the call at the remote location reaches the VDN, the forwarded digits are inserted in the collected digits buffer (a TTR is not needed). The objective is to immediately provide the collected digits to CMS (in a DIGITS message) and to ASAI (via the VDN event report) in the same manner as incoming ANI.
- The collected digits are available for further routing by steps in the assigned and subsequent vectors, and eventual display to the answering agent.
- All interactions with the collected digits are the same as digits collected via a collect step (for example, a subsequent collect step will clear the digits).
- If the call is further interflowed or tandemed over ISDN facilities, the collected digits are tandemed with the call. If more digits are collected at the tandem switch, the latest collected digits are tandemed.

Forwarding accumulated in-VDN time

The following summarizes what happens to forwarded in-VDN time:

- When a call is interflowed, the in-VDN time in seconds [0 - 9999] is included. The in-VDN time is the elapsed time starting from the first reached VDN (originally called) until when the information forwarding message is created.
- If the call has been interflowed to the local system and in-VDN time was received for the call, the previous in-VDN time is added to the local in-VDN time.
- If the accumulated time exceeds the largest value that can be transported, the maximum value is sent.
- The accumulated in-VDN time (received on an incoming interflowed call) is forwarded to CentreVu CMS (via the DNEVENT message) when the call starts VDN/vector processing at the remote location.
- In-VDN time does not pass to BCMS for reporting by BCMS.

Transport via globally-supported methods

The following summarizes information transport via globally-supported methods considerations:

- When a call is LAI/BSR interflowed, the following information is forwarded with the call over public or private ISDN networks using QSIG or non-QSIG protocols: the LAI information, the collected digits, and the in-VDN time data in the ISDN SETUP message. The Multi-Site Routing related data is in addition to the associated ASAI user data (what was previously sent in a non-shared UUI IE) and the UCID data.

  NOTE:
  The forwarded LAI information is the same as what was previously sent in the LAI IE: VDN name also called LAI DNIS, put in queue time-stamp, priority level and type of interflow).

- The other call related information (calling party number (ANI), and name, II-Digits and CINFO digits) that is now tandemed with the interflowed call in the SETUP message continues to be forwarded in the same manner as before.

  NOTE:
  II-digits and CINFO are forwarded as codeset 6 IEs which may be a problem in some switched networks.

- At the remote end, the transported data is separated into its component parts for storage with the call, call vectoring, call processing/display, further interflow or tandeming, and forwarding to adjuncts. For example, the LAI info will be treated as though it was received as an incoming codeset 6 LAI IE including forwarding over ASAI as a codeset 6 LAI IE in event reports.
- When a status poll call is placed to the remote location, DEFINITY only forwards the UCID and caller information received from the original call.
- DEFINITY forwards the reply-best status data in the ISDN DISCONNECT message (in response to a status poll) over public or private ISDN PRI/BRI networks. In this case, the DISCONNECT message has a cause value of 31 “Normal–Unspecified” for wider international interoperability.

Providing LAI backward compatibility

The following summarizes backward compatibility issues:

- There is a trunk group option to specify whether to include an LAI IE (codeset 6 or 7) in the SETUP message for LAI interflowed calls. When this option remains as the default **y**, an LAI interflow (using the existing or enhanced LAI vector command) will include a codeset 6/7 LAI IE per existing operation for inter-operability with mixed previous DEFINITY G3/R5 releases (or earlier), System 85/Definity G2, and R6.3 or later

switches. The option must be **n** if the network does not support codeset 6/7 and/or this IE is not required. With trunk groups to all R6.3 (or later) switches, this option should always be **n**.

NOTE:
We recommend against using this option with a BSR-interflowed call because the remote switch must be R6.3 or later to work with BSR and codeset 0 information transport via shared UUI is required.

- Administer the ISDN Trunk Group option: `Send Codeset 6/7 LAI IE`. This option is valid even if Lookahead Interflow at the remote site is not active for tandem situations. Use of this option for LAI does not depend on the setting of the Vectoring (Best Service Routing™) customer option.
- If the ISDN trunk group option is set to send the LAI IE, this IE is sent in addition to the information forwarding (via codeset 0 shared UUI transport) when a call is LAI interflowed over a trunk in this trunk group. With shared UUI, the LAI data can be optioned to not be included in the UUI IE.
- Administer the Shared UUI priorities. This is important when the network byte limit on user information contents (the user data part of the UUI IE) is not large enough to carry the data needed for the customer application. Note that Shared UUI priorities do not apply to QSIG. To determine customer application data sizes, see "Determining User Information Needs" on page 9-7. For instructions on how to administer Shared UUI, see *DEFINITY ECS Release 6 Administration and Feature Description*, 555-230-522.

Determining User Information Needs

The network byte limit on user information contents (the user data part of the UUI IE) must be large enough to carry the data needed for the customer application.

NOTE:
The UUI IE uses three bytes for the header information and allows from 32 to 128 bytes for the user data portion. For example, if the network specifies that it can transport 32 bytes of user data, the UUI IE length is 35 bytes.

The user information capacity need is determined by adding the spaced required for each data item to be transported based on the following rules:

- Each included data item requires 2 bytes (for the header) plus the data.
- If the data item priority is set to blank in the Shared UUI Feature Priorities form, the data item is not sent (no space is allocated for it).
- For most data items, the data byte length depends on the application (customer configuration) except for UCID, In-VDN time, and Other LAI. These applications have a fixed byte length.

- If the administered Maximum UUI IE Size is exceeded, the lowest priority items are not included until the remaining data fit. If a specific data item at a higher priority exceeds the administered UUI IE size setting, that item will not be sent, leaving room for other lower priority items.
- If the network supports at least 82 bytes of user information, all R6.3 applications (each with maximum data) can be supported without changing the default priority settings. Non-QSIG private networks can support the entire 128 bytes of user information if the maximum size is set to 128. QSIG networks have no size limitations.

NOTE:
In both cases (a network supporting at least 82 bytes of user data in the UUI IE or a QSIG private network), you *do not* need to determine space allocation or administer priorities.

You can use Table 9-2 to determine the space required to send the various user data for your applications.

Table 9-2. Bytes Required to Send User Data

Type of User Data	Total User Data Bytes (with 2-byte header)	This type of user data...
ASAI	2 to 34 or 0 (calculated by 1 byte per byte of ASAI user info)	...is only needed with certain CTI applications. Space is only required when the CTI application sends user information.
UCID	10 or 0	...works well with BSR to track calls across multiple sites. This data item may not be included (taking no space) even if the priority is set to “1” depending on the trunk group setting and/or system feature settings.
In-VDN Time	4	...works with BSR to determine time before answer for calls and tracking as a single call across sites. It could be eliminated for calls with a short waiting time. If the priority field is not blank, it’s always included.

Continued on next page

Table 9-2. Bytes Required to Send User Data

Type of User Data	Total User Data Bytes (with 2-byte header)	This type of user data...
VDN Name	2 to 17 (calculated by 1 byte per character in name (max 15))	...works with BSR but could be eliminated if dedicated VDNs are used at receiving sites (with names assigned that display the equivalent information to the answering agent). An interflowed call received without the originating VDN name will use the incoming VDN name. If the priority field is not blank, the 2-byte header will always be included.
Collected Digits	4 to 11 or 0 (calculated by 1 byte per 2 digits plus 1 (max 16 digits))	...requires a whole byte for an odd number of digits. For example, 1 digit needs 2 bytes (1 plus 1), 7 digits need 5 bytes (4 plus 1), and 16 digits need 9 bytes (8 plus 1).
Other LAI Info	6	...is needed for existing CTI applications that require this data (in-queue time stamp, queue priority, and interflow type) from the LAI IE.
TOTAL	**82 bytes maximum**	

A Quick Example

Let's say that your public network supports only 32 bytes of user information. Your application requires 13 bytes of ASAI user information (15 bytes of user data), UCID (10 bytes of user data), and 8 collected digits (7 bytes of user data—4 plus 1 plus 2 for the header). It does not require Other LAI Information. Also, calls spend very little time at the sending DEFINITY because the calls are not queued before interflow takes place and tracking as a single call is not required.

By dedicating appropriately-named VDNs at the receiving DEFINITYs, the public network can support the application. Because the needed data items require the entire 32 bytes of user data, the priority fields for the `In-VDN Time`, `VDN Name`, and `Other LAI Information` must be set to blank.

Simple Troubleshooting for Information Forwarding

The following troubleshooting hints should be reviewed when information is not forwarded, even though you received no error messages while administering the Shared UUI feature, and all software and connections meet the minimum requirements:

- If DCS is used, make sure *all* ISDN trunks between DEFINITY switches used for DCS or remote AUDIX are configured in the D-channel mode.
- For each ISDN trunk administered with the Shared UUI option, make sure the UUI size does not exceed the UUI IE size that the network can support. For more information, see "Determining User Information Needs" on page 9-7.
- Make sure trunk group options are set correctly for the application and configuration.
- Applications may fail on networks supporting limited UUI transport. Administration determines which application's UUI will be transported in these cases. If a given application is failing, first check the administration to determine if the application in question has the highest priority. This applies to tandem nodes as well as originating nodes.
- Applications that originate UUI on tandem nodes can request that assigned priorities at the tandem node be applied to the resulting UUI. Therefore, it is possible for a tandem node to erase UUI information received from the originator.
- In other words, passing UUI through a tandem node transparently as required for UUS Service 1, does not apply to DEFINITY's proprietary shared UUI procedures.
- When a new application is implemented, you should run the "display events" command on a periodic basis for the appropriate vector. The resulting report will notify you if any enhanced information could not be sent.

Adjunct Routing

10

Introduction

Adjunct Routing allows an associated adjunct to make a call routing decision when it encounters an *adjunct routing* vector command during vector processing.

An adjunct is any processor connected to a DEFINITY System that can use ASAI features. The adjunct makes a routing decision according to caller information and/or agent availability, and it returns the routing response to the system.

Adjunct Routing can be used in conjunction with Call Prompting and Look-Ahead Interflow. When coupled with Call Prompting, Adjunct Routing can pass up to 16 digits that have been collected from the last relevant *collect digits* vector command. When coupled with Look-Ahead Interflow, Adjunct Routing can pass the LAI information element that was passed from the originating switch in the ISDN message.

Functions and Examples

The *adjunct routing* command provides a means for an adjunct ASAI processor to specify the destination of a call. The switch provides information in an ASAI route request message that the ASAI adjunct can use to first access a database and then determine a route for the call. In a typical application, the ASAI adjunct might use the dialed number, the Calling Party Number (CPN/BN), or the digits collected via Call Prompting to access caller information and thereby determine the call route.

An adjunct specified in an *adjunct routing* command can route a call to an internal number, an external number, a split, a VDN, an announcement extension, or a particular agent. An adjunct can also provide priority ringing and priority queuing.

Sending the Call Route Request

Screen 10-1 shows an example of a simple vector that uses adjunct routing:

```
1. adjunct routing link 1111
2. wait-time 60 seconds hearing ringback
3. route-to number 0 with cov n if unconditionally
4. disconnect after announcement 2000
```

Screen 10-1. Adjunct Routing Vector

In this vector, *1111* is the extension number of an ASAI link. Each ASAI link has a unique extension number, even in a configuration where there might be multiple ASAI links to the same adjunct.

When a call encounters an *adjunct routing* command, and if the call is not queued to a split, the switch sends an ASAI message requesting a call route over the specified adjunct link. The following list identifies the contents of the message, along with a comment or a brief explanation for each item:

- **Calling number information**. Calling party number or billing number (CPN/BN) provided by ISDN-PRI or R2MFC signaling facilities. If the call originates from a local switch extension, this extension is the calling number.
- **Originating line information (II-digits)**. Two-digit code provided by ISDN-PRI facilities indicating the type of originating line being used.
- **Called number**. Originally called extension (if a call is forwarded to a VDN), or the first VDN through which the call was routed (if the call was not forwarded to the VDN).
- **Routing VDN**. Last VDN that routed the call to the vector that contains the *adjunct routing* command.
- **Call identifier**. ASAI identifier that permits the ASAI adjunct to track multiple calls via either Event Notification or Third Party Call Control. (See the *DEFINITY Enterprise Communications Server CallVisor ASAI Technical Reference,* 555-230-220, for more information on ASAI.)

- **Look-Ahead Interflow information** (if any). Includes the original VDN display information, the priority level of the call at the originating switch, and the time that the call entered vector processing. (See Chapter 7.)
- **Digits collected via Call Prompting or Caller Information Forwarding (CINFO)** (if any; maximum of 16 digits). Digits are collected by the most recent *collect digits* command. (See Chapter 4 and Chapter 6.)
- **User-to-User Information** (if any). User-provided data associated with the call. If provided by ASAI, this data was provided in a 3rd-Party-Make-Call, Auto-Dial, or Route-Select message. If provided over ISDN, the data was in the SETUP message that delivered the call to this switch.

If the call is queued, the *adjunct routing* step is ignored, and vector processing continues at the next vector step.

NOTE:
For reasons that we discuss later in this chapter, you should always include a *wait-time* step, *announcement* or another *adjunct routing* step after an *adjunct routing* step.

Effects of ASAI Link Failure on Vector Processing

If the ASAI link specified in the *adjunct routing* step is down, the step is skipped.

An ASAI link failure can change the manner in which subsequent treatment (that is, *announcement* and/or *wait-time*) steps (if any) in the vector are processed. In some cases, such processing is influenced by the position the treatment steps occupy in the vector. In other cases, the positioning of these commands along with their relationship to specific *goto* commands come into play. For example, any *announcement* or *wait-time* step that immediately follows an *adjunct routing* step whose ASAI link is down is skipped.

NOTE:
In view of the previous thought, the second step after the *adjunct routing* step is often implemented as a default treatment. In Screen 10-1, for example, the default treatment in step 3 is a route to an attendant. After the switch recognizes that the ASAI link is down, this step executes immediately. (It can take up to 6 minutes for the switch to recognize that the link is down.) Otherwise, the step executes only if the application does not respond with a route within 60 seconds.

On the other hand, if a *goto* step follows such an *adjunct routing* step, the switch executes the *goto* step and then skips various treatment steps according to their position in the vector and based on the action of the *goto* step. Specifically, if the *goto* step *succeeds*, the switch skips any *announcement* or *wait-time* step that is the first non-*goto* step branched to by the *goto* step.

NOTE:
Actually, the first step to which a *goto* step is usually designed to branch is a nontreatment step (that is, a step containing a command other than a *wait-time* or an *announcement* command). Thus, the skipping of a treatment step according to the scenario described just before this note rarely occurs.

On the other hand, if the *goto* step fails, the system skips any *announcement* or *wait-time* step that immediately follows the *goto* step.

NOTE:
The *goto* step that fails can be at the end of a sequence of *goto* steps that branch to each other.

Vectors in Screen 10-2 can be used to illustrate the processes just described.

```
VDN (extension=1040  name=''Ad Route''  vector=40)
Vector 40
      1. adjunct routing link 1000 (link is down)
      2. wait-time 10 seconds hearing ringback
      3. adjunct routing link 2000 (link is down)
      4. goto step 7 if available-agents in split 20 < 1
      5. wait-time 10 seconds hearing ringback
      6. goto vector 50 if unconditionally
      7. goto step 10 if calls-queued in split 20 pri l > 50
      8. announcement 4001
      9. goto vector 50 if unconditionally
     10. route-to number 6000 with cov n if unconditionally
VDN (extension=6000  name=''Message''  vector=60)
Vector 60
      1. announcement 4000 (''We're sorry.  We
         are still unable to connect you to an agent.
         If you'd like to leave a message, please do so
         after the tone.  Otherwise, please call back
         weekdays between 8:00 A.M. and 5:00 P.M.
         Thank you.'')
      2. wait-time 6 seconds hearing silence
      3. messaging split 18 for extension 1500
      4. announcement 4010 (''We're sorry.  We
         were unable to connect you to our voice mail.
         If you'd like to try to leave a message again,
         please do so after the tone.  Otherwise, please
         call back weekdays between 8:00 A.M. and 5:00 P.M.
         Thank you.'')
      5. goto step 2 if unconditionally
```

Screen 10-2. Skipping/Non-Skipping of Treatment Commands with ASAI Link Down

Because we are assuming the adjunct link is down, the *adjunct routing* command in step 1 fails. Because the *wait-time* command in step 2 immediately follows an *adjunct routing* command whose adjunct link is down, the *wait-time* step is skipped. Step 3 contains another *adjunct routing* command whose adjunct link is

also down. As a result, the step fails, and control is passed to the *goto step* command in step 4, which is automatically executed.

Now, let's assume the *goto step* command in step 4 is not successful (that is, no branch is made because there is at least one available agent in split 20). In such a case, the *wait-time* step (step 5) following the unsuccessful *goto* step is skipped, and control is passed to the *goto vector* command in step 6. This step then routes the call to vector 50 (not shown), which is designed to queue the call and provide standard call treatment.

On the other hand, let's backtrack and assume the *goto step* command in step 4 is successful. In such a case, control is passed to step 7, where another *goto step* command determines whether there are more than 50 calls in split 20. If so, control is sent to step 10, where the *route-to number* command sends the call to vector 60, which allows the caller to leave a recorded message. The first step of vector 60 contains an *announcement* command, which is not skipped, since the treatment step is *not* the first non-*goto* step branched to by a *goto* step that follows an adjunct routing command whose ASAI link is down [the *route-to number* step (step 10) in vector 40 is the first such step]. Similarly, neither the *wait-time* step (step 2) nor the second *announcement* step (step 4) is skipped.

Returning to step 7 in vector 40, let's assume that there are 50 or fewer calls in queue. In such a case, the *goto* step fails and, as a result, the *announcement* step (step 8) that immediately follows this step is skipped. (Remember, even though this *goto* step does not immediately follow the *adjunct routing* step, the former step qualifies as a test case because it is branched to by another *goto* step that does immediately follow the *adjunct routing* step.) Thereafter, the *goto vector* step (step 9) routes the call again to vector 50, which is designed to queue the call and provide standard call treatment.

Table 10-1 summarizes the procedures discussed in this section, and refers back to the vectors presented in Screen 10-2.

Table 10-1. Example of the Relationship Between Treatment Steps and Goto Steps that Follow

Goto Step #(s)	Disposition of Goto Step(s)	Treatment Step #(s)	Disposition of Treatment Step(s)	Reason for Disposition of Treatment Step(s)
4	Fails	5	Skipped	Immediately follows failed *goto* step.
4, 7	First step succeeds, second step fails.	8	Skipped	Immediately follows failed *goto* step.

Table 10-1. Example of the Relationship Between Treatment Steps and Goto Steps that Follow

Goto Step #(s)	Disposition of Goto Step(s)	Treatment Step #(s)	Disposition of Treatment Step(s)	Reason for Disposition of Treatment Step(s)
	Both steps succeed.	1, 2 (both of vector 60)	Each step is executed.	Not the first non-*goto* steps accessed by a successful *goto* step. NOTE: Step 4 of vector 60 is also executed if the *messaging split* step (step 3) fails.

Awaiting the Response to the Call Route Request

After the switch sends a route request to the ASAI adjunct, vector processing continues with the following vector steps.

The step that follows the *adjunct routing* step in effect determines the maximum length of time the switch will wait for the ASAI adjunct to reply with a call route. Accordingly, you should almost always include either a *wait-time* step or an *announcement* step immediately after an *adjunct routing* step. Moreover, the system cancels the route request if vector processing encounters a step containing one of the following commands:

- *busy*
- *check split*
- *converse-on split*
- *queue-to split*
- *collect digits*
- *disconnect*
- *messaging split*
- *route-to*

If a valid call route is received by the system before one of the vector commands in the previous list is executed, the system routes the call to the destination

specified by the adjunct route. Otherwise, the route request is terminated without affecting vector processing.

Let's return to our suggested strategy of including a treatment step after the *adjunct routing* step. Accordingly, Screen 10-3 shows an example presented earlier that illustrates this approach:

```
1.  adjunct routing link 1111
2.  wait-time 60 seconds hearing music
3.  route-to number 0 with cov n if unconditionally
4.  disconnect after announcement 2000
```

Screen 10-3. Treatment Step Used as a Delay for Adjunct Routing

In step 2 of this example, the *wait-time* command specifies a delay period of 60 seconds. As a result, the switch in this case will wait up to 60 seconds to receive a reply from the adjunct. On the other hand, replacing the *wait-time* command in step 2 with an *announcement* command enables the switch to wait for no longer than the length of time it takes for the announcement to complete. Accordingly, judgement should be used in determining which of the treatment commands is best for the particular application.

Finally, note that the adjunct can also decide to not route a call by rejecting (that is, negatively acknowledging) the route request sent by the switch. Upon receiving a route request rejection, the switch terminates the *announcement* or *wait-time* step that is being executed for the call and then continues with the next vector step.

The *wait-time hearing i-silent* command is used in cases where it is important to allow the adjunct to decide whether to accept an incoming ISDN-PRI call. When this step is encountered after an *adjunct routing* step, the DEFINITY System does not return an ISDN PROGress message to the originating switch. This is particularly important for Network ISDN features and for the LookAhead Interflow feature.

Receiving and Implementing the Call Route

When the switch receives a call route (destination) from the ASAI adjunct, the switch first validates the route as follows:

1. The switch verifies the VDN's COR permits the call to be terminated at the adjunct-supplied destination.
2. The switch verifies that the adjunct-supplied information (destination number, ACD split, TAC/AAR/ARS access code, etc.) for the route is valid. This includes checking that the destination is compatible with the dial plan, and that the options specified by the adjunct are correct.

3. If the ASAI adjunct specifies the Direct Agent Call (DAC) option, the destination number (agent) must be logged into the adjunct-specified ACD split.
4. If the destination for the call is external, the switch verifies the trunk is available for the call.

If any of these conditions are not met, the route validation fails, and the switch does the following:

1. Discards the route.
2. Notifies the ASAI adjunct that the route is invalid.
3. Continues with vector processing.

If the route is valid, the switch does the following:

1. Terminates vector processing immediately.
2. Notifies the ASAI adjunct that the route is accepted.
3. Routes the call to the destination specified by the ASAI adjunct.

When the call is routed, the caller hears normal call progress tones and feedback. However, if the call is routed to an extension with no available call appearances and no coverage path, the caller hears the busy tone. Any other features that may be in effect at the adjunct-supplied destination (such as Send-All-Calls or Call Forwarding) interact with the routed call.

NOTE:
The operation described in the previous paragraph is similar to that for the *route-to with coverage* command.

Phantom Calls

A phantom call is a call that originates from a non-physical device (via an ASAI application) and may be placed anywhere. In general, phantom calls

- use less resources
- are treated like voice calls.

How Do Phantom Calls Work?

First, an application requests a phantom call by sending an ASAI third_party_ make_call or auto_dial capability message to DEFINITY.

If the specific extension of a station administered without hardware (AWOH) is specified as the originator, DEFINITY will place the call from that extension (if the extension is available).

It is also possible to specify a hunt group extension (with members that are AWOH extensions) as the originator.

How Are Phantom Calls Used?

Applications use phantom calls when they need to originate a call without using a physical device (and not tie up resources). For example, applications may need to

- reserve a queue slot

  Many call centers handle incoming requests as voice, video, data, voice messages, FAXs, and e-mail. Agents working in these call centers therefore need to handle the mix of requests. However, a single queue needs to manage and distribute the work load for these agents.

  For each non-voice request, the application can place a phantom call into the queue. When the phantom call reaches the head of the queue it is delivered to the agent. The agent is then given the corresponding work item on the desktop (for example, the FAX).

- conference control

  Multiple parties (both internal and external) may be conferenced into a call. The initial call is placed as a phantom call. When answered, the call is placed on hold (by the application) and another phantom call is made. The two calls are then conferenced together. This process repeats until all parties are added to the call.

- help with trunk-to-trunk transfers.

  Working with the Single Step Conference feature, applications can use the phantom call feature to help with trunk-to-trunk transfers (transferring a trunk-to-trunk call to another trunk). For information about single step conferences, see the *DEFINITY ECS CallVisor ASAI Technical Reference*, 555-230-220.

- alerts (wake-up, maintenance, and security)

  Applications can use phantom calls to alert users of various conditions (such as wake-up, maintenance, or security).

How Do Phantom Calls Affect Call Vectoring?

Because phantom calls can be directed anywhere, you must properly configure the application and the DEFINITY ECS to ensure that the vector commands executed for these calls make sense. For more information, see the *DEFINITY ECS CallVisor ASAI Technical Reference*, 555-230-220.

DEFINITY does not block phantom calls from executing any vector commands (phantom calls follow the same vector processing as regular voice calls). However, it may not make sense to have phantom calls enter certain vector steps such as

- announcements

  Because there is nobody listening (in a phantom call) to an announcement, there is no sense in playing one.

- “collect” steps.

  The “collect” step will fail because it can’t connect a tone receiver to a station AWOH (Administration without hardware); it will time-out because there is nobody to put in the expected digits.

Currently the “busy” step provides a busy tone to the caller. In this case, the “busy” step will disconnect the call because DEFINITY clears a phantom call when the call can’t terminate at a specific local destination.

Phantom Call Administration

There are no new administration forms for phantom calls. However, for this feature to work, keep the following in mind:

- Some stations AWOH must be administered.
- If a hunt group is specified as originator, a non-ACD hunt group with AWOH members needs to be administered as well.
- It is recommended that you assign meaningful names for the stations AWOH used by phantom calls if the calling party name will appear on the agent’s or Service Observer’s display.

Single-Step Conference

Until the DEFINITY 6.3 release, DEFINITY allowed conferences only if the controlling device was local and already on the call. Therefore, to conference parties through an ASAI application, the application started with a common local party in a call and then requested a sequence of steps like the following:

1. “third_party_hold”
2. “third_party_make_call”
3. “third_party_merge”

The Single-Step Conference (SSC) feature allows an application to

- add a device into an existing call (for example, to play announcements or make voice recordings)
- facilitate application-initiated transfers and conferences.

This is now done with a single request, which eliminates the need for a common party or for a local device to be connected in any call. Stations administered without hardware (AWOH) are eligible for single-step conference. The party may be added to a call with full visibility (listen/talk) or no visibility (listen only).

This feature is only available through an ASAI link (not from any DEFINITY phones). For more information about Single-Step Conference, see the *DEFINITY ECS CallVisor ASAI Technical Reference*, 555-230-220.

How Does SSC Work With Call Vectoring?

The call to which an extension is to be single-step conferenced is not allowed to be in vector processing, unless the visibility option with the Single-Step Conference request indicates “no visibility.”

Multiple Outstanding Route Requests

This feature allows multiple ASAI Route Requests for the same call to be active at the same time (simultaneously). The Route Requests can be over the same or different ASAI links.

The requests are all made from the same vector. They must be specified back-to-back, without intermediate (*wait-time, announcement, goto,* or *stop*) steps. If the adjunct routing commands are not specified back-to-back, current adjunct routing functionality will apply (that is, previous outstanding route requests will be cancelled when an adjunct routing vector step is executed).

The first Route Select response received by the switch will be used as the route for the call, and all other outstanding Route Requests for the call will be cancelled.

With Multiple Outstanding Route Requests, multiple adjuncts can process the route call request without waiting for the first route attempt to fail. An application can make use of this feature to distribute the incoming call load evenly across adjuncts, based on the adjunct’s current CPU load.

User Scenarios

Screen 10-4 shows a typical vector where multiple adjunct route requests to multiple links will be active at the same time. The first adjunct to route the call is the active adjunct (that is, it specifies which VDN the call should be routed to at that point).

```
1. wait-time 0 seconds hearing ringback
2. adjunct routing link 1001
3. adjunct routing link 1002
4. adjunct routing link 1003
5. wait-time 6 seconds hearing ringback
6. route-to number 1847 with cov n if unconditionally (default
   routing)
```

Screen 10-4. Sample Adjunct Routing Vector with Redundancy

Expert Agent Selection

11

Introduction

Expert Agent Selection (EAS) allows Call Center managers to provide the best possible telephone service to the caller by matching the needs of the caller with the skills or talents of the agents. Caller needs and agent skills are matched via Call Vectoring. Skills can be thought of as needs or desires on the part of the caller and also the abilities or talents of the agent.

Matching the call to an agent with the appropriate skills reduces transfers and call-holding time. Accordingly, customer satisfaction is increased. Also, since an entire agent group need not be trained at the same time for the same skills, employee satisfaction is increased.

EAS requires ACD and Call Vectoring. All of the existing ACD features and Call Vectoring capabilities can be used within EAS applications.

As with Call Vectoring calls, EAS calls are directed to VDNs, which in turn point to vectors. However, unlike Basic Call Vectoring, skills can be assigned in EAS to VDNs, or they can be associated with vector steps to represent caller needs. As for Call Vectoring calls, EAS calls are queued to ACD hunt groups. However, with EAS enabled, ACD hunt groups are called skill hunt groups instead of splits.

Skill hunt groups deliver calls to EAS agents. Agent skills are administered on the Agent LoginID form.

NOTE:
These are the same login IDs used by CMS (R3V2 and later releases) and BCMS.

Logical Agent implies that voice terminals are no longer preassigned to hunt groups; only when the agent logs in does the terminal become associated with all of the skill hunt groups assigned to the Agent Login ID.

With EAS optioned and enabled, ACD calls can also be directed to a particular agent, instead of to the skill hunt group, by using the Direct Agent Calling feature. The Direct Agent call is treated like an ACD call, but it waits in queue for a specific agent to become available. Direct Agent calls have a higher priority than skill hunt group calls.

This chapter describes EAS, and it explains, via a number of examples, how EAS is implemented. The chapter also discusses EAS upgrades. However, before you start with this chapter, you should take note of the following:

- With EAS, skill hunt groups replace splits. Skill hunt groups and splits cannot be administered simultaneously on a switch. This implies that all ACD hunt groups must be administered as either splits or skills. If EAS is optioned, all ACD hunt groups must be skill hunt groups.
- With EAS, all skill hunt groups (except for AUDIX hunt groups) must be vector-controlled.
- With EAS, non-ACD hunt groups are allowed, but they cannot be vector-controlled.
- Agent Login IDs are extensions in the dial plan, and they decrease the total number of stations that can be administered.
- With EAS, agents have a different login procedure and a single set of work mode buttons, regardless of the number of skills assigned to the agents.
- Skill hunt groups can distribute a call to the most-idle agent (UCD) or to the most-idle agent with the highest skill level for that skill (EAD). In either of these cases, the call can route to the most-idle agent for the specified skill, or to the most-idle agent in all the skills. Direct Department Call (DDC) distribution is not allowed for skill hunt groups.
- With either UCD or EAD distribution, the system can be administered to deliver calls based either upon greatest need, or agent skill level. This is the Call Handling Preference administered on the Agent LoginID form. When calls are in queue, greatest need delivers the highest priority oldest call waiting for any of the agent's skills. With skill level administration, the system delivers the highest priority oldest call waiting for the agent's highest level skill with calls in queue.
- The EAS-PHD customer option adds additional capabilities to the basic EAS capabilities.
  - — It increases the number of skills an agent can log into from four to 20.
  - — It increases the number of agent skill priority levels from two to 16.

For information on converting a Call Center to EAS, refer to Appendix L.

Identifying Caller Needs

Caller needs for a particular call can be identified by any of the following methods:

- Interpreting information (in the form of DNIS digits or ISDN messages) passed from the network
- Processing Call Prompting digits, digits entered at a Voice Response Unit (VRU), or CINFO digits forwarded by the network.
- Using ASAI or a VRU (such as CONVERSANT) in a host database lookup

To illustrate how a Call Center manager might match caller needs and agent skills (which can be viewed as capabilities needed from the caller's perspective), let's assume that a Call Center receives inbound calls from auto club members who speak Spanish or English. The callers in this case either need to plan a vacation route or have car trouble and are calling for assistance. See Table 11-1.

Table 11-1. Example of Caller Need—Agent Skill Matching

Caller Need	Capability Needed
tourist information	knowledge of the region
to speak Spanish	bilingual
emergency assistance	handle stressful callers
tow truck	access to dispatch systems)

Note in each case that the capability needed is such that it can accommodate the caller need. Let's examine the strategy behind matching these caller needs to capabilities as deemed appropriate by the Call Center manager:

- Tourist information/Knowledge of the region

  Travelers may need information while traveling or regarding a future trip. All assigned agents can provide this information.

- To speak Spanish/Bilingual

  Separate numbers are published and used as part of Spanish membership information, or Call Prompting is used after a general number is dialed.

- Emergency assistance/Handle stressful callers

  Separate emergency Road Service numbers are published and used, or Call Prompting is used after a general number is dialed. (For example, a number is provided for towing.)

Note that the Call Center chose to implement Call Prompting to identify Spanish-speaking callers and callers who require emergency assistance. This allows for quicker and more specialized treatment and therefore better satisfies the caller's needs.

In addition, some customers might prefer to speak to the agent he or she spoke to on a previous call. To accommodate this request, a Call Center manager can implement Direct Inward Dialing (DID) at the Call Center. Also, direct agent calling can be used to direct a call to a specific agent.

The following sections explain further how caller needs are identified.

DNIS/ISDN Called Party

Recall from Chapter 2 that a set of DNIS digits can be interpreted as a VDN. Table 11-2 presents four services and their corresponding telephone number (including DNIS digits) that might be provided to the caller.

Table 11-2. Examples of Services and Corresponding DNIS Digits

Service	Telephone Number	Corresponding DNIS
Emergency Road Service (English)	800-765-1111	6001
Emergency Road Service (Spanish)	800-765-2222	6002
Route Planning (English)	800-765-3333	6003
Route Planning (Spanish)	800-765-4444	6004
General (Call Prompting)	800-765-5555	6005

NOTE:
DNIS digits must be extensions that are reflected in the dial plan.

Call Prompting/VRU Digits/CINFO Digits

The Call Prompting/VRU/CINFO digits are entered by the caller in response to any recorded question about a caller's needs, or in the case of CINFO ced or cdpd digits, are provided by the call center host computer. For example, a hotline for a product may request that a product code be entered, or a travel service may request a 2-digit state code to indicate the state to which the caller would like to travel. Table 11-3 provides a prompt that encourages the caller to enter the appropriate Call Prompting digit for the needed service from the auto club.

Table 11-3. Example of a Prompt for Entering Call Prompting Digits

“For emergency road service, dial 1.
Para asistencia con su automovil, marque el dos.
For travel route directions, dial 3.
Para informacion sobre rutas, marque el cuatro.”

In Table 11-3, the caller is requested to dial the appropriate number between “1” and “4 (cuatro),” inclusive.

Host Database Lookup

A host database lookup uses DNIS and ANI (calling party’s number) to determine what skills are required or even the agent desired. For example, the database may show that the caller speaks Spanish and has been working with Agent 1367. To access host information, either ASAI or a VRU in conjunction with a *converse-on skill* step is used.

Direct Agent Calling

Some callers might prefer to speak to the agent he or she spoke to on a previous call. To accommodate this request, a Call Center manager can implement Direct Agent Calling at the Call Center. The caller can dial the LoginID of the desired agent directly. As an alternative, the caller can dial an 800 number and then be prompted to enter the LoginID of the desired agent. Vectors can be designed to handle the Call Prompting function.

Functions and Examples

This section explains how EAS is implemented. To this purpose, skill administration, the delivering of calls to a skill queue, and the routing of calls to an agent are discussed.

Administering Skills

A skill is an attribute that is:

- Administered as a skill hunt group
- Administered to VDNs (VDN skill preference)
- Assigned to agents (agent skill)

A skill hunt group is administered for each skill. A skill hunt group is a set of agents trained to meet particular customer needs.

Generally, if the ability “Spanish speaking” is assigned to skill 127, for example, it follows that Agent skill 127 and VDN skill 127 both signify “Spanish speaking.” However, note that the agent skill might be assigned a skill term that is broader than that for the corresponding VDN skill. For example, Agent skill 127 might be labeled “bilingual.” The implication is that agents with skill 127 can handle calls from Spanish callers as well as from callers who speak English.

Skills for an application can be illustrated via a table. Table 11-4 presents a very abbreviated example of such a skill distribution for an auto club. We will refer back to Table 11-4 several times in this chapter.

Table 11-4. Example of a Skill Table for an Auto Club

Supergroup-99	
Emergency Road Service-Bilingual-22	Route Planning-Bilingual-44
Emergency Road Service-English-11	Route Planning-English-33

In Table 11-4, five skills are defined. Each skill indicates knowledge or an ability (on the part of the agent) or a need for knowledge (on the part of the caller) vis-a-vis an auto club. One or more of these skills can be attributed to the agent according to the agent’s expertise with the corresponding highway service(s) and his or her language-speaking ability. Similarly, one or more of these skills can be considered “needs” on the part of the caller.

Table 11-4 is arranged in such a manner that the agents at the top level have the broadest knowledge (that is, these agents can handle emergency road service and route planning calls and can speak Spanish). The top level (skill group) here is called “Supergroup,” and it contains agents who, as a group, can take any type of call regarding the auto club. Accordingly, this skill group serves as a “backup” skill group. As you descend through the table, each sublevel corresponds to a group of agents who have more specific skills and can therefore take more specialized calls.

Calls can be distributed to the most-idle agent by using either the Uniform Call Distribution (UCD) option or the Expert Agent Distribution (EAD) option. UCD distributes calls from the skill hunt group to the most-idle agent who has this skill assigned at any priority level. This scenario provides a more even distribution to calls and therefore keeps agents equally busy. EAD distributes calls from the skill hunt group to agents to an available agent who has the highest skill level. Skills assigned to an agent at higher skill levels indicate a higher level of expertise or preference by the agent than any lower skill level skills assigned to that agent. EAD distribution provides the caller with the best or most expert agent match.

Agents are usually given a preference for higher skill level calls. However, the system can be administered to give agents a preference for the greatest need call. The greatest need call is the highest priority oldest call waiting for any of the agent's skills.

Multiple Call Handling on Request and Forced Multiple Call Handling allow an agent to receive additional ACD calls either after putting a call on hold, or when active on another ACD call. Forced Multiple Call Handling can be used to give priority to an ACD call over an in-progress non-ACD call, or to give priority to a call from one skill over an in-progress call from a different skill. See "Multiple Call Handling" in the *DEFINITY Enterprise Communications Server Release 6 Administration and Feature Description*, 555-230-522*,* for more information.

To administer skills, the relevant Hunt Group form must be completed. The form appears as in Screen 11-1.

```
                                HUNT GROUP

    Group Number:                Group Extension:          Group Type:
      Group Name:                          Skill?                ACD?
           Queue?                        Vector?                AAS?
   Security Code:      Night Service Destination:                COR:
ISDN Caller Disp:                 Coverage Path:                 TN:

             Measured:                            Supervisor Extension:
 Priority On Intraflow?                         Inflow Threshold (sec):
   Controlling Adjunct:                         Adjunct Link Extension:
Multiple Call Handling?                Acceptable Service Level (sec):
            Objective:

           Queue Length:
Calls Warning Threshold:         Calls Warning Port:           Extension:
 Time Warning Threshold:          Time Warning Port:           Extension:
     Timed ACW Interval:
                    Redirect on No Answer (rings):     Redirect to VDN:
               Forced Entry of Stroke Counts or Call Work Codes?
```

Screen 11-1. Hunt Group Form

The skill, ACD, and Vector fields must all contain "y." Instructions for completing this form are included in *DEFINITY Enterprise Communications Server Release 6 Administration and Feature Description*, 555-230-522.

VDN Skills

VDN skills (along with certain vector commands) define the requirements for routing calls to an ACD agent with a particular set of skills. A maximum of three different skills can be administered to a VDN. Skills administered to a VDN are commonly called VDN skill preferences. VDN skill preferences are labeled “1st,” “2nd,” and “3rd.”

NOTE:
While skills can be optionally assigned to VDNs, the vector controls when and to what VDN skill the call queues. The application of VDN skills is described later in this chapter.

It is recommended that the 1st skill administered to a VDN be the skill required or desired to service a call to that VDN. The 2nd and 3rd skills administered to a VDN should represent other skills that are allowed to handle calls to that VDN. All skills to a VDN are optionally administered.

Table 11-5 illustrates how skill preferences can be assigned to the five VDNs used for the auto club that we discussed earlier. For each VDN, the corresponding call type and the number of the vector to which the VDN points are indicated. Be sure to refer back to Table 11-4 for a description of each skill.

Table 11-5. Example of VDN Skill Preferences Assignments

		Skill Preferences			
Call type	**VDN**	**1st**	**2nd**	**3rd**	**Vector**
General number	6005				1
Emergency Road Service (English)	6001	11	22	99	3
Emergency Road Service (Spanish)	6002	22		99	2
Route Planning (English)	6003	33	44	99	3
Route Planning (Spanish)	6004	44	99		2

From Table 11-5 note that two VDNs point to Vector 3, two VDNs point to Vector 2, and one VDN points to Vector 1. Note also that a 1st and 3rd VDN skill Preference, but no 2nd VDN skill Preference, are assigned to VDN 2222. Such a scenario implies that the call to this VDN (if not already answered) will wait longer before queuing to the backup skill (Supergroup-99, in our example), provided the vector is designed to execute accordingly.

Now, let's take a look at Table 11-6, which illustrates the skill preferences assigned for one specific VDN (6003) that is used for the auto club:

Table 11-6. Skill Preferences Assignments for VDN 6003

VDN 6003 - Skill Preferences		
1st:	33	Directed to an agent who is knowledgeable about Route Planning and speaks English
2nd:	44	Directed to an agent who is knowledgeable about Route Planning and is bilingual
3rd:	99	Directed to an agent who can field all calls

In Table 11-6, the first VDN skill preference corresponds to a knowledge area that could be considered a subset of the knowledge area represented by the second (and, taking it a step further, the third) preference. Similarly, the second VDN skill Preference corresponds to a knowledge area that could be considered a subset of the knowledge area represented by the third preference. Such an approach is commonly used to assign VDN skill preferences. The result of this approach is that the longer a call waits, the larger the pool of agents that the ACD considers for handling the call.

Now, recall that the vector number to which each VDN associated with the auto club has already been provided in Table 11-5 of this section. A quick glance at the table shows that VDN 6003 points to Vector 3. As such, the skill requirements associated with the VDN are "passed along" to the vector. This process can be illustrated as in Figure 11-1.

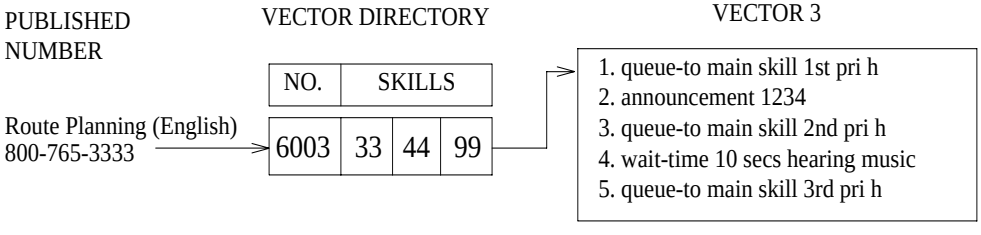


Figure 11-1. Example of VDN Skill Implementation

Let's assume that the English-speaking caller needs information on Route Planning and dials the appropriate number (800-765-3333). Network 800

features direct the call to the DEFINITY System extension 6003 (a VDN), the call enters the switch and is directed to VDN 6003, which points to the appropriate vector. As illustrated in Table 11-6, VDN skill Preferences 33, 44, and 99 are administered as the 1st, 2nd, and 3rd skill preferences, respectively, for VDN 6003.

We will hold off explaining the vector processing of our application until "Delivering the Call to the Skill Queue" on page 11-13.

Vector Directory Number (VDN) Form

The Vector Directory Number (VDN) form is used to administer VDN skills. The form appears as in Screen 11-1.

```
                                                    Page  1 of   1
                     VECTOR DIRECTORY NUMBER

                       Extension: 50000
                            Name:
              Allow VDN Override? n
                             COR: 59
                              TN: 1
                    Vector Number: 234
                       AUDIX Name:
           Messaging Server Name:
                        Measured: none
Acceptable Service Level (sec):
 VDN of Origin Annc. Extension: 301
                       1st Skill:
                       2nd Skill:
                       3rd Skill:

              Return Destination:
          VDN Timed ACW Interval:
                 BSR Application:
  BSR Available Agent Strategy: 1st-found
```

Screen 11-1. Vector Directory Number (VDN) Form

NOTE:
While skills can be optionally assigned to VDNs, the vector controls when and to what VDN skill the call queues.

Complete instructions for completing the form are included in *DEFINITY Enterprise Communications Server Release 6 Administration and Feature Description,* 555-230-522.

Call Vector Form

Completion of the Call Vector form is required for using vectors with EAS. The form contains three pages. However, if your vector contains 11 or fewer instructions, you need populate only the first page of the form. Page 1 of the Call Vector Form appears as in Screen 11-2.

```
                                                          Page 1 of 3
                              CALL VECTOR
  Number:                  Name: ________________    Multimedia? n     Lock? n


    Basic? y   EAS? n    G3V4 Enhanced? y   ANI/II-Digits? y   ASAI Routing? n
Prompting? y   LAI? n   G3V4 Adv Route? y           CINFO? y            BSR? n

  01 _____________
  02 _____________
  03 _____________
  04 _____________
  05 _____________
  06 _____________
  07 _____________
  08 _____________
  09 _____________
  10 _____________
  11 _____________
```

Screen 11-2. Call Vector Form (Page 1 of 3)

NOTE:
While skills can be optionally assigned to VDNs, the vector controls when and to what VDN skill the call queues.

Complete instructions for completing the Call Vector form are provided in *DEFINITY Enterprise Communications Server Release 6 Administration and Feature Description,* 555-230-522.

Agent Skills

Agents are trained or hired to accommodate specific caller needs. Agent skills represent and define the ability of the agent to handle calls that require these skills. Agents are assigned skill numbers based on such characteristics as training or knowledge, access to systems or information, language ability, and interpersonal traits. Examples of agent skills include the following: speaks Spanish, knows about widget “X,” can handle complaint calls, has access to a database, etc.

If an agent has multiple skills, a single skill group can be created for each set of skills. Agent skills are assigned to agents by completing the Agent Login ID form (see the “ACD Login ID Dialing” section later in this chapter).

It is highly recommended that you create a separate skill hunt group for direct agent calls. Direct agent calls are queued to the skill that is administered as the Direct Agent Skill on the Agent LoginID form. If an agent is not able to log in to his or her Direct Agent Skill, Direct Agent calls are queued to the first-administered highest-level skill.

Table 11-7 illustrates the assignment of agent skills.

NOTE:
Refer to Table 11-4 for a description of the skills indicated in Table 11-7.

Table 11-7. Example of Agent Skill Assignments

Agent	Skills Assigned			
Jan O'Hara	22 (L1)	44 (L2)		
Sam Thomas	99 (L1)			
Sue Carlson	22 (L1)	11 (L1)	44 (L2)	33 (L2)
Mark Davis	44 (L1)			
Amy Brown	44 (L1)	22 (L2)		

Table 11-7 indicates the skills assigned to five specific agents along with the associated skill level. Remember, without EAS-PHD a maximum of four agent skills may be assigned to any one agent with one of two preference levels. With EAS-PHD up to 20 skills can be assigned to each agent with one of sixteen preference levels. In Table 11-7, we see that four agent skills (22, 11, 44, 33) are assigned to Sue Carlson. These assignments indicate that Sue is bilingual and can service callers who need emergency road service or information on route planning. On the other hand, we see that only one agent skill (99—Supergroup) is assigned to Sam Thomas. This means that Sam is serving only as a backup.

In Table 11-7, a L1 or L2 next to the skill number indicates whether the agent skill is assigned as a level 1 or level 2 skill. For example, Jan O'Hara has "Emergency Road Service-Bilingual" as a level one skill and "Route Planning-Bilingual" as a level two skill. This means that whenever Jan O'Hara becomes available for an ACD call, provided that the Call Handling Preference is skill-level, the ACD software first looks for English-speaking callers requesting information on "Emergency Road Service" from the agent. Only if there are no callers requesting "Emergency Road Service" does the ACD software look for English-speaking callers requesting information on "Route Planning." If the Call Handling Preference was greatest-need, Jan O'Hara would receive the highest priority oldest call waiting for either "Emergency Road Service" or "Route Planning-Bilingual" each time she becomes available.

For any given application, EAS puts no restrictions on which agent skills can be assigned to an agent.

NOTE:
Agent skills are administered by completing the Agent Login ID form. This form is presented in "ACD Login ID Dialing" on page 11-21. Complete instructions for completing the form are provided in *DEFINITY Enterprise Communications Server Release 6, Administration and Feature Description,* 555-230-522.

Delivering the Call to the Skill Queue

Now that we have defined and illustrated VDN skills and agent skills, we are ready to discuss (with the help of an example) how a call is delivered to a skill hunt group queue via vector processing.

The skills assigned to a VDN define the requirements in the vector for routing calls to an ACD agent with a particular set of skills. These skills become active for an ACD call whenever a *queue-to skill* command is executed. The skills also become active whenever a *check skill* command is executed and the threshold condition is met. Once a skill is active for an ACD caller, the call cannot be delivered to an available ACD agent unless the agent also has one of the active VDN skills. Take a look at Figure 11-2.

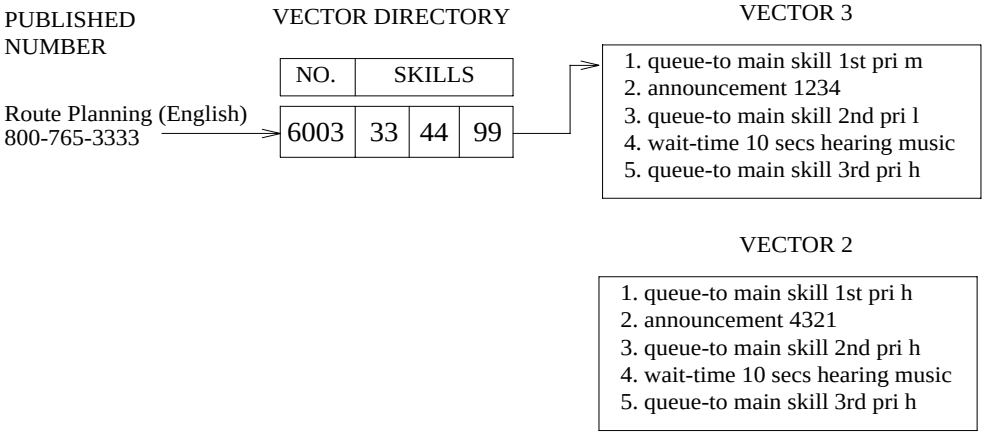


Figure 11-2. Process for Delivery of a Call to a Skill Queue

Let's assume that an English-speaking caller needs information on "Route Planning" and dials the appropriate number (800-765-3333). In such a case, the call enters the switch and is directed to VDN 6003, which points to Vector 3. Once vector processing starts, the *queue-to skill* command in step 1 queues the

call to the skill hunt group corresponding to the 1st VDN skill (33-Route Planning-English). If an agent with skill 33 is available, this agent answers the call. If such an agent is not available, the call is eventually queued to the skill hunt group corresponding to the 2nd VDN skill (44-Route Planning-Bilingual) by the *queue-to skill* command in step 3. This time, if an agent with skill 44 is available, this agent answers the call. If the call is still not answered, the call is eventually queued to the skill hunt group corresponding to the 3rd VDN skill (99-Supergroup) by the *queue-to skill* command in step 5.

Note that Figure 11-2 also shows Vector 2. This vector would have been executed if a Spanish-speaking caller had called into the switch. Accordingly, the announcement provided in Vector 2 is in Spanish, whereas the announcement in Vector 3, which is executed in our example, is in English.

Note also that each of the *queue-to skill* commands in Vector 2 queues the call at a *high* priority, whereas only one of the *queue-to skill* commands in Vector 3 queues the call at this high a priority level. The strategy presented here is very valuable in cases where there is a limited number of bilingual agents in that such agents would be available to service callers who speak only Spanish more quickly.

VDN skills can also be used in *check skill, messaging skill,* and *converse-on skill* commands. Within any of these commands, a specific skill number can be used instead of a VDN skill Preference, provided the relevant skill hunt group is correctly administered. For example, step 5 might have read “queue-to skill 99 pri h.” We will discuss this concept further in the “Super Agent Pool” section.

Procedure Using Call Prompting

The procedure described in the previous section can be enhanced by using Call Prompting. For example, the user can dial a “general” telephone number whose VDN points to a Call Prompting vector.

Staying with our auto club example, recall that in Table 11-2 we define “800-765-5555” as the “general” telephone number for the service. Recall also that in Table 11-5 we identify “6005” as the VDN for this 800 number. Also, we indicate that VDN 6005 points to Vector 1.

Screen 11-3 shows an example of how Vector 1 might appear:

```
1.  wait-time 0 seconds hearing ringback
2.  collect 1 digits after announcement 5678
    ("For emergency road service, dial 1.
    Para asistencia con su automovil, marque el dos.
    For travel route directions, dial 3.
    Para informacion sobre rutas, marque el cuatro.")
3. route-to number 6001 with cov n if digit = 1
    (English Emergency Road Service VDN)
4. route-to number 6002 with cov n if digit = 2
    (Bilingual Emergency Road Service VDN)
5. route-to number 6003 with cov n if digit = 3
    (English Route Planning VDN)
6. route-to number 6004 with cov n if digit = 4
    (Bilingual Route Planning VDN)
7. route-to number 6002 with cov n if unconditionally
    (Bilingual Emergency Road Service VDN)
```

Screen 11-3. Call Prompting Vector for the Auto Club

Once the caller dials “800-765-5555,” the call enters the switch and is directed to VDN 6005, which points to our Call Prompting vector. At this point, vector processing begins. Step 1 provides ringback if the caller has to queue for the announcement in step 2. The *collect digits* command in step 2 first provides an announcement requesting the caller to dial “1,” “2,” “3,” or “4,” depending upon the caller need and the caller’s language speaking ability. If the caller dials a digit other than one of the four specified, each of the *route-to...if digits* commands in steps 3 through 6 fails, and control is passed to the *route-to...if unconditionally* command in step 7, which unconditionally routes the call to VDN 6002. This VDN is assigned the “Bilingual Emergency Road Service” skill and points to Vector 2, which is provided in the previous section.

On the other hand, returning to the *collect digits* step, let’s assume that the caller dials “4.” In such a case, steps 3 through 5 fail because the required digit (“1,” “2,” or “3,” respectively) has not been dialed. Thereafter, control is passed to step 6, where the *route to...if digit* command finds a digit match and consequently routes the call to VDN 6004. This VDN is assigned the “Bilingual Route Planning” skill and also points to Vector 2, which is provided in the previous section.

NOTE:
VDN Override applies to the skills assigned to the VDN. See “VDN Override” on page 2-11 for more information.

Super Agent Pool

EAS allows a skill hunt group to function as a super agent pool. A super agent pool is a “backup” group of one or more agents that is able to handle many if not all types of calls coming into the application. In our auto club examples, Skill Hunt Group 99 (Supergroup) serves as a super agent pool. Also, you might recall

that “99” appears as both a VDN skill and an Agent skill. However, a super agent pool can be assigned a skill hunt group number that is not assigned to a VDN skill. This can (and should) be done whenever the application requires four levels within the skill table distribution as shown in Table 11-8.

To illustrate this, let’s first modify and expand on the Auto Club Skill Table (Table 11-4) presented earlier in this chapter:

Table 11-8. Modified Skill Table for the Auto Club

Supergroup-99			
Emergency Road Service-Bilingual-88		Route Planning- Bilingual-77	
English-66	Spanish-55	English-44	Spanish-33
Bostonian-11	Castilian-13	Bostonian-15	Castilian-17
New Yorker-12	South American-14	New Yorker-16	South American-18

Besides a new skill numbering scheme, our modified skill table has four levels instead of the three levels provided in Table 11-4. Except for the skill numbering scheme, the top two levels (Supergroup-99 and Emergency Road Service-Bilingual-88/Route Planning-Bilingual-77) remain unchanged. However, note that the next level has been reorganized into segments to indicate the ability to speak English or Spanish. Finally, note that a new level has been added to denote particular types of accents or pronunciation in English and Spanish.

Now, let’s take a look at Table 11-9, which illustrates how some of the skills in Table 11-8 are administered to one relevant VDN (1616):

Table 11-9. Skill Preferences Assignments for VDN 1616

VDN 1616 - Skill Preferences		
1st:	16	Knows about Route Planning, speaks English, has New York accent
2nd:	44	Knows about Route Planning, speaks English
3rd:	77	Knows about Route Planning, is bilingual

Now we are ready to consider Screen 11-4, which is a variation of a vector presented earlier in this chapter.

```
1. queue-to skill 1st pri m
2. announcement 4555
3. queue-to skill 2nd pri l
4. wait-time 10 seconds hearing music
5. check skill 3rd pri l if calls-queued < 3
6. announcement 4666
7. check skill 99 pri l if available-agents > 0
```

Screen 11-4. Modified Vector to Accommodate a Super Agent Pool

Let's assume an English-speaking caller needs information on "Route Planning" and would like to speak to an agent with a New Yorker accent. In such a case, the caller dials the appropriate number (800-765-1616, for example). Accordingly, the call enters the switch and is directed to VDN 1616, which points to the vector in the previous screen. Once vector processing starts, the *queue-to skill* command in step 1 queues the call to the skill group corresponding to the 1st VDN skill (New Yorker-16). If an agent with skill 16 is available, this agent answers the call. If such an agent is not available, the call is eventually queued to the skill group corresponding to the 2nd VDN skill (English-44) by the *queue to main skill* command in step 3. This time, if an agent with skill 44 is available, this agent answers the call. If the call is still not answered, the *check skill* command in step 5 attempts to queue the call according to the parameter indicated (if calls-queued < 3) to the skill group corresponding to the 3rd VDN skill (Route Planning-Bilingual-77). If the call is queued, and if an agent with skill 77 is available, this agent answers the call. If the call is not queued, or if it is queued and an agent with skill 77 is not available, the *check skill* command in step 7 is executed.

Before we discuss the execution of step 7, note that a specific skill hunt group number (99) and not a VDN skill Preference designation (1st, 2nd, or 3rd) is included within the *check skill* command. Since the skill table for the application involves four levels of skills, and since there can be no more than three VDN skills, the specific skill group number (99) for the super agent pool must be included within the queuing command to allow caller access to the pool. Whereas a VDN skill is always represented in a vector by the term "1st," "2nd," or "3rd," a super agent pool is always represented by a whole number according to the parameters of the relevant switch (see the manual pages for the queuing commands in Appendix A).

Returning to the vector execution, the *check skill* command in step 7 attempts to queue the call according to the parameter indicated (if available-agents > 0) to the super agent pool (Supergroup-99). If the call is queued, and if an agent in the super agent pool is available, this agent answers the call.

NOTE:
If the call has already queued to all three VDN skill hunt group preferences, it will not queue to the specific skill hunt group. This reflects the restriction

that a call can only queue to a maximum of three splits or skills. The best approach is to test the splits/skills first to determine where to queue the call. Also, see "Expected Wait Time (EWT)" on page 5-2.

Routing the Call to an Agent

With EAS optioned, an agent becomes associated at login with one or more skill hunt groups. A single set of work mode buttons applies to all the skills assigned to a logged-in agent (for example, if the agent selects "Aux Work," the agent is in Aux Work for all the skills associated with the agent). Therefore, logged-in agents need only a single set of work-mode buttons for all relevant skill hunt groups.

Calls may be routed to the agent from a skill hunt group, by dialing an Agent Login ID, or by dialing an agent voice terminal extension directly. The following sections discuss these procedures.

Delivery from a Skill Hunt Group

An incoming call is matched to an agent who has at least one of the three VDN skills required to handle the call. This matching is done via ACD queuing and via *queue-to skill, check skill, messaging skill,* or *converse-on skill* commands in the vector. If more than one agent is available for a call, the call is delivered according to whether EAD or UCD is administered for the skill hunt group.

For any one login session, an agent can have a maximum of four skills, or a maximum of twenty skills with EAS-PHD. Each agent skill is administered with a skill level.

Remember that when the Call Handling Preference is administered as greatest-need, the agent receives the highest priority oldest call waiting for any of the agent's skills. If the Call Handling Preference is skill-level, the ACD software distributes the call that is waiting for the agent's highest skill-level skills whenever the agent becomes available. If no calls are waiting for the highest skills, the queued calls for the next highest skills are distributed to the agent, and so on. See Expert Agent Selection in the *DEFINITY Enterprise Communications Server Release 6 Administration and Feature Description (*555-230-522) for a more thorough description of call routing when agents are available or when calls are in queue. The following scenario describes call distribution when the Call Handling Preference is skill-level.

Once an agent becomes available, he or she receives a waiting call in the following order:

1. Oldest Direct Agent call waiting for the agent if the Direct Agent Skill is administered at the agent's highest skill level
2. Oldest call waiting at the highest priority for the highest skill-level skill
3. Oldest call waiting at the next highest skill-level skill, and so on.

For example, let's assume that Jill is the only agent with skills 22 (L1), 13 (L1), 23 (L1) and 47 (L2). Let's also assume that, while Jill is in AUX work mode, five calls are queued, as illustrated in Table 11-10 (which also provides the skill and priority level associated with each call):

Table 11-10. Example of Skill Call Queue Sequence

Call	Time In Queue	Skill No.	Priority Level
A	8:00	13	Medium
B	8:01	47	Top
C	8:02	23	Direct Agent
D	8:03	22	Top
E	8:04	22	Medium

Given this scenario, Table 11-11 indicates and explains the order in which Jill would handle the five calls.

Table 11-11. Example of Skill Call Distribution for a Single Agent

Call Handled	Reason
C	Only Direct Agent call queued at highest level skill.
D	Oldest call waiting at the highest priority for highest skill-level skills (Call B has the same priority level (Top), but it is assigned a lower skill level (47). Also, Call E has the same skill (22), but it has a lower priority level (Medium) and has not been waiting as long as Call D).
A	Oldest call waiting at the highest priority level for highest skill-level skills (Call E also has a primary skill (22) and the same priority level as Call A, but Call A has been waiting four minutes longer than Call E).
E	Only remaining call with the highest skill level (22) (Call B has a lower skill level (47)).
B	Last remaining call, and the only one that has the lower skill level (47).

If no calls are waiting when an agent becomes available, the agent is placed into the agent queue according to the call distribution method in effect. For UCD, the

agent is placed at the bottom of the most-idle agent queue. For EAD, the agent is placed at the bottom of the agents with the same skill level.

Table 11-12 illustrates a call scenario that is valid for either UCD or EAD:

Table 11-12. Example of UCD/EAD Call Scenario

Time	Event	Skills
9:00	Jill logs in	22(L1), 13(L1), 47(L2)
9:01	Jill available	22(L1), 13(L1), 47(L2)
9:02	Jack logs in	22(L1), 47(L1)
9:03	Jack available	22(L1), 47(L1)
9:04	Call A arrives	47
9:05	Call A drops	47
9:06	Call B arrives	13
9:07	Call B drops	13
9:08	Call C arrives	22

Given this scenario, Table 11-13 illustrates how Calls A, B, and C are distributed via UCD and EAD:

Table 11-13. Example of Call Distribution via UCD/EAD

Time	UCD or EAD?	Result	Reason
9:04	UCD	Jill receives Call A.	Jill is the most idle agent for skill 47.
	EAD	Jack receives Call A.	Jack is the “more expert” agent because he has skill 47 as a level 1 skill whereas Jill has skill 47 as a level 2 skill.
9:06	UCD	Jill receives Call B.	Jill is the only agent logged into skill 13.

Continued on next page

Table 11-13. Example of Call Distribution via UCD/EAD

Time	UCD or EAD?	Result	Reason
	EAD	Jill receives Call B.	Jill is the only agent with skill 13.
9:08	UCD	Jill receives Call C.	Jill is the most idle agent for skill 22. She receives Call C even if she handled Call A.
	EAD	Jill receives Call C.	Both Jill and Jack have skill 22 as a level 1 skill, but Jill has been logged in two minutes longer than Jack (that is, she is the most idle agent).

ACD Login ID Dialing

The ACD Login IDs used in EAS are extension numbers included in a station numbering plan but not administered as stations. These IDs are administered by using the Agent Login ID form, Screen 11-5. If EAS-PHD is not optioned you can only administer four skills.

```
add agent-loginID 9011                                        Page  1 of  1
                              AGENT LOGINID

                Login ID: 9011_                                    AAS? _
                    Name: ____________________________            AUDIX? _
                      TN: 1__                              LWC Reception: msa-spe
                     COR: 1                  AUDIX Name for Messaging: _______
           Coverage Path: ____   Messaging Server Name for Messaging: _______
           Security Code: ____              LoginID for ISDN Display? n
      Direct Agent Skill: ___                                Password: _________
Call Handling Preference: skill-level          Password (enter again): _________
       Service Objective? _                              Auto Answer: ______

   SN  RL SL  PA      SN  RL SL  PA       SN  RL SL  PA       SN  RL SL  PA
1: __  _  __  ___  6: __  _  __  ___  11: __  _  __  ___  16: __  _  __  ___
2: __  _  __  ___  7: __  _  __  ___  12: __  _  __  ___  17: __  _  __  ___
3: __  _  __  ___  8: __  _  __  ___  13: __  _  __  ___  18: __  _  __  ___
4: __  _  __  ___  9: __  _  __  ___  14: __  _  __  ___  19: __  _  __  ___
5. __  _  __  ___ 10: __  _  __  ___  15: __  _  __  ___  20: __  _  __  ___

      WARNING:  Agent must log in again before skill changes take effect
```

Screen 11-5. Agent Login ID Form

With EAS, an Agent's ACD Login ID is associated with a specific voice terminal only when the agent actually logs in at that terminal. When the agent logs off, the association of the agent's ACD Login ID with a specific voice terminal is removed. If an agent does not answer a call, or if the agent is logged out, the call goes to the busy points on the coverage path.

When the agent logs in, the voice terminal display indicates the agent's skill assignments.

The agent logs in by doing the following:

- Going off-hook or selecting a line appearance
- Upon hearing the dial tone, entering the login Feature Access Code (FAC) or selecting the Login Abbreviated Dialing (AD) button
- Upon hearing the dial tone, entering the 1- to 5-digit Login ID

  NOTE:
  If someone is already logged in at that terminal, the agent will hear intercept.

- Upon hearing the dial tone, entering (optionally) the 0- to 9-digit password.

  NOTE:
  If the agent is using a DCP voice terminal such as CallMaster, the password digits will not be shown unless an abbreviated dial button is used. BRI voice terminals will show the password digits.

Once the login is accepted, confirmation tone is given. Also, the skills assigned are displayed for five seconds on the voice terminal display. If more skills are assigned then can be displayed, a plus sign (+) appears at the end of the display. If a skill is administered but the agent was not logged into the skill, the skill number is displayed with a star (*). The previous login sequence allows an ACD call to be directed to a specific agent and to have that call tracked and treated as an ACD call.

When an EAS agent logs into a station with the station administered for audible message waiting, the agent will receive Audible Message Waiting tone only when calls are waiting for the agent login ID extension. When the agent logs out, Audible Message Waiting tone then applies again to messages waiting for the physical extension. This field does not impact whether an agent hears the EAS Login-ID Message Waiting tone during the login process.

The Message Waiting Lamp by default tracks the status of messages waiting for the logged in EAS agent LoginID rather than messages for the physical terminal. The operation of the Message Waiting Lamp can be changed so that it tracks the status of messages waiting for the physical terminal where the agent is logged in. See the Feature-Related System-Parameters form in the *DEFINITY Enterprise*

Communications Server Release 6 Administration and Feature Description (555-230-522) for more information.

In addition to skill assignments, the following capabilities are associated with agents' login IDs.

- **Call Routing**

  A call to the Login ID reaches the agent independent of the voice terminal the agent is currently using. In other words, such a call is sent to the voice terminal at which the agent is currently logged in.

  If the proper Class of Restrictions (COR) is set, callers can initiate a Direct Agent call either by dialing the Login ID extension directly or by calling a VDN that points to a vector containing first a prompt for the Login ID and then a *route-to digits* command. This allows external callbacks via Direct Inward Dialing (DID) or an 800 number. Both the receiving agent's Login ID COR and the originator's (caller's) COR must have "Direct Agent Calling" set to "y." The caller's COR is for the following:

  — Voice terminal extension (for internal calls or transfers)

  — Trunk group (for DID calls)

  — VDN (for prompted calls)

  If the call covers or is forwarded, the COR of the originator (or VDN) and the final agent is used. All feature functionality for ACD calls, except Queue Status indications, is available for Direct Agent Calls.

  Internal and external users can originate Direct Agent calls by dialing the agent's login ID. Also, Direct Agent calling can be used to transfer ACD calls from one agent to another agent.

  If an agent receiving the Direct Agent Call is staffed but unavailable, the call waits in front of the "skill" calls in the skill administered as the agent's Direct Agent Skill until either the call is answered or a coverage timeout occurs. Also, the caller hears an optional direct-agent announcement followed by music or silence. There is one Direct Agent announcement per system. The agent, on the other hand, receives a ring-ping, and the current work mode button flashes. If the agent is available, the call is delivered to the agent according to the answering and alerting options. Calls are answered and handled in the same manner as ACD calls. See the Feature-Related System-Parameters form in the *DEFINITY Enterprise Communications Server Release 6 Administration and Feature Description* (555-230-522) for more information.

- **Login ID Name on the Voice Terminal Display**

  A call to a logged-in EAS Login ID by default displays the name associated with the Login ID and not the name associated with the voice terminal. This is also true on the receiving party's display for a call made from a voice terminal with an agent logged in. However, the user can display the name of the physical terminal where the EAS agent is logged in. The user must be active on a call with the agent, and must have a

terminal with an alphanumeric display and an inspect button. When the inspect button is pressed during a call to or from the EAS agent, the physical terminal name of the agent is displayed.

- **Coverage Path**

  Call coverage can occur whether or not the agent is logged in. If the agent is not logged in, the busy criteria is met and the call follows the points on the coverage path. If the agent is logged in but fails to answer, the don't answer criteria is met and the call follows the points on the coverage path. A call to the Login ID goes to the coverage path assigned to the Login ID rather than to the coverage path assigned to the voice terminal extension.

- **Agent Restrictions**

  A call to the Login ID or from the agent uses the restrictions associated with the agent and not the voice terminal.

Voice terminals are fully functional if an agent is not logged in. The restrictions, coverage, and name revert to the voice terminal administration when the agent logs out.

If a number of users are sharing one voice terminal (due to job sharing or shifts, for example), a unique Login ID extension is assigned to each user. Therefore, whenever a user is logged out, any calls to that user (login ID) are sent to his or her coverage path. As a result, Login IDs can be used to reach people independent of where they happen to be. Such people include those who use more than one phone because they have more than one office or (in the case of security guards, for example) sit at more than one desk.

Because AAS/AUDIX ports are not mobile, these ports are administered to Agent Login IDs. Whenever the AAS or AUDIX field is set to "y," a field requesting the port number is brought up, and the password field disappears.

Interactions Involving EAS

This section discusses the feature and adjunct interactions involving EAS.

Feature Interactions

This section discusses the feature interactions involving EAS. Unless otherwise specified, the feature interactions for skill hunt groups are the same as for vector-controlled splits.

- Abbreviated Dialing

  Abbreviated dialing can be used to log in or log out EAS agents. Abbreviated dialing lists or buttons can be administered only for stations.

- Administration Without Hardware

  Although EAS Login IDs are extensions without hardware, they are not a part of the Administration without Hardware feature.

- Agents in Multiple Splits Feature

  With EAS, the “Agents in Multiple Splits” feature is called “Agents in Multiple Skills.” This feature allows an EAS agent to be logged into multiple skills.

- Agent Work Modes

  With EAS optioned, an agent can be in only a single work mode for all skills at any one time (for example, an agent cannot be in AUX work mode in one skill hunt group and also available in another skill hunt group). Also, if the After Call Work (ACW) mode button is selected, the agent is placed into ACW for the first skill administered and logged into.

- Assist

  This feature is used for skill hunt groups (that is, there is one supervisor per skill hunt group). A voice terminal can be administered with one or more Assist buttons for each skill that agents using the terminal might have. An Assist button can also be administered with no associated skill. In this case, the supervisor for the skill that the agent is currently active on is called. If the agent is not active on any skill, the supervisor for the agent’s first skill is called.

  Any assist button selected is tracked as an assist for the current call, regardless of the skill assigned (if any) to the button. The administered association of an Assist button with a particular skill and assigned supervisor is not affected when an EAS agent logs into that station.

- Audible Message Waiting

  If messages are waiting for an EAS agent login-ID extension, an agent will hear a special 5-burst EAS Login-ID Message Waiting tone (instead of confirmation tone) after successfully logging in. This does not require Audible Message Waiting to be assigned to the voice terminal or the system.

  If Audible Message Waiting is optioned for the system and assigned to an agent’s voice terminal, and messages are waiting for the agent login ID extension, the agent will hear the Audible Message Waiting tone whenever the agent goes off-hook, or selects a line appearance and hears dial tone. Messages waiting for the physical extension will not cause an Audible Message Waiting tone when an EAS agent is logged in.

- Auto-Available Skills

  If a skill hunt group is administered as an Auto-Available Skill (AAS), the EAS Login IDs assigned to this skill must also be administered as Auto-Available. When the switch reinitializes, these Login IDs are automatically logged in with the AUTO-IN work mode. If any switch

features attempt to change the work mode to anything except AUTO-IN, this attempt is denied. Agents cannot have both Auto-Available and Non-Auto-Available Skills. This feature is not intended for human agents.

- Automatic Answering with Zip Tone

  This feature can be administered only for a physical extension. The feature is not associated with a LoginID.

- Basic Call Management System (BCMS)

  The BCMS user interface remains the same when EAS is optioned. The only change is that the labeling of the headings is changed from split to skill. When EAS is enabled, BCMS agent reports are based on the Agent Login IDs.

  BCMS tracks Direct Agent calls as skill calls. Direct Agent calls affect ACD talk time, ACW time, and Average Speed of Answer. Whenever Direct Agent calls are waiting, BCMS displays an asterisk immediately after the CALLS WAITING column.

- Best Service Routing™ (BSR)

  EAS VDN skills (1st, 2nd, 3rd) can be used in *consider split/skill* commands. EAS skills levels are used for the EAD-MIA and EAD-LOA BSR Available Agent Strategies.

- Bridging

  ACD calls do not alert on bridged appearances. However, bridged users can activate features on behalf of agents. Features that can be activated include: log in, log out, change work modes, and assist.

- Call Coverage

  Call coverage can occur whether or not the agent is logged in. If the agent is not logged in, the busy criteria is met and the call follows the points on the coverage path. If the agent is logged in but fails to answer, the don't answer criteria is met and the call follows the points on the coverage path. A call to the Login ID goes to the coverage path assigned to the Login ID rather than to the coverage path assigned to the voice terminal extension.

- Call Detail Recording (CDR)

  For skill calls, the called party field can optionally be the Agent Login ID.

- Call Forwarding

  Skill hunt groups (since they are vector-controlled) cannot be call forwarded. EAS agent Login IDs cannot be forwarded, but the physical extension where the EAS agent is logged in can be forwarded. If another station (with console permissions) tries to forward an EAS Login ID, an intercept is given.

- Call Park

  To retrieve a parked call via a Feature Access Code (FAC), the agent dials the Answer-Back FAC and the extension where the call is parked. If the person un-parking the call dials the Answer-Back FAC and the physical extension of the station where the call is parked, they are connected to the parked call.

  In some cases, the person un-parking the call may also be able to dial the Answer-Back FAC and the logical agent extension of the agent who parked the call. This operation is possible if the Class of Restriction (COR) of both the agent parking the call and the terminal or agent un-parking the call have a COR with the Direct Agent Calling flag set to “y.” If the terminal un-parking the call is not a logged in agent, then the terminal must have a COR with Direct Agent Calling set to “y.” If the station un-parking the call is a logged in agent, then the COR of the logical agent extension must have Direct Agent Calling set to “y.”

- Call Pickup

  Skill hunt group extensions and EAS Login ID extensions cannot be members of a Call Pickup group.

- Class of Restriction

  Skill hunt groups do have a Class of Restriction (COR). The COR is used if the skill hunt group extension is called directly. The COR for an EAS agent Login ID overrides the physical extension’s COR of the voice terminal that an agent logged into.

- Class of Service

  EAS agents do not have a COS associated with their Login ID. Instead, the COS is associated with the physical extension. Therefore, the COS of the voice terminal is not affected when an EAS agent logs into that terminal.

- Dial Plan

  Agent Login IDs are part of the dial plan, and they reduce the total number of stations.

- Direct Agent Calling

  If a called EAS Agent Login ID and the call originator (extension, trunk, or VDN) both have a COR that allows Direct Agent calls, the call to the Login ID is treated as a Direct Agent call. A call to the voice terminal extension where an EAS agent is logged in, or a call to an EAS Agent Login ID where either the originator’s or the Login ID’s COR does not allow Direct Agent calls, is treated as a personal (non-ACD) call.

- Displays—Voice Terminal

  When an EAS agent is logged in, the display for originators who call the Login ID shows the Login ID and agent name (as administered via the Agent Login ID form). Calls originated by the agent show the Agent Login ID and agent name at the receiving terminal’s display. However, the user can display the name of the physical terminal where the EAS agent is

logged in. The user must be active on a call with the agent, and must have a terminal with an alphanumeric display and an inspect button. When the inspect button is pressed during a call to or from the EAS agent, the physical terminal name of the agent is displayed.

Calls to the physical extension show the physical extension's number and name on the originator's display.

- Leave Word Calling

  When an EAS agent is logged into a station, the agent can only retrieve LWC messages left for that agent's login ID. To retrieve LWC messages left for that station, the agent must log out.

  When an EAS agent is logged into a station, its Message lamp defaults to tracking the status of LWC messages waiting for the station. However, you can assign the Message lamp to track the status of LWC messages waiting for the agent's login ID.

- Look-Ahead Interflow

  Skills are not sent to another system when a call interflows using Look-Ahead Interflow. If skills have the same meaning on both ACDs, a Look-Ahead Interflow command to a VDN with the same skills assigned can provide a mapping of the skills.

- Multiple Split Queuing

  When EAS is enabled, the Multiple Split Queuing feature is called Multiple Skill Queuing, which has the same functionality. With Multiple Split/Skill Queuing a call can queue to a maximum of 3 splits/skills.

- OCM/EAS

  If EAS is enabled on the DEFINITY switch, The Outbound Call Management (OCM)/Expert Agent Selection (EAS) feature is required for a CallVisor ASAI adjunct application to launch predictive Outbound Call Management (OCM) calls. Predictive Calling is an OCM feature that is often used in applications, such as sales or "cold-calling," where it does not matter which agent is accessed by a caller and for which it is important to keep the agents utilized fully.

  While OCM predictive calling is an outbound call management application, the EAS environment provides a number of desirable features for inbound call handling. The OCM/EAS feature allows the customer to enable both types of call handling on the switch. From a technical standpoint, if EAS is enabled, the feature is needed for the following reasons:

  — All skill hunt groups are vector-controlled. (However, to launch a predictive OCM call in a traditional ACD environment, the ACD split cannot be vector-controlled.)

  — The traditional ACD environment and EAS cannot be enabled on the switch at the same time.

The OCM/EAS feature extends the ASAI features to include launching predictive OCM calls from a VDN extension. Previously, ASAI hosts could launch predictive calls only from ACD split extensions. A limited number of Call Vectoring commands are supported in the VDNs used to launch or process OCM predictive calls. These commands are listed in the following section.

Commands for OCM Predictive Calls

The Call Vectoring commands indicated in Table 11-14 are available for processing OCM Predictive Calls. The customer must design vectors intended for processing predictive calls in such a manner that the vectors are limited to the supported steps.

Table 11-14 lists the supported commands and provides a brief comment for each command.

Table 11-14. Commands for OCM Predictive Calls

Command	Comment
queue-to skill (single occurrence)	This command queues the call for handling by an agent in the skill pool.
announcement	This command plays an announcement if there are no agents available and if the queue (if any) is full.
stop	This command ends vector processing. The command also disconnects any call that is not queued.
adjunct routing	EAS supports adjunct routing to any of the following: skill extension, direct agent call, announcement, or local extension. The command does not involve routing to an external number.
wait-time	**NOTE:** This command is used with the *adjunct routing* step to determine how long the switch will wait for an adjunct route before continuing with vector processing.

- Queue Status Indications

  Physical extensions can be administered with Queue Status Indicator buttons and lamps for skill hunt groups that operate in the same manner as split Queue Status Indicators for traditional ACD splits. As long as enough buttons are available, Queue Status Indicators can be administered for all skills needed by agents using that physical extension. Also, any waiting Direct Agent calls are not reflected in the queue status indicators.

- Reason Codes

  With Reason Codes, an EAS agent can enter a numeric code that identifies his or her reason for entering AUX work mode, or for logging out.

- Service Observing

  Service Observing is activated in the EAS environment by dialing either the physical extension of the terminal where an EAS agent is logged in, or the EAS agent Login ID.

- Remote Service Observing

  Remote access to the Service Observing (SO) FACs can be provided via the Remote Access feature or through Service Observing vectors. See "Service Observing" on page 3-18 and "Creating Service Observing Vectors" on page 4-13 for additional information.

- VDN Override

  If VDN Override is set to y (yes) on the "previous" VDN, the VDN skills of the "current" VDN are used. If VDN Override is set to n (no) on the "previous" VDN, the VDN skills of the "previous" VDN are used.

- VuStats

  VuStats can display information for all 20 agent skills.

- Work Mode Buttons

  Only a single set of agent work mode buttons is needed. If multiple buttons are assigned, all lamps for that work mode (for example, manual-in) light whenever any one button is pushed.

Adjunct Interactions

This sections discusses the adjunct interactions involving EAS.

ASAI

ASAI support for EAS may be organized into the following categories: call control, feature requests, value queries, event notification, and adjunct-controlled skills. This section provides a high-level overview of the behavior of ASAI in the EAS environment.

Call Control

Call control capabilities work exactly the same in the EAS environment as in the traditional ACD environment except for the following:

- User-classified third party make calls (calls classified by originator) may originate from an EAS Login ID and terminate to a Login ID.
  User-classified calls terminating to a Login ID are given the same Direct Agent treatment provided for such calls that are dialed from a station extension.

- Switch-classified third party make calls (which are classified by a call classifier board and delivered (when answered) to the originating hunt group) may originate from or terminate to EAS Login IDs.
- Direct Agent third-party make calls (ACD calls terminated to a selected member of an ACD skill group) may be requested by including a Direct Agent option, an agent's physical extension and a skill group extension (compatibility mode), or by requesting a user-classified third-party make call with a Login ID destination. The primary differences between the two methods of requesting Direct Agent calls are that the compatibility mode allows the adjunct to specify the skill hunt group to which a given Direct Agent call is queued and that the non-compatibility mode allows the adjunct to direct the call to a Login ID, regardless of which station an agent is logged into. Direct Agent third-party make calls may not originate from an EAS Login ID.
- Supervisor assist third party make calls (supervisor assist calls originated by a selected member of an ACD split) may originate from an EAS Login ID, and they may terminate to an EAS Login ID. Unlike dialed Direct Agent calls, supervisor assist calls terminated to a Login ID behave as though they have been previously directed to the requested Login ID's physical extension (for example, they do not cover if the requested agent is not logged in and if the originator's display shows the agent's physical extension and not the agent's Login ID).
- Extension (Domain) control may not be requested for an EAS Login ID, but it may be requested on behalf of a Logical Agent's physical extension. Auto-dial calls (calls initiated by an extension-controlled station) may be terminated to an EAS Login ID, in which case the call is given Direct Agent treatment.
- Adjunct routing calls (vector calls routed by an ASAI adjunct via the *adjunct routing* Call Vectoring command) are similar to third party make calls. Such calls may include a Direct Agent option, an ACD agent's physical extension, and a skill extension. If this is true, these calls are given compatibility mode Direct Agent treatment and may be terminated to an EAS Login ID (in which case they behave like dialed Direct Agent calls).
- If EAS is optioned, ASAI launches OCM switch-classified or predictive calls from a VDN extension via the OCM/EAS feature. On the other hand, to launch a predictive call in a traditional ACD environment, an adjunct OCM application sends to the switch an ASAI request with an ACD split number as the "originating number." The application also sends flags identifying the call as a switch-classified call. In the traditional ACD environment, the ACD split cannot be vector-controlled.

Feature Requests

In the EAS environment, agent login, logout and change work-mode requests are fully supported. Agent login requests must contain an EAS Agent Login ID and optional password (delimited by '#') in the login request's user code IE. Agent logout requests and change work-mode requests may contain the desired

agent's physical extension or Login ID. Call Forwarding and Send all Calls feature requests are denied for EAS Login IDs but may be requested for EAS physical extensions where an EAS agent is logged in.

Multiple Monitors

Multiple Monitors provides the ability for up to three ASAI applications to monitor the same ACD Split or VDN domain.

This is not only helpful in environments were ICM is primary, it can also be used to add an OCM application to launch calls at off-peak times without disrupting the primary application in any way. Multiple Monitors can also be used to monitor an ACD split over 2 links in call environments where ASAI link failure recovery is important.

Value Queries

Value queries function identically in the EAS and traditional environments, except that the Extension Type/Class Information Query returns a new indication that a requested extension is an EAS Login ID along with an indication of whether the Login ID is currently logged in and where (in other words, at which physical extension).

Event Notification

Because all skill hunt groups are vector-controlled, event notification may not be requested on the basis of a skill hunt group extension. Event notification may, however, be requested on the basis of a controlling VDN extension. Generally, all event reports involving EAS agents contain the agent's physical extension rather than the agent's Login ID.

Adjunct-Controlled Skills

Agents with adjunct-controlled skills are considered to be adjunct-controlled agents. Adjunct-controlled agents exhibit the same behavior as agents within adjunct-controlled splits in the traditional ACD environment. The following list provides more details:

- Stations are locked for all logged-in adjunct-controlled agents. The only action an agent can take from the station is to go onhook (or unplug the headset) from an auto-answer station, which causes the agent to be logged out.
- Stations are unlocked whenever the controlling adjunct's ASAI link goes down. Stations are locked again when the adjunct's link is reestablished.
- The adjunct controls all skill/agent activities such as login, logout and change work-mode (with the exception of agent logout via onhook).
- Only adjunct-controlled calls can terminate to the extension of an adjunct-controlled agent.

- Only adjunct-controlled calls can terminate to an adjunct-controlled skill hunt group extension.
- Adjunct-controlled EAS Agents can be administered with only one skill. Accordingly, EAS agents may not mix adjunct-controlled and non-adjunct-controlled skills.

AUDIX

Calls to the EAS Agent Login ID can cover to AUDIX. Each agent must enter his or her Agent Login ID when calling AUDIX to obtain messages.

AUDIX agents are assigned to EAS agent extensions. These Login IDs are used for CMS and BCMS tracking if the associated AUDIX skill hunt group is externally measured. The “aut-msg-wt” button or message waiting light can be used to indicate that the Login ID has a message.

An agent cannot have both AUDIX and non-AUDIX skills.

CMS (R3V2 and later releases)

The following is true for R3V2 and later release CMS Agent Tables:

- Separate Direct Agent database items starting with “DA_” are tracked.
- Standard reports combine statistics for Direct Agent calls and skill calls. However, reports can be customized to separate these statistical groupings.

The following is true for R3V2 CMS Skill Tables:

- Skill queues can be monitored for Direct Agent calls on the “Queue/Agent Summary” report.
- Direct Agent calls are not tracked.
- Agent time while on a Direct Agent call is tracked as “other” time.
- Non-ACD calls while in Direct Agent ACW are tracked.

The following is true for R3V2 CMS VDN/Vector Tables:

- Direct Agent calls and skill calls are combined as ACD calls.

Speech-Processing Adjuncts

Speech-processing adjuncts which have a “line” interface to the switch are able to initiate Direct Agent calls by dialing the Login ID for an agent.

Listing Agents Logged into a Split or Skill

When administering a split or skill, you can use the **list members hunt-group** command to verify that all agents are logged out and to identify any agents who are logged in. You can list all logged in agents for a split or skill, or limit the list to a range of login IDs or physical extensions:

- **list members hunt-group <hunt group *nnn*>**
- **list members hunt-group <loginid nnnn to-loginid nnnn>**
- **list members hunt-group <ext nnnn to-ext nnnn>**

For example, to list the agents logged in to skill 37:

1. Type **list members hunt-group *37*** and press RETURN.

   The List Hunt Group Members report screen appears.

```
list members hunt-group 37

                          HUNT GROUP MEMBERS

Group Number: 37     Group Name: Platinum Card    Group Extension: 3002
  Group Type: ucd-mia       ACD? y     Skill? y           Members: 4

    Phys  Phys            Login Login          Agt     Per       Wrk
    Ext   Name            Ext   Name           Prf Lvl All SO DF Tim Occ
 1: 1002  1002-Al MacInni 2902  Agent 2902     grt 04      y      10  33
 2: 1022  1022-Kelly Chas 2901  Agent 2901     lvl 14      n      15  55
 3: 1001  1001-Chris Pron 2904  Agent 2904     pal R2         n    0   0
 4: 1021  1021-Maria Esta 2903  Agent 2903     pal 08   30    y   18  45
```

This screen shows a system using EAS and CentreVu Advocate. For systems without either of these features,the related columns will be blank.

NOTE:
You can also use this command to list the agents *administered* in non-ACD hunt groups. However, since non-ACD hunt groups don't use agent logins the report will not identify agents who are currently active.

Other Forms that Support EAS Agent LoginID

Table 11-15 indicates which DEFINITY forms can have an EAS Agent loginID administered on them.

Table 11-15. EAS LoginID Table

Feature	Accepts LoginID?
Abbreviated Dialing Buttons	
7103A	Yes
Enhanced	Yes
Group	Yes
Personal	Yes
System	Yes
Agent-LoginID	
Port Extension	No
Announcements	No
Buttons	
abrdg_app	No
aut-msg-wt	Yes
brdg_app	No
busy-ind	Yes
data_ext	No
man_msg_wt	No
q-calls	No
q-time	No
signal	No
Call Processing	
Auto-Callback	No
Call Forward from Agent Login ID	No
Call Forward to Agent Login ID	Yes
Call Park	Yes

Continued on next page

Table 11-15. EAS LoginID Table — *Continued*

Feature	Accepts LoginID?
Hundreds group	No
LWC Retriever gets lagt msgs	Yes
Service observ Agent Login ID	Yes
CDR Parameters	
Primary Extension	No
Secondary Extension	No
Code-Calling	Yes
Communication Link Form	
Communication Link Digits	No
Console Parameters	
CAS-backup ext	No
IAS Att Access Code	No
Coverage Groups	
Answer Group Member	No
Path	Yes
Measured Principals	
Coverage Measurement	No
Feature-Related Parameters	
ACA-referral dest.	No
ACA - long holding	No
ACA - short holding	No
Controlled out restriction	No
Controlled Terminal	No
Controlled Stn-to-Stn	No
DAA Extension	No
DID/Tie/ISDN announcement	No
Emergency Access Redirection	No
CDR output extension	No
SVN referral destination (announcement)	Yes

Continued on next page

Table 11-15. EAS LoginID Table — *Continued*

Feature	Accepts LoginID?
System LWC retriever	No
System Printer	No
Hospitality Parameters	
Journal Printer	No
LWC wakeup	No
PMS ext	No
PMS log	No
Routing on Voice Synthesis	No
Hunt Group Form	
Announcement extension	No
ASAI link	No
AUDIX extension	No
Calls Warning extension	No
Member	No
Night Service	No
Supervisor	Yes
Time Warning extension	No
Intercom Group Member	No
Intra-switch CDR	Yes
Listed Directory Number	
Member	No
Night Destination	Yes
Malicious Call Trace	
MCT Member	No
Permanent Switched Calls	No
Personal CO Line	No
Pickup Group Member	No
Remote Access Extension	No
Term Extension Group Member	No

Continued on next page

Table 11-15. EAS LoginID Table — *Continued*

Feature	Accepts LoginID?
Trunk Group	
Night Service	Yes
Incoming Destination	Yes
Member Night Service	Yes
Vector Administration	
adjunct extension	No
announcement	No
messaging	Yes
route-to	Yes

Upgrading to a DEFINITY ECS EAS Environment

For information about how to create a Call Center that uses EAS see "Call Vectoring/ EAS Option" on page K-11. For information on converting a Call Center to EAS, refer to Appendix L.

Call Vectoring Applications

12

Introduction

This chapter is intended to present several generic Call Vectoring applications a customer might use. Each application is based on one or more of the Call Vectoring features discussed in this guide. Vector exercises are provided at the end of the chapter.

Table 12-1 identifies the feature(s) used within each example in this chapter. The examples are numbered according to the order in which they appear within the chapter. The name of the section in which each example appears is listed first.

Table 12-1. Applications and Corresponding Feature(s)

Section Title	Example No.	Feature(s) Used
Customer Service Center	1	Basic Call Vectoring
Automated Attendant	2	Call Prompting
DIVA and Data/Message Collection	3	Call Prompting, Basic Call Vectoring
Distributed Call Centers	4	Look-Ahead Interflow, Basic Call Vectoring
Help Desk	5	Adjunct Routing, Call Prompting, Basic Call Vectoring

Continued on next page

Table 12-1. Applications and Corresponding Feature(s)

Section Title	Example No.	Feature(s) Used
Insurance Agency/Service Agency	6	Basic Call Vectoring, Call Prompting, Rolling ASA, EWT, VDN Calls, and ANI Routing
Warranty Service	7	Basic Call Vectoring, EAS
Resort Reservation Service	8	Basic Call Vectoring, Adjunct Routing, Call Prompting, EAS

Customer Service Center

Example 1 presents a scenario where a customer service center is open weekdays from 8 a.m. until 5 p.m. The center provides two separate telephone numbers. One number is for ordinary customers, while the other number is for priority customers. The following three vectors in Screen 12-1 illustrate how calls to the customer service center are handled.

```
VDN (extension=1021  name=''Customer Serv''  vector=21)
Vector 21:
      1. goto vector 29 if time-of-day is all 17:00 to all 08:00
      2. goto vector 29 if time-of-day is fri 17:00 to mon 08:00
      3. goto step 10 if calls-queued in split 1 pri l > 10
      4. queue-to split 1 pri m
      5. wait-time 10 seconds hearing ringback
      6. announcement 3521
      7. wait-time 50 seconds hearing music
      8. announcement 3522
      9. goto step 7 if unconditionally
     10. busy
VDN (extension=1022  name=''Priority Cust''  vector=22)
Vector 22:
      1. goto vector 29 if time-of-day is all 17:00 to all 08:00
      2. goto vector 29 if time-of-day is fri 17:00 to mon 08:00
      3. goto step 12 if calls-queued in split 1 pri h > 10
      4. queue-to split 1 pri h
      5. announcement 3521
      6. wait-time 10 seconds hearing music
      7. check split 2 pri h if oldest-call-wait < 20
      8. check split 3 pri h if oldest-call-wait < 20
      9. announcement 3522
     10. wait-time 60 seconds hearing music
     11. goto step 7 if unconditionally
     12. route-to number 0 with cov n if unconditionally
No VDN
Vector 29:
      1. announcement extension 3529
      2. wait-time 10 seconds hearing silence
      3. disconnect after announcement 3529
```

Screen 12-1. Example 1: Customer Service Center

First, let's assume that a priority customer places a call. In such a case, if the correct number is dialed, vector 22 is accessed. The first two steps of this vector determine if the call arrives during nonbusiness hours. If the call arrives between 5:00 p.m. and 8:00 a.m. on any given day, step 1 routes the call to Vector 29. Step 2 does the same if the call arrives during the weekend (that is, between 5:00 p.m. Friday and 8:00 a.m. Monday). If vector 29 is accessed, the caller is given the appropriate announcement twice (steps 1 and 3) and is then disconnected (step 3).

If the call is placed during business hours, step 3 of vector 22 determines if the number of calls queued in the main split exceeds 10. If so, control is sent to step 12, which routes the call to the attendant. If not, the call is queued to the main split (step 4). Thereafter, if necessary, the appropriate announcement is provided (step 5), followed by a wait period (step 6).

If the call is not answered by this time, steps 7 and 8 attempt to queue the call to a backup split (*2* and *3*, respectively). The call is queued to either split if the oldest call therein has been waiting fewer than 20 seconds. Whether or not the call is queued, steps 9 through 11 implement an announcement-wait cycle that continues until an agent answers the call, or until the caller abandons the call.

A call placed by a nonpriority customer is processed by vector 21. Vector 21 provides a treatment similar to that provided by vector 22. The three differences are that: the nonpriority customer's call is not given the chance to be queued to more than one split; the priority customer's call is given a higher priority in the queue; and the priority customer's call routes to an operator when there are too many calls queued whereas the nonpriority customer routes to a busy signal.

Automated Attendant

Example 2, Screen 12-2, illustrates Automated Attendant, which is one of the applications that can be supported by the Call Prompting feature. Automated Attendant allows the caller to enter the extension of the party the caller would like to reach. Depending upon the parameters established, the user can enter up to 16 digits from a touch-tone phone.

Automated Attendant is usually used for customers without DID trunks whose callers know the extension of the people they are calling. Because it reduces the need for "live attendants," Automated Attendant allows the customer to reduce costs.

Screen 12-2 shows an example of a vector that implements Automated Attendant:

```
1. wait-time 0 seconds hearing ringback
2. collect 5 digits after announcement 30001
   (''You have reached Ridel Publications in Greenbrook.
   Please dial a 5-digit extension or wait for the
   attendant.'')
3. route-to digits with coverage y
4. route-to number 0 with cov n if unconditionally
5. stop
```

Screen 12-2. Example 2: Automated Attendant

Step 1 of this vector contains the *wait-time* command, which is placed before the *collect digits* command in step 2 to provide the caller with ringback in the event that a TTR is not immediately available. (Recall that a TTR must be connected in order for the *collect digits* command to take effect.) Once a TTR is connected, the caller is prompted to enter the destination extension of the party he or she would like to reach (step 2). The *collect digits* command in step 2 collects the digits. Thereafter, the *route-to digits* command in step 3 attempts to route the call to the destination.

If the *route-to digits* command fails (because the caller fails to enter any digits, or because the digits entered do not comprise a valid extension), the *route-to number* command in step 4 routes the call to the attendant. However, as long as the destination is a valid extension, the *route-to digits* command succeeds, coverage applies, and vector processing terminates. (Even if the destination is

busy, vector processing terminates because coverage call processing takes effect.)

DIVA and Data/Message Collection

Example 3 involves a mutual fund company that is open 24 hours a day, seven days a week. All incoming calls are directed to a single VDN extension that maps to a main vector. The main vector presents a menu of options to the calling party, and it uses Call Prompting to determine the desired service. Three services are offered, and they are identified and described as follows:

- *New accounts* enables the customer to open a new account.
- *Account inquiries* enables the customer to make inquiries concerning his or her account.
- *Net asset values* enables the customer to hear information concerning the net asset values of company's funds.

If the caller selects "account inquiries," he or she is prompted to input his or her account number before being answered by an agent. The agent can display this number via use of the CALLR-INFO button, if the button is available and needed.

NOTE:
If the agent has two-line display supported by the system, the account number is automatically displayed on the second line. The supported display terminals include: 7404, 7407, 7444, 8434 and CALLMASTER.

NOTE:
CALLMASTER sets are unable to display CALLR-INFO information for a ringing call. However, this information is displayed once the call is answered.

This example uses three other applications that can be supported by the Call Prompting feature. These applications are described as follows:

- **Data In/Voice Answer (DIVA)** allows a caller to receive information on a topic selected at the prompt. The caller selects the desired topic by entering the appropriate digit(s).
- **Data Collection** provides a method of collecting digits from a caller. The digits requested comprise an official number of some sort (for example, Social Security Number), and they help the Call Center process the call more efficiently.
- **Message Collection** allows the caller to leave a recorded message in lieu of waiting for the call to be answered.

The following four vectors in Screen 12-3 illustrate how the mutual fund company handles telephone calls.

```
VDN (extension=1030  name=''ABC Inv''  vector=30  display override=''y'')
Vector 30
      1. wait-time 0 secs hearing ringback
      2. collect 1 digits after announcement 3531
         (''Thank you for calling ABC Investments. If
         you wish to open a new account, please dial 1.  If
         you wish to make an account inquiry, please dial 2.
         If you wish to know the current net asset values of
         our funds, please dial 3.'')
      3. route-to number 1031 with cov y if digit = 1
      4. route-to number 1032 with cov y if digit = 2
      5. route-to number 1033 with cov y if digit = 3
      6. route-to number 0 with cov n if unconditionally
      7. disconnect after announcement none
VDN (extension=1031  name=''New Account''  vector=31)
Vector 31
      1. queue-to split 1 pri t
      2. announcement 3535
      3. wait-time 10 secs hearing music
      4. collect 1 digits after announcement 4020
         (''We're sorry.  All of our operators are busy at
         the moment.  If you'd like to leave your name and
         telephone number so that we can get back to you,
         dial 1.'')
      5. goto step 9 if digit = 1
      6. announcement 3537
      7. wait time 50 secs hearing music
      8. goto step 6 if unconditionally
      9. messaging split 5 for extension 4000
     10. announcement 3538 (''We're sorry, we cannot take
         your message at this time.  You may continue to hold, or
         you can call back later.'')
     11. goto step 6 if unconditionally
VDN (extension=1032  name=''Account Enq''  vector=32)
Vector 32:
      1. wait-time 0 secs hearing ringback
      2. collect 6 digits after announcement 3533
         (''Please enter your 6-digit account number.'')
      3. queue-to split 1 pri m
      4. announcement 3535
      5. wait-time 60 secs hearing music
      6. goto step 4 if unconditionally
VDN (extension=1033  Name=''Net Asset Val''  Vector=33)
Vector 33:
      1. disconnect after announcement 3534
         (''The net asset values of our funds at the close
         of the market on Wednesday, May 15 were as follows:
         ABC Growth.....33.21.....up 33 cents; ABC
         High Yield.....11.48.....down 3 cents.'')
```

Screen 12-3. Example 3: DIVA and Data/Message Collection

When the call is placed, vector processing begins in vector 30, which is the main vector. Step 1 of the vector contains the *wait-time* command, which is placed before the *collect digits* command in step 2 to provide the caller with feedback in the event a TTR is not immediately available. Once a TTR is connected, the *collect digits* command provides an announcement requesting the caller to enter 1, 2, or 3, depending upon the service desired. If the caller enters a digit other

than one of the three mentioned, or if the caller fails to enter any digits within 10 seconds, the command fails, and the call is routed to the attendant (step 6). On the other hand, if the caller enters 1, 2, or 3 within 10 seconds, the call is routed to the vector specified in the appropriate *route-to number* command, which appears in steps 3, 4, and 5.

Let's say that, when prompted, the caller enters 3 because he or she wants to learn about the net asset values of the company's funds. In such a case, the *route-to number* commands in step 3 and in step 4 fail because, in each case, the digit tested for in the condition portion of the command is not 3. However, the *route-to number* command in step 5 succeeds because the digit tested for matches the one entered by the caller. Accordingly, the call is routed to VDN extension 1033, and vector processing continues in vector 33.

The *announcement* command in step 1 of vector 33 provides the caller with the information on net asset values and then disconnects the call.

The process just described, whereby the caller receives information as a result of making a request at the prompt, is an example of the *Data In/Voice Answer (DIVA)* application.

Returning to the main vector, suppose another caller wants to make an enquiry into his or her account and the caller enters 2 when prompted. In such a case, step 3 fails, but step 4 succeeds. Accordingly, the call is routed to VDN extension 1032, and vector processing continues in vector 32.

The *collect digits* command in step 1 of vector 32 first requests the caller to enter his or her 6-digit account number. The command then collects the digits entered by the caller. Whether or not the caller correctly enters the digits, the *queue-to split* command in step 2 queues the call. If an agent does not immediately answer the call, the standard announcement is provided in step 3 and, if necessary, a delay is provided in step 4. The *goto step* command in step 5 returns call control back to step 3, thus ensuring that the announcement-wait cycle will continue until the agent answers the call, or until the caller abandons the call.

The process just described, whereby the caller, when prompted, enters digits that comprise an official number (an account number, in this case), is an example of the Data Collection application. If the agent has a CALLR-INFO button or a two-line display, the agent can see the digits entered by the caller. As a result, the agent need not request the account number from the caller.

Finally, suppose a third caller wants to open an account and that he or she enters 1 when prompted in the main vector. In such a case, step 3 of the main vector is successful. Accordingly, the call is routed to VDN extension 1031, and vector processing continues in vector 31.

In step 1 of vector 31, the call is queued to the main split. Thereafter, if necessary, step 2 provides the appropriate announcement, and step 3 provides a delay period. The announcement in step 4 provides the caller with the option of

leaving a recorded message for the mutual fund company instead of having his or her call wait in queue. (This is an example of the Message Collection application.) The caller is instructed to enter 1 if he or she wishes to leave a recorded message. If the caller does not enter 1, the goto step command in step 5 fails, and an announcement-wait cycle is implemented by steps 6, 7, and 8 until the call is answered or abandoned. If the caller does enter 1 within 10 seconds, step 5 passes control to step 9. The messaging split command in step 9 attempts to connect the caller to an AUDIX or Message Center split so that the caller can leave a message. If the connection is made, the caller first hears ringback and can then leave a message. If the connection is not made, the step is unsuccessful, and step 10 provides an announcement indicating that a connection could not be made. Thereafter, the goto step command in step 11 sends call control back to step 6, which is the first step of the aforementioned announcement-wait cycle.

Distributed Call Centers

Example 4 involves two distributed call centers, one in New York and the other in Denver. Calls to the New York call center are queued to up to two splits. If calls remain unanswered for a period of time, a Look-Ahead Interflow call attempt is made to the Denver call center. If there are 10 or fewer queued calls in Denver, the Look-Ahead call attempt is accepted and serviced there. Otherwise, the call is denied and remains in queue in New York until an agent becomes available. The following two vectors in Screen 12-4 illustrate this procedure.

NOTE:
For other examples of Look-Ahead Interflow, see Chapter 7, "Look-Ahead Interflow". To learn how to integrate distributed call centers using multi-site Best Service Routing™, see Chapter 8, "Best Service Routing™"

```
SENDING SWITCH:
VDN (extension=1080  name=‘‘New York Office’’  vector=80)
Vector 80:
       1. goto step 11 if calls-queued in split 1 pri m > 5
       2. queue-to split 1 pri m
       3. announcement 3580 (‘‘All of our agents
          are busy.  Please hold and you will be answered
          by the first available agent.’’)
       4. wait-time 6 seconds hearing music
       5. route-to number 913035661081 with cov n if unconditionally
       6. check split 2 pri m if calls-queued < 5
       7. wait-time 6 seconds hearing music
       8. announcement 3581 (‘‘All of our agents
          are still busy.  Please hold and you will be
          serviced by the first available agent.’’)
       9. wait-time 60 seconds hearing music
      10. goto step 5 if unconditionally
      11. busy
RECEIVING SWITCH:
VDN (extension=1081 Name=‘‘Denver Inflow’’ Vector=81)
Vector 81:
       1. goto step 7 if calls-queued in split 3 pri l > 10
       2. wait-time 0 seconds hearing music
       3. queue-to split 3 pri h
       4. announcement 3582 (‘‘We apologize
          for the delay.  Please hold and you will be
          serviced by the first available agent.’’)
       5. wait-time 60 seconds hearing music
       6. goto step 5 if unconditionally
       7. disconnect after announcement none
```

Screen 12-4. Example 4: Distributed Call Centers

In this example, vector 80 is on the sending switch from a call center in New York, while vector 81 is on the receiving switch at a call center in Denver.

In the sending switch, the call is queued to split 1 at a medium priority (step 2) if the condition in step 1 is met. If the condition is not met, the call is routed to busy in step 11.

If the call is queued but not immediately answered, an announcement (step 3) and music (step 4) are provided. If the call is still not answered at this point, step 5 places a Look-Ahead Interflow call attempt to the receiving switch, on which vector 81 resides.

Step 1 in the receiving switch determines whether the call can be serviced in Denver. If the number of calls queued at any priority in split 3 is greater than 10, vector 81 cannot service the call. In such a case, control is passed to step 7, which rejects the Lookahead Interflow call attempt. However, if the test in step 1 succeeds, the call is queued by the receiving switch in split 3 at a high priority (step 3). This results in the Lookahead Interflow call attempt being accepted. Accordingly, the call is removed from the main split queue in New York, and

control is passed to the Denver switch, where vector processing continues at step 4.

If the receiving switch does not accept the Look-Ahead Interflow call attempt, control is passed to step 6 of the sending vector. This step then queues the call to split 2 at a medium priority, provided that there are fewer than five calls queued in that split. Thereafter, the customary announcement-wait sequence is implemented (steps 7, 8, and 9). Finally, if necessary, step 10 sends control back to step 5, which makes another Look-Ahead Interflow attempt, and the cycle is repeated.

NOTE:
To avoid confusing the caller, the treatment provided at the receiving switch should be consistent with the treatment provided at the sending switch. In Example 4, note that the caller hears music (and never ringback or silence) at the sending switch. Accordingly, music should be (and, in our example, is) featured at the receiving switch.

Help Desk

Example 5 involves a help desk at a computer firm. The help desk is configured into three groups. One group handles hardware problems, the second group handles software problems, and the third group handles general problems. For this application, the information provided in the ASAI Route request (that is, calling party number, called number, collected digits) is used to route the call to the most appropriate agent. Such an agent might be the one that last serviced the caller, or it might be the next available agent for the specific caller. Also, based on switch traffic conditions and the caller-entered digit, the call can be diverted to other destinations, such as other ACD splits, announcements, or switches.

The following vector, Screen 12-5, illustrates the help desk application:

```
1. collect 1 digits after announcement 4704
   (''Welcome to the TidyBits Computer Corporation help desk.
   If you have a question about hardware, please dial 1.
   If you have a question about software, please dial 2.
   If you have a general question, please dial 3.'')
2. adjunct routing link 2400
3. wait-time 4 seconds hearing ringback
4. route-to number 3710 with cov y if digit = 1
5. route-to number 3720 with cov y if digit = 2
6. route-to number 3730 with cov y if digit = 3
7. route-to number 0 with cov n if unconditionally
8. stop
```

Screen 12-5. Example 5: Help Desk

In step 1 of this vector, the caller is instructed to enter 1, 2, or 3, depending upon the service (hardware, software, general) he or she desires. Thereafter, the *adjunct routing* command in step 2 instructs the switch to send a Route request to the adjunct processor, which is connected to extension 2400. The Route request contains the called party number, the calling party number, and the digit collected in step 1, along with the other pertinent information for adjunct routing (see Chapter 10). If one of these digits is not entered, and if the adjunct does not return a route, the call is eventually routed to the attendant (step 7).

If the *adjunct routing* command in step 2 succeeds, the adjunct uses the information included in the Route request to select the appropriate route for the call. Let's assume the caller enters 1 and the *adjunct routing* command *succeeds*. In such a case, if the caller is judged to be a "prime" hardware customer, the call might be routed to one of a handful of specific agents who are assigned to handle such customers. On the other hand, if the caller is judged to be a "casual" hardware customer, the call might be routed to a larger group of ACD agents before being queued, or to an appropriate announcement.

Finally, let's assume that the caller enters 1 and that the *adjunct routing* command fails. In such a case, the call is routed by the *route-to number* command in step 4, probably to a vector that queues the call or provides an appropriate announcement.

Insurance Agency/Service Agency

Example 6 is an insurance company Call Center. It handles calls from: independent field agents; policy holders with claims; policy holders needing customer service; and several general service agency type 800 number client accounts. Each of the different types of calls has its own 800 number that routes the calls to associated VDNs. The following list describes the Call Center requirements.

- The independent field agents require fast service. They call the company to find out the latest rates for specific clients, to setup policies, to make adjustments, and so on. Often their clients are waiting as they call. Therefore the insurance company wants to maintain an Average Speed of Answer (rolling-ASA) of 30 seconds or less for field agent calls. These are the most important calls and are given high priority in queues.
- The calls to claims must be separated by area code. The claims agents receive different training based on the area of the country for the claim. A particular group of agents can be given training for more than one area code. Therefore, area codes do not need to be tested individually and can be grouped in Vector Routing Tables.
- The insurance company wants to give customer service callers an announcement indicating how long they can expect to wait for service.
- The insurance agency is also selling spare call center capacity to client accounts. The account contracts are provided on the basis that only so many calls to a particular account will be accepted at any given time.

In this example, rolling ASA Routing is used to maintain the rolling ASA objective of 30 seconds or less for field agent calls. ANI Routing is used to partition calls based on area code and route the call to the appropriate claims agents. EWT Routing is used to notify customer service callers of their expected wait time if it is longer than 60 seconds. VDN Calls Routing is used to regulate the number of calls to service agency clients.

Table 12-2 shows the VDNs and vectors associated with each type of call.

Table 12-2. VDN Table for Insurance/Service Agency

Type of Service	VDN Number	Vector Number
Field Agents	1001	1
Claims	1002	2
Customer Service	1003	3
Client 1	1004	4
Client 2	1005	5

NOTE:
To clearly demonstrate the features described in this example, the sample vectors do not include tests for unstaffed or full queues, out of hours operation and so on.

The following vector Screen 12-6 could be used to maintain a rolling ASA of 30 seconds for field agent calls.

```
VDN 1001 -- Field Agent Calls

1. queue-to split 10 pri h
2. goto step 6 if rolling-asa for split 10 <= 30
3. check split 11 pri h if rolling-asa <= 30
4. check split 12 pri h if rolling-asa <= 30
5. check split 13 pri h if rolling-asa <= 30
6. announcement 10000
7. wait-time 40 secs hearing music
8. goto step 3 if unconditionally
```

Screen 12-6. Example 6: Field Agent Vector

Step 1 queues the call to the main split. If the main split is currently answering calls within the target time of 30 seconds step 2 bypasses all of the backup splits and goes directly to the announcement in step 6. The assumption is that the call will be handled by split 10 within the time constraints. However, if the call is not

answered by the time vector processing reaches step 8, the backup splits are checked at that time.

If the rolling ASA for the main split is greater than 30 seconds, steps 3, 4, and 5 check backup splits. The call is queued to any of these splits that have a rolling ASA of 30 seconds or less. If the call still is not answered by the time vector processing reaches step 8, then the backup splits are checked again.

The following vector Screen 12-7 could be used to route claims calls by area code.

```
VDN 1002 -- Claims Calls

 1. goto step 10 if ani = none
 2. goto vector 21 if ani = 201+
 3. goto vector 22 if ani = 212+
 4. goto vector 23 if ani in table 1
 5. goto vector 24 if ani in table 2
 6. goto vector 25 if ani in table 3
 7. goto vector 26 if ani in table 4
 8. goto vector 27 if ani in table 5
 9. goto vector 30 if unconditionally
10. wait-time 0 seconds hearing ringback
11. collect 3 digits after announcement 10001 (“Please dial your area
      code”)
12. goto vector 30 if digits = none
13. goto vector 21 if digits = 201+
14. goto vector 22 if digits = 212+
15. goto vector 23 if digits in table 1
16. goto vector 24 if digits in table 2
17. goto vector 25 if digits in table 3
18. goto vector 26 if digits in table 4
19. goto vector 27 if digits in table 5
20. goto vector 30 if unconditionally
```

Screen 12-7. Example 6: Claims Vector

Each Vector Routing Table referred to in Screen 12-7 contains a list of area codes with the “+” wildcard. Each list of area codes is handled by a specific group of agents. Vectors 21 through 27 queue calls to the appropriate group of agents. Vector 30 provides a live agent to screen calls that have area codes not listed in any table or vector step. It also provides access to an agent when ANI is not available and the caller has not entered an area code when prompted.

The following vector Screen 12-8 will notify customer service callers of their expected wait time unless they will not have long to wait

```
VDN 1003 -- Customer Service Calls

 1. goto step 10 if expected-wait for split 32 pri l > 600
 2. queue-to split 32 pri l
 3. wait-time 20 seconds hearing ringback
 4. goto step 8 if expected-wait for call > 40
 5. announcement 1100
 6. wait-time 40 seconds hearing music
 7. goto step 5 if unconditionally
 8. converse-on split 80 pri l passing wait and none
 9. goto step 5 if unconditionally
10. disconnect after announcement 1400
```

Screen 12-8. Example 6: Customer Service Vector

In step 1, callers who would wait more than 10 minutes are routed to a “call back later announcement.” step 4 routes callers to a Conversant VRU to be given the expected wait time announcement while they hold their place in queue.

The following vectors Screen 12-9 can be used to regulate the number of calls to service agency clients. In this example, Client 1 has contracted for 100 simultaneous calls while client 2 has contracted for only 50 simultaneous calls.

```
VDN 1004-- Client 1 Calls

1. goto step 3 if counted-calls to vdn 1004 <= 100
2. busy
3. queue-to split 60 pri l
4. wait-time 20 seconds hearing ringback
5. announcement 12000
6. wait-time 60 seconds hearing music
7. goto step 5 unconditionally


VDN 1005 -- Client 2 Calls
1. goto step 3 if counted-calls to vdn 1005 <= 50
2. busy
3. queue-to split 60 pri l
4. wait-time 20 seconds hearing ringback
5. announcement 12000
6. wait-time 60 seconds hearing music
7. goto step 5 unconditionally
```

Screen 12-9. Example 6: Service Agency Clients Vectors

In both vectors the first step routes calls to queue if the number of contracted calls is not exceeded. Otherwise callers receive busy signal.

Warranty Service (with EAS)

Example 7 deals with a major appliance company that offers one year warranties and extended warranties on its major appliances (dishwashers, refrigerators, washers, and dryers). The warranties are printed in English and Spanish to accommodate customers who speak and understand these languages. Naturally, callers need to speak with someone who is familiar with the appliances they have bought and who speaks the appropriate language. Accordingly, 800 numbers are provided for calling English-speaking agents as well as Spanish-speaking agents. Bilingual agents with Spanish-speaking skills are hired so that they can back up the groups of English-speaking agents. Agents are trained first on all appliance models of a certain type and then on all appliance models for a room (such as the kitchen, the laundry room, etc.).

The Skills shown in Table 12-3 are needed for the warranty service call center:

Table 12-3. Skill Table for the Warranty Service Call Center

Appliance Type	English Skill #	Spanish Skill #
Kitchen Appliances	10	20
Dishwashers	11	21
Refrigerators	12	22
Laundry Appliances	30	40
Washers	31	41
Dryers	32	42
Supervisors		100

The VDN Skill Preferences are set up as in Table 12-4.

Table 12-4. VDN Skill Table for the Warranty Service Call Center

VDN - Skill Preferences							
English				Spanish			
Dish-washer	Refrig	Washer	Dryer	Dish-washer	Refrig	Washer	Dryer
VDN: 1100	VDN: 1101	VDN: 1102	VDN: 1103	VDN: 1200	VDN: 1201	VDN: 1202	VDN: 1203

Continued on next page

Table 12-4. VDN Skill Table for the Warranty Service Call Center

1st:11	1st:12	1st:31	1st:32	1st:21	1st:22	1st:41	1st:42
2nd:10	2nd:10	2nd:30	2nd:30	2nd:20	2nd:20	2nd:40	2nd:40
3rd:20	3rd:20	3rd:40	3rd:40	3rd:	3rd:	3rd:	3rd:

The agent skills are set up as in Table 12-5.

Table 12-5. Agent Skills for the Warranty Service Call Center

	Agent Skills			
Agent	**Skill Level 1**		**Skill Level 2**	
Jan	42	40	41	30
Judy	100	--	--	--
Sam	31	--	--	--
Sue	32	--	30	--

Once skills are assigned to VDNs and to agents, calls are directed to the appropriate vector.

The goal of the warranty service call center is to answer 80 percent of the incoming calls within 20 seconds. Accordingly, if a call directed to a vector is not answered by the time the announcement finishes, a second group of agents is viewed, thus enlarging the agent pool. If the call is not answered within the following 10 seconds, a third group of agents is viewed.

Since the call center has only a few bilingual agents, the center's management wants to reserve these agents for Spanish-speaking callers. This can be done by giving Spanish-speaking callers a higher priority in the vector or by assigning a higher skill level to Spanish skills. Also, if a Spanish-speaking caller waits more than 30 seconds for service, a supervisor of the Spanish-speaking skills takes the calls.

Figure 12-1 and Figure 12-2 illustrate the setup for the warranty service call service. Specifically, the figures show the vectors and call flows for callers with a broken washer or dryer. Separate vectors are used to provide an announcement in Spanish and in English (see step 2). The same two vectors can be used for callers with broken dishwashers and refrigerators.

Figure 12-1 shows how the call comes into the network and is then directed to the appropriate VDN, which in turn points to the appropriate vector. For each VDN, the corresponding VDN skills are indicated.

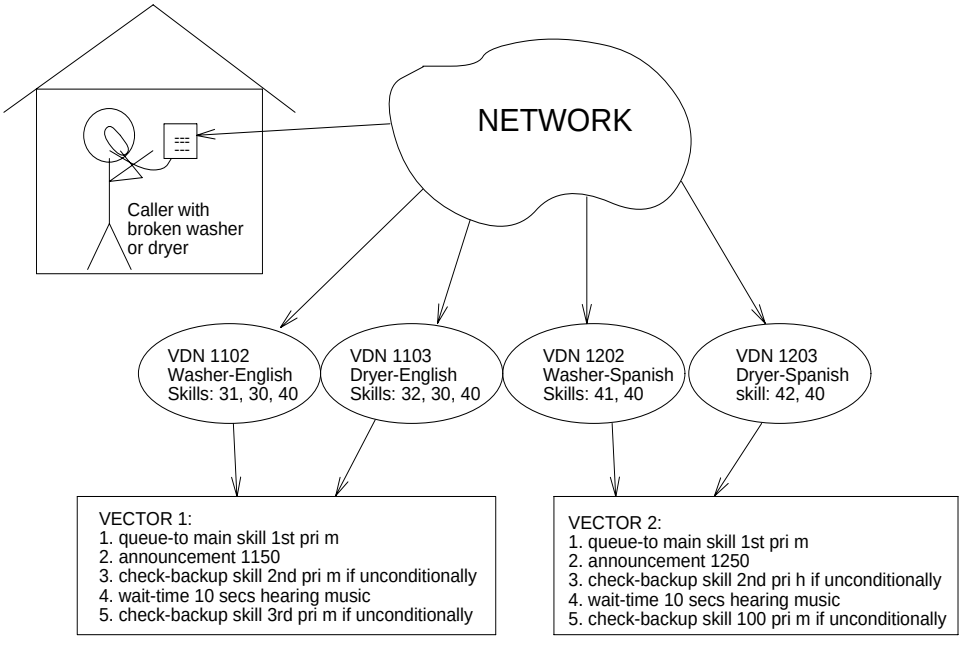


Figure 12-1. Example 6: Warranty Service Call Center (Part 1)

Figure 12-2 shows how the vector-processed call is directed to the appropriate call queue. Figure 12-2 also shows how the call is directed to the appropriate agent(s). The agent skills are indicated below each agent's name. Dashed lines indicate backup or secondary skills.

NOTE:
Only a small sample of agents is shown in Figure 12-2.

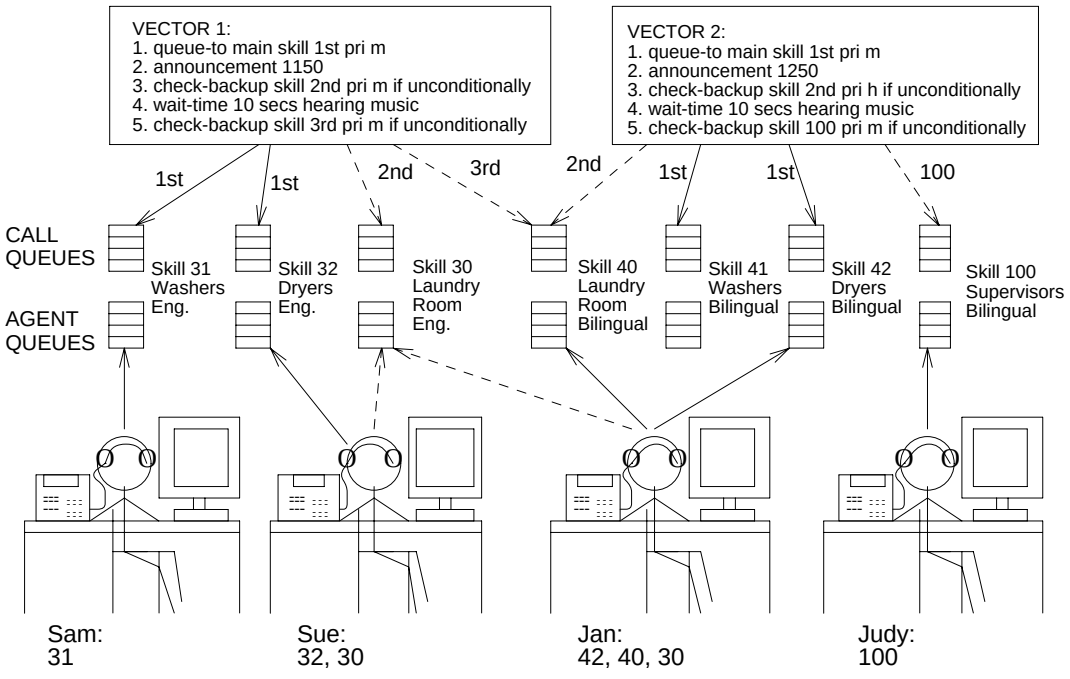


Figure 12-2. Example 6: Warranty Service Call Center (Part 2)

Let's assume that a Spanish-speaking caller has a broken dryer and decides to call the warranty service call center. In such a case, the caller dials the appropriate number. The call then enters the network (switch) and is directed to VDN 1203, which points to Vector 2. As illustrated earlier, VDN skill preferences 42 (dryers) and 40 (laundry appliances) are administered as the 1st and 2nd skill preferences, respectively, for VDN 1203. (Recall also that these preferences are actually subpreferences under the skill "Spanish.")

Once vector processing starts, the *queue-to skill* command in step 1 of Vector 2 queues the call to the skill group corresponding to the first VDN skill (42-Dryers Bilingual). If an agent with skill 42 (Jan, for example) is available, this agent answers the call. If such an agent is not available, the appropriate delay announcement in step 2 is played. Thereafter, the *check skill* command in step 3 attempts to queue the call to the skill group corresponding to the 2nd VDN skill (40-Laundry Appliances Bilingual). Accordingly, if an agent with skill 40 is available (Jan, for example), that particular agent answers the call. Otherwise, as a last resort, a wait period is provided in step 4, and the *check skill* command in step 5 checks the "specific" skill (100-Supervisors Bilingual) for available agents.

Resort Reservation Service (with EAS)

Example 8 deals with a resort company that places a variety of advertisements in magazines for information on a particular resort or state. Callers responding to these advertisements can dial one of several numbers provided in the advertisement. A call center makes the reservations for the resort company. To satisfy one request voiced by many callers to the service, an effort is made to have callers connected to an agent who has visited the resort they are interested in visiting. Also, the resort company has determined it is easier to sell additional sightseeing packages if the agent has a regional accent.

Placing the Reservation

To respond to an advertisement, the caller can dial a number that directly routes him or her to a VDN for that state's resorts. As an alternative, the caller can dial the general number for the resort chain and be serviced via Call Prompting. The following sections discuss these methods.

Specific Number Dialing

The call center is set up in such a way that a VDN with an accompanying set of VDN Skill Preferences is assigned to each state that has a resort. For example, Table 12-6 shows how Skill Preferences are assigned to Texas VDN 3222:

Table 12-6. VDN 3222 Skill Preferences Assignments for the Resort Reservation Service

Texas VDN 3222 - Skill Preferences		
1st:	30	Agent who has a Texas accent and has visited resorts in Texas
2nd:	31	Agent who has visited resorts in Texas
3rd:	130	Any agent who can take a reservation

Figure 12-3 shows how a call to VDN 3222 can be processed by Call Vectoring:

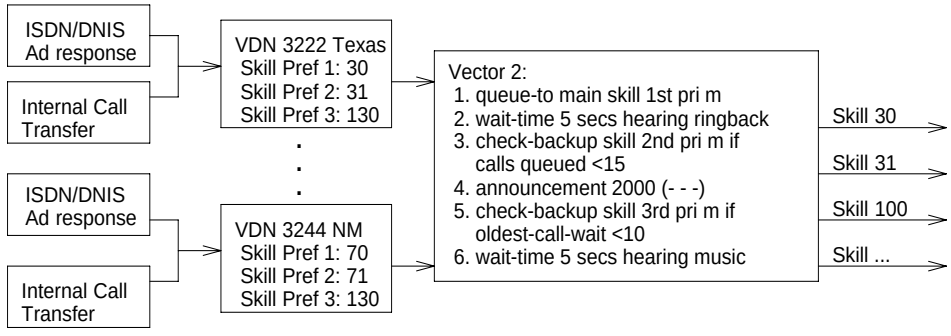


Figure 12-3. Example 7A: Process Involving Specific Number Dialing

For this process, a single VDN for each state is assigned to Vector 2. Accordingly, Figure 12-3 shows the VDN and the associated VDN skills for two states: Texas and New Mexico.

Let's assume that a caller would like information on resorts in Texas and dials the appropriate number (for example, 615-3222). In such a case, the call enters the switch and is directed to VDN 3222, which points to Vector 2.

Once vector processing starts, the *queue-to skill* command in step 1 queues the call to the skill group corresponding to the 1st VDN skill (30-Agent with Texas accent who has visited resorts in Texas). If an agent with skill 30 is available, this agent answers the call. If such an agent is not available, the *check skill* command in step 3 attempts to queue the call vis-a-vis the stated conditions (if calls-queued < 15) to the skill group corresponding to the 2nd VDN skill (31-Agent who has visited resorts in Texas). If step 3 fails, the *check skill* command in step 5 attempts to queue the call vis-a-vis the stated conditions (if the oldest-call waiting < 10) to the skill group corresponding to the 3rd VDN skill (100-Any agent who can take a reservation).

General Number Dialing

This option allows the caller to dial the general number provided (for example, 615-3111). The caller is then serviced in part via Call Prompting.

Figure 12-4 shows how a call to VDN 3111 can be processed vis-a-vis Call Vectoring.

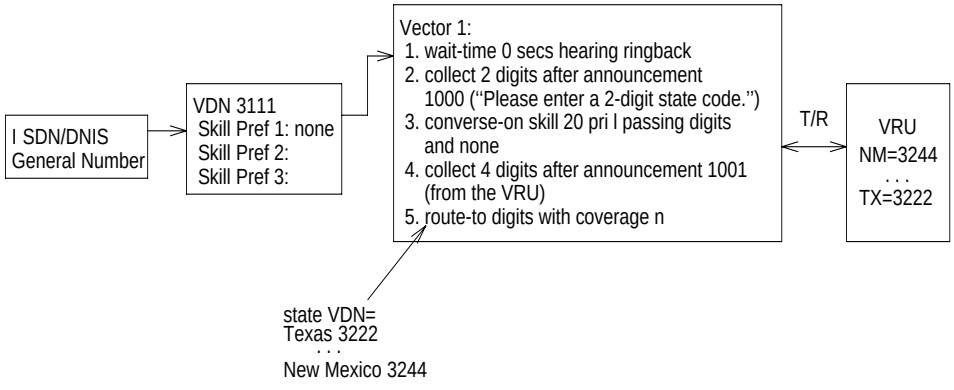


Figure 12-4. Example 7B: Process Involving General Number Dialing

After the number is dialed, the call is directed to VDN 3111, which points to Vector 1. Note there are no skill preferences assigned to VDN 3111. Also, VDN 3111 is the only VDN administered to point to Vector 1. Therefore, this VDN is used for calls from all states.

The *collect digits* command in step 2 of the previous vector first requests the caller to enter the appropriate 2-digit state code and then collects the digits. Let’s assume that the caller enters the correct code for Texas, which is “05.” In such a case, the *converse-on skill* command in step 3 delivers the call to the converse skill if there is a queue for the skill and the queue is not full, or if a VRU port is available.

NOTE:
The *converse-on* command is discussed in Chapter 3.

When the VRU port responds, the step then outpulses the state code “05” to the VRU via the *passing digits* parameter included in the command. Once the VRU receives this state code, the VRU in turn outpulses the Texas VDN (3222) to the switch. Thereafter, the *collect digits* command in step 4 collects the digits comprising this VDN. Finally, the *route-to digits* command in step 5 routes the call to Texas VDN 3222, which points to Vector 2 (illustrated in the previous section).

Call-Back Provisions

After a caller makes a reservation for a resort site, the caller is given a call-back number. Such a number is helpful if the caller needs more information or would like to check on some arrangement that was previously made. Figure 12-5 illustrates one approach for enabling call-back provisions.

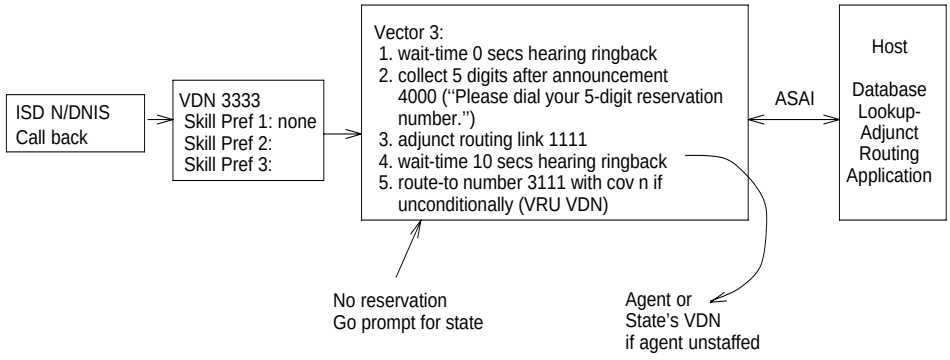


Figure 12-5. Example 7C: Call-Back Provisions

After the number is dialed, the call is directed to VDN 3333, which points to Vector 3. Note there are no skill Preferences assigned to VDN 3333. Also, VDN 3333 is the only VDN administered to point to Vector 3. Therefore, this VDN is used for calls from all states.

The *collect digits* command in step 2 of the previous vector first requests the caller to enter his or her 5-digit reservation number and then collects the digits. Once the digits are collected, the *adjunct routing* command (if successful) in step 3 causes the switch to send the collected digits (along with other information) to the host in the ASAI adjunct routing request. The host then uses these digits to perform a database lookup for the agent who made the reservation and the resort corresponding to the reservation. If the agent is currently logged in, the call is automatically routed to the agent. Once this happens, information on the relevant reservation is displayed at the agent’s data terminal, thus providing quicker and more personal service. On the other hand, if the agent is not logged in, the call is routed to step 5, where the *route to* command unconditionally routes the call to the VRU VDN 3111 (discussed in the “General Number Dialing” section’).

Vector Exercises

This section presents several typical business world scenarios involving telephone usage, and it shows how to write one or more vectors to handle each of these scenarios.

Note that the vectors presented here are intended to be “suggested solutions.” The customer should take into account his or her requirements and budget in selecting and/or writing vectors.

NOTE:
Exercise 1 in this section presents two solutions, one of which involves Call Prompting, which is discussed in Chapter 4.

Exercise 1: Emergency and Routine Service

Write a vector that does the following:

- Delivers the following message to handle emergency calls: “We are aware of the power outage in the northeastern part of the city. Crews have been dispatched. If you are calling for other reasons, please hold to see if an operator is available.”
- Enables the caller to speak with an agent (if available) concerning a nonemergency matter.

Suggested Solution 1

```
 1. wait-time 0 seconds hearing ringback
 2. announcement 4100 ("We are aware of the
    power outage in the northeastern part of the city.
    Crews have been dispatched.  If you are calling for
    other reasons, please hold to see if an operator
    is available.")
 3. wait-time 2 seconds hearing ringback
 4. goto step 10 if calls-queued in split 1 pri l > 20
 5. queue-to split 1 pri l
 6. wait-time 6 seconds hearing music
 7. announcement 4200 ("We’re sorry.  All of
    our operators are busy.  Please hold.")
 8. wait-time 10 seconds hearing music
 9. goto step 7 if unconditionally
10. disconnect after announcement 4200 ("We’re
    sorry.  All of our operators are busy at the moment.
    Please call back at your convenience.")
```

Screen 12-10. Emergency and Routine Service (Call Vectoring Option)

In step 2 of this vector, Screen 12-10, the *announcement* command provides the caller with the appropriate emergency information, and it invites the caller to hold if he or she wishes to speak with an operator on another matter. If the caller holds, the caller hears several seconds of ringback provided by the *wait-time* command in step 3. Thereafter, the *goto step* command in step 4 checks whether there are more than 20 calls queued in split 1. If so, a branch is made to step 10, where the *disconnect after announcement* command first informs the caller that the call cannot be serviced at this time and then drops the call. On the other hand, if 20 or fewer calls are queued to split 1, the call is queued to the split

by the *queue-to split* command in step 5. Thereafter, unless the call is answered, feedback in the form of music is provided by step 6 and an announcement urging the caller to hold is provided by step 7. After another wait with music period (if necessary) provided by step 8, the *goto step* command in step 9 branches back to the aforementioned “please hold” announcement in step 7. The resulting “announcement-wait” loop (steps 7 through 9) is then repeated until either an agent answers the call or the caller hangs up.

Suggested Solution 2

```
VDN (extension=1030   name="Hub"   vector=30)
Vector 30:
      1. wait-time 0 seconds hearing ringback
      2. collect 1 digits after announcement 3000
         ("We are aware of the power outage in the northeastern
         part of the city.  Crews have been dispatched.  If
         you are calling for other reasons, please press 1.
         Otherwise, please hang up now.")
      3. route-to number 1031 with cov y if digit = 1
      4. announcement 3100 ("Entry not understood.  Please
         try again.")
      5. goto step 2 if unconditionally
VDN (extension=1031   name="Service"   vector=31)
Vector 31:
      1. announcement 4000 ("Please hold.  We will
         try to connect you to an operator.")
      2. wait-time 2 seconds hearing ringback
      3. goto step 9 if calls-queued in split 1 pri l > 20
      4. queue-to split 1 pri l
      5. wait-time 6 seconds hearing music
      6. announcement 4200 ("We’re sorry.  All of
         our operators are busy.  Please hold.")
      7. wait-time 10 seconds hearing music
      8. goto step 6 if unconditionally
      9. disconnect after announcement 4200 ("We’re
         sorry.  All of our operators are busy at the moment.
         Please call back at your convenience.")
```

Screen 12-11. Emergency and Routine Service (Call Vectoring and Call Prompting Option)

Suggested Solution 2 involves both Call Vectoring and Call Prompting. Also, it involves two vectors instead of just one vector, and it assumes the caller has a touch-tone telephone.

The announcement portion of the *collect digits after announcement* command in step 2 of Vector 30 first provides the caller with the appropriate emergency information. It then invites the caller to press “1” if the caller is calling for some other reason. If this is not the case, it finally suggests that the caller hang up.

First, let’s assume the caller wants to hold the line but enters the incorrect touch-tone digit (“2,” for example). In such a case, the *route-to number* command in step 3 attempts to route the call to VDN extension 1031 according to

the entered digit. However, because a number other than “1” has been entered, the call is not routed to the VDN extension. Instead, control is passed to step 4, where the *announcement* command first informs the caller of the input error and then invites the caller to try again. Thereafter, the *goto step* command in step 5 unconditionally sends control back to step 2, where the *collect digits* command ultimately collects the digit entered by the caller. The digit-input loop (steps 2 through 5) continues for as long as the caller enters an incorrect digit.

On the other hand, let’s assume that the caller correctly enters the digit “1” as requested by the *collect digits* command in step 2. In such a case, the *route-to number* command in step 3 sends control to the vector whose VDN extension is “1031” (that is, to Vector 31). Thereafter, the call is processed almost identically to the procedure explained in Suggested Solution 1.

Exercise 2: Late Caller Treatment

Your ACD is staffed by union agents. The latest union agreement stipulates that these agents are free to leave promptly at 5:00 P.M. However, you are concerned about the callers who will call shortly before 5:00 P.M. on any given day and find themselves waiting in queue (and, in effect, ignored) after the top of the hour.

Write a vector that warns late callers that their call may not be serviced. (Business hours are from 8:00 A.M. to 5:00 P.M., Monday through Friday.)

Suggested Solution

```
 1. goto step 15 if time-of-day is all 1700 to all 0800
 2. goto step 15 if time-of-day is fri 1700 to mon 0800
 3. goto step 16 if calls-queued in split 1 pri l > 20
 4. queue-to split 1 pri l
 5. goto step 10 if time-of-day is all 1645 to all 1700
 6. wait-time 20 seconds hearing ringback
 7. announcement 100 ("We're sorry, all of our
    agents are busy...Please hold...")
 8. wait-time 998 seconds hearing music
 9. stop
10. announcement 200 ("It is almost closing time.
    We will try to service you before we close for the day.
    However, if we are unable to do so, please call back
    at your convenience between 8:00 A.M. and 5:00 P.M.,
    Monday through Friday.")
11. wait-time 30 seconds hearing music
12. goto step 14 if time-of-day all 1700 to all 1710
13. goto step 11 if unconditionally
14. disconnect after announcement 300 ("We're
    sorry, our office is now closed.  Please call back
    at your convenience between 8:00 A.M. and 5:00 P.M.,
    Monday through Friday.")
15. disconnect after announcement 400 ("We're
    sorry, our office is closed.  Please call back at
    your convenience between 8:00 A.M. and 5:00 P.M.,
    Monday through Friday.")
16. disconnect after announcement 500 ("We're
    sorry, we cannot service your call at this time.
    Please call back at your convenience between
    8:00 A.M. and 5:00 P.M., Monday through Friday.")
```

Screen 12-12. Late Caller Treatment

This vector, Screen 12-12, provides specific treatment for calls coming into the switch after working hours, during the weekend, or as the working day comes to a close.

The *goto step* command in step 1 checks whether the call is being placed during nonworking hours during the week (that is, between 5:00 P.M. and 8:00 A.M. on any day of the week). If the call is being placed at this time, a branch is made to step 15, where the *disconnect after announcement* command first informs the caller that the office is closed and then drops the call. On the other hand, if the call is not being placed at this time, control is passed to step 2, where another *goto step* command checks whether the call is being placed during "weekend" hours (that is, between 5:00 P.M. Friday and 8:00 A.M. Monday). If so, a branch is made to step 15, as is the case for a failure of the *goto step* command in step 1. On the other hand, if the call is not being placed at this time, control is passed to step 3.

The *goto step* command in step 3 checks for the number of calls in split 1. If more than 20 calls are queued to split 1, control is passed to step 16, where the *disconnect after announcement* command first informs the caller that the call

cannot be serviced at this time and then disconnects the call. On the other hand, if 20 or fewer calls are queued to split 1, control is passed to step 4, where the *queue-to split command* queues the call to split 1.

Control is then passed to step 5, where the *goto step* command checks whether the current time is any time between 4:45 P.M. and 5:00 P.M. inclusive (or, in other words, very close to [if not] closing time). If the current time does not fall within this clock range, the *wait-time* command in step 6 provides the caller with 20 seconds of ringback. Thereafter, the *announcement* command in step 7 plays the appropriate “hold” message, and the *wait* command in step 8 provides the caller with 998 seconds of music. Finally, the *stop* command in step 9 halts vector processing, and the call remains in queue until either the agent answers the call or the caller hangs up.

On the other hand, if the current time is any time between 4:45 P.M. and 5:00 P.M., inclusive when step 5 is executed, a branch is made to step 10, where the appropriate “late caller” announcement is provided to the caller. Thereafter, the *wait-time* command in step 11 provides the caller with 30 seconds of music. Control is then passed to step 12, where the *goto step* command checks whether the time is currently any time between 5:00 P.M. and 5:10 P.M., inclusive. If so, control is passed to step 14, where the *disconnect after announcement* command first informs the caller that the office is now closed and then invites the caller to call back at the appropriate time before finally disconnecting the call. On the other hand, if the time is currently not between 5:00 P.M. and 5:10 P.M. inclusive, control is passed to step 13, where the *goto step* command branches back to the *wait-time* command in step 11. The resulting loop consisting of steps 11 through 13 is repeated for as long as the time is between 5:00 P.M. and 5:10 P.M. inclusive, or until the caller hangs up. Once step 12 is executed at least a second after 5:10 P.M., control is passed to step 14 as described previously.

Exercise 3: Messaging Option

Write a vector that:

- Does the following if the oldest call waiting has been in queue longer than 75 seconds:
  - — Sends the call to AUDIX (if possible)
  - — Delivers to the caller the following personalized AUDIX message: "All of our MegaSports agents are busy...Please leave your name and telephone number."
- Plays for the caller 30 seconds of ringback
- Plays for the caller (after the ringback) an announcement followed by music

Suggested Solution

```
 1. goto step 8 if oldest-call-wait in split 50 pri l > 74
 2. goto step 8 if calls-queued in split 50 pri l > 20
 3. queue-to split 50 pri l
 4. wait-time 30 seconds hearing ringback
 5. announcement 1000 ("All of our MegaSports
    agents are busy...Please wait...")
 6. wait-time 998 seconds hearing music
 7. stop
 8. announcement 2000 ("We're sorry, all of our
    MegaSports agents are busy.  If you'd like to leave a
    message, please do so after the tone.  Otherwise, please
    call back between 8:00 A.M. and 5:00 P.M, Monday through
    Friday. Thank you.")
 9. messaging split 20 for extension 4000
10. disconnect after announcement 2050 ("We're sorry, we are unable
    to take your message at this time.  Please call back
    between 8:00 A.M. and 5:00 P.M., Monday through Friday.
    Thank you.")
```

Screen 12-13. Messaging Option

The *goto step* command in step 1 of this vector, Screen 12-13, checks whether the oldest call waiting in split 50 has been waiting for 75 seconds or more. If so, control is passed to step 8, where the *announcement* command first informs the caller that all the agents are busy and then invites the caller to either call back at the appropriate time or leave a recorded message for the agent. If the caller elects to leave a message, the *messaging split* command in step 9 is executed. Upon execution of the *messaging split* command, an attempt is made to connect the caller to AUDIX so that he or she can leave a recorded message. If the split queue is full, or if the AUDIX link is down, termination to AUDIX is unsuccessful, and vector processing continues at the next vector step, which (as is the case here) usually contains an announcement that provides the caller with the appropriate apology and subsequent directives. If the caller is successfully connected to AUDIX, vector processing terminates, and a message may be left for the specified mailbox (4000, in this case).

Returning to step 1, if on the other hand the oldest call waiting in split 50 has been waiting fewer that 75 seconds, control is passed to step 2, where another *goto step* command checks for the number of calls in split 50. If more than 20 calls are queued to split 50, control is passed to step 8. Thereafter, the procedure for the messaging option provided in the previous paragraph is implemented. On the other hand, if there are 20 or fewer calls waiting in split 50, control is passed to step 3, where the *queue-to split* command queues the call to the split. Thereafter, the obligatory *wait-time* and *announcement* steps (steps 4 through 6) are executed, followed by the *stop* step (step 7).

Call Vectoring Commands

A

Introduction

This appendix is intended to provide various information about the commands used within Call Vectoring. Specifically, the following information is presented:

- Table that contains a brief description of each command's function and also the appendix page where the command can be referenced.
- Table that identifies the commands available in Basic Call Vectoring and/or Call Prompting.
- Job aid tables that graphically illustrate how to use the Call Vectoring commands.
- Manual page directory that details the purpose and function of the Call Vectoring commands and also any relevant interactions involving the commands.
- Table that summarizes the criteria for the success/failure of the Call Vectoring commands.

The information in this appendix applies to DEFINITY® ECS G3R6.3.

Command Description/Reference

Table A-1 provides a brief description of the function of each of the Call Vectoring command. See the listed page number for a complete description of the command.

Table A-1. Command Description/Reference Table

Command	Function	Page
adjunct routing	To request adjunct to route call.	A-17
announcement	To connect caller to delay recording.	A-24
busy	To connect caller to busy tone.	A-26
check split/skill/best	To connect/queue call on a conditional basis.	A-28
collect digits	To prompt caller for digits.	A-34
consider split/skill/location	To obtain BSR status data from a local split (skill) or a remote location	A-39
converse-on split/skill	To deliver a call to a converse split (skill) and to activate a Voice Response Unit (VRU).	A-46
disconnect	To force disconnect of call with optional announcement.	A-58
goto step	To cause unconditional/conditional branch to another step in the vector.	A-60
goto vector	To cause unconditional/conditional branch to another vector.	A-67
messaging split (skill)	To allow caller to leave message for callback.	A-74
queue-to split/skill/best	To connect/queue call to the primary split (skill) or to the best resource found by a consider series	A-78
reply-best	To send BSR status data to the primary vector in a multi-site application	A-83
route-to	To connect call to destination entered via *collect digits* command, or to connect call to internal/external destination.	A-85
stop	To stop further vector processing.	A-94
wait-time	To initiate feedback to caller (if needed) and delay processing of the next step.	A-96

Command/Option Summary

Table A-2 indicates which Call Vectoring commands can be used within Basic Call Vectoring and/or Call Prompting. Other options or later releases may be required for certain commands or functions as noted. See Appendix M for more details about feature availability. Note, if EAS is enabled, “skill” replaces “split.”

Table A-2. Command/Option Summary Table

Command	Basic	Prompting	Other Options Required
adjunct routing	x		ASAI
announcement	x	x	
busy	x		
check best	x		ACD; G3V4 Advanced Routing; Best Service Routing
check split/skill if <condition>	x		ACD
check split/skill if rolling-asa	x		ACD; G3V4 Enhanced; G3V4 Advanced Routing
check split/skill if expected-wait	x		ACD; G3V4 Enhanced; G3V4 Advanced Routing
check best if expected-wait	x		ACD; G3V4 Enhanced; G3V4 Advanced Routing; BSR
check split/skill if oldest-call-wait pri	x		ACD; G3V4 Enhanced
check split/skill/best if wait-improved	x		ACD; G3V4 Advanced Routing; BSR
collect digits		x	
collect ced/cdpd digits		x	Vectoring (CINFO)
consider location	x		ACD; G3V4 Advanced Routing; Best Service Routing; Look-Ahead Interflow
consider split/skill	x		ACD; G3V4 Advanced Routing; Best Service Routing
converse-on split/skill	x		
converse-on split/skill passing wait	x	x	ACD; G3V4 Enhanced; G3V4 Advanced Routing

Continued on next page

Table A-2. Command/Option Summary Table — *Continued*

Command	Basic	Prompting	Other Options Required
disconnect	x		
goto step/vector if unconditionally	x	x	
goto step/vector if <condition> in split/skill	x		ACD
goto step/vector if digits		x	
goto step/vector if time-of-day	x		
goto step/vector if oldest-call-wait pri	x		ACD; G3V4 Enhanced
goto step/vector if rolling-asa	x	x	ACD; G3V4 Enhanced; G3V4 Advanced Routing
goto step/vector if expected-wait	x	x	ACD; G3V4 Enhanced; G3V4 Advanced Routing
goto step/vector if expected-wait for best	x	x	ACD; G3V4 Enhanced; G3V4 Advanced Routing; Best Service Routing
goto step/vector if counted-calls	x	x	G3V4 Enhanced; G3V4 Advanced Routing
goto step/vector if ani	x	x	G3V4 Enhanced; G3V4 ANI/II-Digits Routing
goto step/vector if ii-digits	x	x	G3V4 Enhanced; G3V4 ANI/II-Digits Routing
goto step/vector if wait-improved	x		ACD; G3V4 Advanced Routing; BSR
goto step/vector if interflow-qpos	x		ACD; LAI
messaging split/skill	x	x	
messaging split/skill active/latest[1]	x	x	
queue-to best	x		ACD; G3V4 Advanced Routing; Best Service Routing
queue-to split/skill	x		ACD
reply-best	x		ACD; G3V4 Advanced Routing; Best Service Routing; Look-Ahead Interflow

Continued on next page

Table A-2. Command/Option Summary Table — *Continued*

Command	Basic	Prompting	Other Options Required
route-to digits with cov y (n)		x	
route-to number if digit		x	
route-to number if unconditionally with cov y (n)[1]	x	x	
route-to number if digit with cov y (n)[1]		x	
route-to number if unconditionally	x	x	
route-to number if interflow-qpos	x	x	ACD, Look-Ahead Interflow
stop	x	x	
wait-time	x	x	
wait-time hearing i-silent[1]	x	x	
wait-time hearing <extension> then <treatment>	x	x	G3V4 Enhanced

1. If G3V4 software has not been purchased, these commands require the G3V4 maintenance load.

Command Job Aid

Table A-3. Vectoring Commands

adjunct routing link _____ (extension of adjunct port)
announcement[1] _____ (1- to 5-digit extension)
busy
collect[1] ____ digits after announcement _____ (1-16) (Optional: 1- to 5-digit extension or none)
collect[1] ____ digits (ced or cdpd[2])
disconnect after announcement _____ (1- to 5-digit extension or none [default])
messaging split[1] _____ for extension _____ (1-99 [vs/csi/si*]) (1- to 5-digit extension, active [default], latest)[3] (1-600 [r])
EAS only messaging skill[1] _____ for extension _____ (1-99 [vs/csi/si*]) (1- to 5-digit extension, active [default], latest)[3] (1-600 [r]) (1st, 2nd, 3rd [VDNskill])

1.This command is also available with Call Prompting.
2.The Vectoring (CINFO) command must be active.
3.Active means the called VDN as changed by VDN override. Latest means the VDN assigned to the vector in which the call is currently being processed.

Table A-4. Vectoring Commands

queue-to split _____ pri __ (**l**ow, **m**ed, **h**igh, **t**op) (1-99 {vs/csi/si]) (1-600 [r])
queue-to best
EAS only queue-to skill _____ pri __ (1-99 {vs/csi/si]) (**l**ow, **m**ed **h**igh, **t**op) (1-600 [r]) (1st, 3nd, 3rd] VDN skill])
route-to digits[1] with coverage__ y or n [default] route-to number _____ with cov __ if digit[1] ____ __ (1- to 16-digit number, which may include ~p, ~s, ~w, ~W, ~m, *, or #) (y or n [default] (<, >, <=, >=) (<>, =) (0-9) (0-9, #) interflow-qpos ____ __ (=, <, <=) (1-9) unconditionally
stop[2]
wait-time[2] _____ ____ hearing ____ (0-999) (secs) (music, ringback, silence, i-silent) (0-480) (mins) ____ then ____ (0-8) (hrs) (1- to 5-digit extension) (continue, music, ringback,silence)

1. The Call Prompting feature must be enabled.
2. This command is also available with Call Prompting.

Table A-5. Vectoring Commands

consider split	_____ (1-99 {vs/csi/si]) (1-600 [r])	pri __ (**l**ow, **m**ed, **h**igh, **t**op)	adjust-by	____ (0–100)
consider location	_____ (1-255)	pri __ (**l**ow, **m**ed, **h**igh, **t**op)	adjust-by	____ (0–100)
reply-best				
EAS only				
consider skill	_____ (1-99 {vs/csi/si]) (1-600 [r]) (1st, 3nd, 3rd [VDN skill])	pri __ (**l**ow, **m**ed **h**igh, **t**op)	adjust-by	____ (0–100)

Table A-6. Vectoring Commands

check split	_____ (1-99 {vs/csi/si]) (1-600 [r])	pri __ (**l**ow, **m**ed, **h**igh, **t**op)	if	rolling-asa	<	_____ (1-999)
				available-agents	>	_____ (0-199[vs/csi/si]) (0-998 [r])
				calls-queued	<	_____ (1-200[vs/csi/si]) (1-999 [r])
				expected-wait	<	_____ (1-9999)
				oldest-call-wait	<	_____ (1-999)
				staffed-agents	>	_____ (1-99 {vs/csi/si]) (0-998 [r])
				wait-improved	>	_____ (0–9999)
				unconditionally		
check best			if	expected-wait	<	_____ (1-9999)
check best			if	wait-improved	>	_____ (0-9999)
check best			if	unconditionally		

Continued on next page

Table A-6. Vectoring Commands — *Continued*

EAS only						
check skill	_____ (1-99 {vs/csi/si]) (1-600 [r]) (1st, 2nd, 3rd [VDN skill])	pri __ (**l**ow, **m**ed, **h**igh, **t**op)	if rolling-asa	<	_____ (1-999)	
			available-agents	>	_____ (1-99 {vs/csi/si]) (0-998 [r])	
			calls-queued	<	_____ (1-200 [vs/csi/si]) (1-999 [r])	
			expected-wait	<	_____ (1-9999)	
			oldest-call-wait	<	_____ (1-999)	
			staffed-agents	>	_____ (1-99 {vs/csi/si]) (0-998 [r])	
			wait-improved	>	_____ (0–9999)	
			unconditionally			
check best		if	expected-wait	<	_____ (1-9999)	
check best		if	wait-improved	>	_____ (0-9999)	
check best		if	unconditionally			

Table A-7. Vectoring Commands

converse-on split	_____ (1-99 [vs/csi/si]) (1-600 [r])	pri __ (**l**ow, **m**ed, **h**igh, **t**op)	DATA 1[1] and (string of up to six digits or asterisks, “vdn,” “ani,”[2] “digits,”[3] “qpos,” “wait,” “#,” or “none”)	DATA 2 (string of up to six digits or asterisks, “vdn,” “ani,”[2] “digits,”[3] “qpos,” “wait,” “#,” or “none”)
EAS only				
converse-on skill	_____ (1-99 [vs/csi/si]) (1-600 [r]) (1st, 2nd, 3rd [VDN skill])	pri __ (**l**ow, **m**ed, **h**igh, **t**op)	DATA 1[1] and (string of up to six digits or asterisks, “vdn,” “ani,”[2] “digits,”[3] “qpos,” “wait,” “#,” or “none”)	DATA 2 (string of up to six digits or asterisks, “vdn,” “ani,”[2] “digits,”[3] “qpos,” “wait,” “#,” or “none”

1. If DATA 1 is administered as “none,” DATA 2 must also be administered as “none.”
2. The Vectoring (G3V4 Advanced Routing) option must be enabled.
3. The Call Prompting feature must be enabled.

Table A-8. Vectoring Commands

goto step[1]	____ if (1-32)	ani[2]	_____ (<, >, <=, >=)[3] (<>, =)	_____ (1-16 characters: 0-9) (1-16 characters, 0-9, ?, +, none)[4]		
		ani[2]	_____ (in, not-in)	table	_____ (1-10 [vs/csi/si]) (1-100 [r])	
		rolling-asa for[5]	vdn	_____ (VDN, active, latest)[6]	_____ (<, <=) (>, >=, <>, =)	_____ (1-999) (0-999)
		rolling-asa for[5]	split	_____ 1-99 [vs/csi/si]) 1-600 [r] (The models don't limit the comparators or ranges)	_____ (<, <=) (>, >=, <>, =)	_____ (1-999) (0-999)
		available-agents in split	_____ (1-99 [vs/csi/si]) 1-600 [r])	_____ (<, <=) (>, >=, <>, =)	_____ (1-200 [vs/csi/si], 1-999 [r]) (0-199 [vs/csi/si], 0-998 [r])	
		calls-queued in split	_____ (1-99 [vs/csi/si]) 1-600 [r])	pri __ (**l**ow, **m**ed, **h**igh, **t**op)	_____ (<, <=) (>, >=, <>, =)	_____ (1-200 [vs/si], 1-999 {r]) (0-199 [vs/si], 0-998 [r])
		counted-calls to vdn[5]	_____ VDN, active, latest)[6]	_____ (<, <=) (>, >=, <>, =)	_____ (1-999) (0-999)	
		digits[7]	_____ (in, not-in)	table	_____ (1-5 [vs/csi/si]) (1-100 [r])	
		digits[7]	_____ (<, >, <=, >=) (<>, =)	_____ (1-16 characters: 0-9) (1-16 characters: 0-9, #, ?, +, none)[4]		
		expected-wait for[5]	split	_____ (1-99 [vs/csi/si]) 1-600 [r])	pri __ (**l**ow, **m**ed, **h**igh, **t**op)	_____ (<, <=) (>, >=, <>, =) _____ (1-9999) (0-9999)

1. The "goto vector" command is identical to the "goto step" command, except the word step is replaced by the word vector. The valid values for vector are: 1-256 [vs/csi/si], 1-512 [r].
2. The Vectoring (ANI/II-Digits Routing) and Vectoring (G3V4 Enhanced) options must be enabled.
3. The comparator <= means less than or equal to, >= means greater than or equal to, <> means not equal to, and = means equal to.
4. A ? can be entered in any character position and matches any character in that single character position. A + must be entered as either the first or last character of the string and matches any or no characters. If + is used the maximum length of the field is 15.
5. The Vectoring (G3V4 Advanced Routing) and Vectoring (G3V4 Enhanced) options must be enabled.
6. Active means the called VDN as changed by VDN override. Latest means the VDN assigned to the vector in which the call is currently being processed.
7. The Call Prompting feature must be enabled.

Table A-9. Vectoring Commands

goto step[1]	____ if (1-32)	expected-wait[2] call for		____ (<, <=)[3] (>, >=, <>, =)	____ (1-9999) (0-9999)		
		expected-wait for	best	____ (<, <=) (>, >=, <>, =)	____ (1-9999) (0-9999)		
		wait-improved for split[4]		___ (1-99 [vs/csi/si]) (1-600 [r])	pri __ (**l**ow, **m**ed, **h**igh, **t**op)	____ (<, <=) (>, >=, <>, =)	____ (1-9999) (0-9999)
		wait-improved for best[4]		____ (<, <=) (>, >=, <>, =)	____ (1-9999) (0-9999)		
		interflow-qpos		____ (=, <, <=, >, >=, <>)	____ (1-9)		
		ii-digits[5]	____ (in, not-in)	table	____ (1-10 [vs/csi/si]) (1-100 [r])		
		ii-digits[5]	____ (<, >, <=, >=) (<>, =)	____ (00-99) [00-99, +, +?, ?+, ??, ?x (where x = 0-9), x?, none][6]			
		oldest-call-wait in split	_____ (1-99 [vs/csi/si]) (1-600 [r])	pri __ (**l**ow, **m**ed, **h**igh, **t**op)	____ (<, <=) (>, >=, <>, =)	____ (1-999) (0-999)	
		staffed-agents in split	____ (1-99 [vs/csi/si]) 1-600 [r])	____ (<, <=) (>, >=, <>, =)	____ (1-200 [vs/csi/si], 1-999 [r]) (0-199 [vs/csi/si], 0-998 [r])		
		time-of-day is	____ (mon, tue, wed, thu, fri, sat, sun, all)	____ to (hh:mm in military time)	____ (mon, tue, wed, thu, fri, sat, sun, all)	____ (hh:mm in military time)	
		unconditionally[7]					

1. The “goto vector” command is identical to the “goto step” command, except the word step is replaced by the word vector. The valid values for vector are: 1-48 [vs/csi/si], 1-512 {r].
2. The Vectoring (G3V4 Advanced Routing) and Vectoring (G3V4 Enhanced) options must be enabled.
3. The comparator <= means less than or equal to, >= means greater than or equal to, <> means not equal to, and = means equal to.
4. Vectoring (Best Service Routing) must be enabled.
5. The Vectoring (ANI/II-Digits Routing) and Vectoring (G3V4 Enhanced) options must be enabled.
6. A ? matches any digit 0-9 in that single digit position. A + matches any or no characters.
7. This command is also available with Call Prompting.

Table A-10. Vectoring Commands

EAS only

goto step[1]	____ if (1-32)	ani[2]	____ (<, >, <=, >=)[3] (<>, =)	____ (1-16 characters: 0-9) (1-16 characters, 0-9, ?, +, none)[4]		
		ani[2]	____ (in, not-in)	table	____ (1-10 [vs/csi/si]) (1-100 [r])	
		rolling-asa for[5]	vdn	____ (VDN, active, latest)[6]	____ (<, <=) (>, >=, <>, =)	____ (1-999) (0-998)
		rolling-asa for[5]	skill	____ (1-99 [vs/csi/si]) (1-600 [r]) (1st, 2nd, 3rd [VDN skill])	____ (<, <=) (>, >=, <>, =)	____ (2-998) (0-998)
		available-agents in skill	____ (1-99 [vs/csi/si]) 1-600 [r]) (1st, 2nd, 3rd [VDN skill])	____ (<, <=) (>, >=, <>, =)	____ (1-200 [vs/csi/si], 1-999 [r]) (0-199 [vs/csi/si], 0-998 [r])	
		calls-queued in skill	____ (1-99 [vs/csi/si]) (1-600 [r]) (1st, 2nd, 3rd [VDN skill])	pri __ (**l**ow, **m**ed, **h**igh, **t**op)	____ (<, <=) (>, >=, <>, =)	____ (1-200 [vs/csi/si], 1-999 {r]) (0-199 [vs/csi/si], 0-998 [r])
		counted-calls to vdn	____ VDN, active, latest)[6]	____ (<, <=) (>, >=, <>, =)	____ (1-999) (0-999)	
		digits[7]	____ (in, not-in)	table	____ (1-5 [vs/csi/si]) (1-100 [r])	
		digits[7]	____ (<, >, <=, >=) (<>, =)	_____ (1-16 characters: 0-9) (1-16 characters: 0-9, #, ?, +, none)[4]		

1. The “goto vector” command is identical to the “goto step” command, except the word step is replaced by the word vector. The valid values for vector are: 1-48 [vs/csi/si], 1-512 {r].
2. The Vectoring (ANI/II-Digits Routing) and Vectoring (G3V4 Enhanced) options must be enabled.
3. The comparator <= means less than or equal to, >= means greater than or equal to, <> means not equal to, and = means equal to.
4. A ? can be entered in any character position and matches any character in that single character position. A + must be entered as either the first or last character of the string and matches any or no characters. If + is used the maximum length of the field is 15.
5. The Vectoring (G3V4 Advanced Routing) and Vectoring (G3V4 Enhanced) options must be enabled.
6. Active means the called VDN as changed by VDN override. Latest means the VDN assigned to the vector in which the call is currently being processed.
7. The Call Prompting feature must be enabled.

Table A-11. Vectoring Commands

EAS only

goto step[1]	____ if (1-32)	expected-wait for[5] skill	_____ (1-99 [vs/csi/si]) (1-600 [r])	pri __ (**l**ow, **m**ed, **h**igh, **t**op)	_____ (<, <=) (<, <=, <>, =)	_____ (1-9999) (0-9999)
		expected-wait[2] for call	_____ (<, <=)[3] (>, >=, <>, =)	_____ (1-9999) (0-9999)		
		expected-wait for best	_____ (<, <=) (>, >=, <>, =)	_____ (1-9999) (0-9999)		
		wait-improved for skill[4]	___ (1-99 [vs/csi/si]) (1-600 [r])	pri __ (**l**ow, **m**ed, **h**igh, **t**op)	_____ (<, <=) (>, >=, <>, =)	_____ (1-9999) (0-9999)
		wait-improved for best[4]	_____ (<, <=) (>, >=, <>, =)	_____ (1-9999) (0-9999)		
		interflow-qpos	_____ (=, <, <=, >, >=, <>)	_____ (1-9)		
		ii-digits[5] _____ (in, not-in)	table	_____ (1-10 [vs/csi/si]) (1-100 [r])		
		ii-digits[5] _____ (<, >, <=, >=) (<>, =)	_____ (00-99) [00-99, +, +?, ?+, ??, ?x (where x = 0-9), x?, none][6]			
		oldest-call-wait in skill _____ (1-99 [vs/csi/si]) (1-600 [r]) (1st, 2nd, 3rd [VDN skill])	pri __ (**l**ow, **m**ed, **h**igh, **t**op)	_____ (<, <=) (>, >=, <>, =)	_____ (1-999) (0-999)	
		staffed-agents in skill _____ (1-99 [vs/csi/si]) (1-600 [r]) (1st, 2nd, 3rd [VDN skill])	_____ (<, <=) (>, >=, <>, =)	_____ (1-200 [vs/csi/si], 1-999 [r]) (0-199 [vs/csi/si], 0-998 [r])		
		time-of-day is _____ (mon, tue, wed, thu, fri, sat, sun, all)	_____ to (hh:mm in military time)	_____ (mon, tue, wed, thu, fri, sat, sun, all)	_____ (hh:mm in military time)	
		unconditionally[7]				

1. The “goto vector” command is identical to the “goto step” command, except the word step is replaced by the word vector. The valid values for vector are: 1-48 [vs/csi/si], 1-512 {r].
2. The Vectoring (G3V4 Advanced Routing) and Vectoring (G3V4 Enhanced) options must be enabled.
3. The comparator <= means less than or equal to, >= means greater than or equal to, <> means not equal to, and = means equal to.
4. Vectoring (Best Service Routing) must be enabled.
5. The Vectoring (ANI/II-Digits Routing) and Vectoring (G3V4 Enhanced) options must be enabled.
6. A ? matches any digit 0-9 in that single digit position. A + matches any or no characters.
7. This command is also available with Call Prompting.

Command Directory

The manual page directory in this section lists and discusses all of the commands used within Call Vectoring. For each command presented, the following is provided: purpose, syntax, valid entries, requirements, an example, description of the command's operation, answer supervision considerations, feature interactions, and CMS/BCMS interactions. The following points concerning the appearance of the command line are in effect:

- Data that must be entered as part of the command line is shown in **bold**.
- Variable fields that (in most cases) must be completed are enclosed in *< >*.
- Optional fields are enclosed in [ ].

NOTE:
If a variable field appears within an optional field, an entry for the variable field appears only if the optional field is included during command execution.

NOTE:
If EAS is enabled, "skill" replaces "split."

Adjunct Routing Command

```
adjunct routing
```

Purpose

Causes a message to be sent to an adjunct requesting routing instructions.

Syntax

adjunct routing link <extension>

Valid Entries

Valid ASAI link extension number

Requirements

Basic Call Vectoring and ASAI software must be installed. Also, an ISDN-BRI port is required, and the port must be connected to an ASAI host.

Example

adjunct routing link *765*

Operation

The *adjunct routing* command provides a means for an adjunct ASAI processor to specify the destination of a call. The switch provides information in an ASAI route request message that the ASAI adjunct can use to first access a data base and then determine a route for the call. In a typical application, the ASAI adjunct might use the dialed number, the calling party number (CPN/BN), or the digits collected via Call Prompting or Caller Information Forwarding (CINFO) to access customer information and thereby determine the call route. A maximum of 16 digits collected from the last *collect digits* command can be passed.

An adjunct specified in an *adjunct routing* command can route a call to an internal number, an external number, a split, a VDN, an announcement extension, or a particular agent. An adjunct can also provide priority ringing and priority queuing.

When a call encounters an *adjunct routing* command, the switch sends to the specified adjunct an ASAI message requesting a call route. The following list identifies the contents of the message, along with a comment or a brief explanation for each item:

- **Calling number information**. Calling party number or billing number (CPN/BN) provided by ISDN-PRI or R2MFC signaling facilities. If the call originates from a local switch extension, this extension is the calling number.
- **Originating line information (II-digits)**. Two-digit code provided by ISDN-PRI facilities indicating the type of originating line being used.
- **Called number**. Originally called extension (if a call is forwarded to a VDN), or the first VDN through which the call was routed (if the call was not forwarded to the VDN).
- **Routing VDN**. Last VDN that routed the call to the vector that contains the *adjunct routing* command.
- **Call identifier**. ASAI identifier that permits the ASAI adjunct to track multiple calls via either Event Notification or Third Party Call Control. (See *DEFINITY Enterprise Communications Server CallVisor ASAI Technical Reference* (555-230-220) for more information on ASAI.)
- **Look-Ahead Interflow information** (if any). Includes the original VDN display information, the priority level of the call at the originating switch, and the time that the call entered vector processing.
- **Digits collected via Call Prompting** (if any). Digits are collected by the most recent *collect digits* command. These could be CINFO digits, but if so it will not be indicated by ASAI. (See Chapter 4.)
- **User-to-User Information** (if any). User-provided data associated with the call. If provided by ASAI, this data was provided in a 3rd-Party-Make-Call, Auto-Dial, or Route-Select message. If provided over ISDN, the data was in the SETUP message that delivered the call to this switch.

The *wait-time hearing i-silent* command is used in cases where it is important to allow the adjunct to decide whether to accept an incoming ISDN-PRI call. When this step is encountered after an *adjunct routing* step, the DEFINITY System does not return an ISDN PROGress message to the originating switch. This is particularly important for Network ISDN features and for the Look-Ahead Interflow feature.

If the call is queued, the *adjunct routing* step is ignored, and vector processing continues at the next vector step.

If the ASAI link specified in the *adjunct routing* step is down, the step is skipped.

An ASAI link failure can change the manner in which subsequent treatment (that is, *announcement* and/or *wait-time*) steps (if any) in the vector are usually processed. In some cases, such processing is influenced by the position that the treatment steps occupy in the vector. In other cases, the positioning of these commands along with their relationship to specific *goto* commands come into play. For example, any *announcement* or *wait-time* step that immediately follows an *adjunct routing* step whose ASAI link is down is skipped.

The second step after the *adjunct routing* step is often implemented as a default treatment (for example, a route-to an attendant). If the ASAI link is down, the default step executes immediately. Otherwise, the step executes only if the application does not respond with a route within the time period specified by the *wait-time* step.

On the other hand, if a *goto* step follows such an *adjunct routing* step, the switch executes the *goto* step and then skips various treatment steps according to their position in the vector and based on the performance of the *goto* step. Specifically, if the *goto* step *succeeds*, the switch skips any *announcement* or *wait-time* step that is the first non-*goto* step branched to by the *goto* step.

NOTE:
Actually, other than to another *goto* step, the first step to which a *goto* step is usually designed to branch is a nontreatment step (that is, a step containing a command other than a *wait-time* or an *announcement* command). Thus, the skipping of a treatment step according to the scenario described just before this note rarely occurs.

On the other hand, if the *goto* step fails, the switch skips any *announcement* or *wait-time* step that *immediately* follows the *goto* step.

NOTE:
The *goto* step that fails can be at the end of a sequence of *goto* steps that branch to each other.

After the switch sends a route request to the ASAI adjunct, vector processing continues with the vector steps that follow.

The step that follows the *adjunct routing* step, in effect, determines the *maximum* length of time the switch will wait for the ASAI adjunct to reply with a call route. Accordingly, you should always include either a *wait-time* step or an *announcement* step immediately after an *adjunct routing* step. Moreover, the switch cancels the route request if vector processing encounters a step containing one of the commands that follow:

- *adjunct routing*
- *busy*
- *check split*
- *collect digits*
- *converse-on split*
- *disconnect*
- *messaging split*
- *queue-to split*
- *route-to*

NOTE:
Actually, if another *adjunct routing* step is encountered, the route request information is not lost. Although the initial route request is cancelled, a second route request is sent, and this route request includes the same information included in the first route request.

If a valid call route is received by the switch before one of the vector commands in the previous list is executed, the switch routes the call to the destination specified by the adjunct route. Otherwise, the route request is terminated without affecting vector processing.

Finally, note that the adjunct can also decide to not route a call by rejecting (that is, negatively acknowledging) the route request sent by the switch. Upon receiving a route request rejection, the switch terminates the *announcement* or *wait-time* step that is being executed for the call and then continues with the next vector step.

When the switch receives a call route (destination) from the ASAI adjunct, the switch first validates the route as follows:

1. The switch verifies that the VDN's COR permits the call to be terminated at the adjunct-supplied destination.
2. The switch verifies that the adjunct-supplied information (destination number, ACD split, TAC/AAR/ARS access code, etc.) for the route is valid. This includes checking that the destination is compatible with the dial plan, and that the options specified by the adjunct are correct.
3. If the ASAI adjunct specifies the Direct Agent Call (DAC) option, the destination number (agent) must be logged into the adjunct-specified ACD split.
4. If the destination for the call is external, the switch verifies the trunk is available for the call.

If any of these conditions are not met, the route validation fails, and the switch does the following:

1. Discards the route.
2. Notifies the ASAI adjunct that the route is invalid.
3. Continues with vector processing.

If the route is valid, the switch does the following:

1. Terminates vector processing immediately.
2. Notifies the ASAI adjunct that the route is accepted.
3. Routes the call to the destination specified by the ASAI adjunct.

When the call is routed, the caller hears normal call progress tones and feedback. However, if the call is routed to an extension with no available call appearances and no coverage path, the caller hears the busy tone. Any other features that may be in effect at the adjunct-supplied destination (such as Send-All-Calls or Call Forwarding) interact with the routed call.

NOTE:
The operation described in the previous paragraph is similar to that for the *route-to with coverage* commands.

Answer Supervision Considerations

The command has no bearing on answer supervision.

If adjunct routing is used with ISDN-PRI then an *adjunct routing* command followed by a *wait-time hearing* silence, music, or ringback will signal to the originating switch that the DEFINITY System has accepted the call even though answer supervision has not been provided. To prevent this from occurring, the *wait-time hearing i-silent* should be used following the *adjunct routing* step.

Feature Interactions

For a call coming in directly to a VDN, the command is treated like a *route-to with coverage=y* command. However, for a call that is covered to a VDN, the command is treated like a *route-to with coverage=n* command. A covered call that is routed by an *adjunct routing* command to a destination that has Call Forwarding activated is not further redirected (since the call has already been redirected by coverage).

For Look-Ahead Interflow or Network ISDN features, the adjunct routing command is considered a neutral vector command in all cases. However, the command is usually followed by an *announcement* or *wait-time* command, each of which is a call acceptance command. The G3V4 *wait-time hearing i-silent* command can be used when a neutral *wait-time* command is required to allow the adjunct to accept or reject the call.

If an *announcement* command follows a failed *adjunct routing* command, the announcement is interrupted. If the *adjunct routing* command succeeds (that is, the switch receives a destination from the ASAI adjunct), the announcement terminates immediately.

If an ASAI adjunct has supplied dial-ahead digits for a *collect digits* step, and the vector processes a *collect ced digits* or *collect cdpd digits* step, the ASAI supplied dial-ahead digits are discarded without notification to the adjunct.

If a TTR is connected to a call because an ASAI adjunct has requested digit collection, and the vector processes a *collect ced digits* or *collect cdpd digits* step, the TTR is disconnected from the call.

CMS Interactions

R3 CMS: Adjunct routing attempts are stored in the ADJATTEMPTS database item and reported as "Adjunct Routing Attempts" in standard reports. If the call is queued to a split when the *adjunct routing* command is encountered, the step is skipped, and no messages are sent to CMS. Accordingly, "Adjunct Routing Attempts" is not reported for this call.

When a routing response from the adjunct is successfully executed by the switch, this action is tracked in the ADJROUTED and ADJROUTTIME database items and shown as "Adjunct Routing Completions" in standard reports.

Additional tracking of the "Adjunct Routing" command varies based on the destination successfully routed to as follows:

Routed to Station or to Attendant

Database Item	Report Heading	Notes
OUTFLOWCALLS/ OUTFLOWTIME	Vector Flow Out	
INTIME	Avg Time In Vector	
CONNECTCALLS/ CONNECTTIME	Other Calls Connect	answered calls on R5

Routed to Trunk

Database Item	Report Heading	Notes
OUTFLOWCALLS/ OUTFLOWTIME	Vector Flow Out VDN Flow Out	
INTERFLOWCALLS / INTERFLOWTIME	VDN Flow-Interflow	
INTIME	Avg Time In Vector	

Routed to VDN

Database Item	Report Heading	Notes
OUTFLOWCALLS/ OUTFLOWTIME	Vector Flow Out VDN Flow Out	
INTIME	Avg Time In Vector	
INFLOWCALLS	Vector Flow In VDN Flow In	new vector new VDN

Routed to Split or to Hunt Group

Database Item	Report Heading	Notes
CALLSOFFERRED		new split
LOWCALLS/MEDCALLS		no priority/priority

Split calls are also shown in the standard reports based on the final disposition of the call.

The presence of the command in a vector enables the calls serviced by the vector to be vector-directed. When such a call is answered by an agent, the call is tracked as ACDCALLS/ANSTIME, and it is reported as “ACD Calls,” “Split ACD Calls,” and “Avg Speed Ans.”

A call abandoned after the command routes the call to a station or an attendant is tracked in the VDN tables as ABNCALLS/ABNTIME.

R2 CMS: Adjunct routing attempts are not stored in R2 CMS. Successful adjunct-routed calls are stored in OUTFLOW and FLOWTIME in the Vector and VDN tables. These calls are not shown on standard reports. VECTIME is recorded for the vector and reported as “Avg Time in Vector.”

Calls that route to a split are tracked as INFLOW and are reported as “Flow In.” Calls that route to a trunk are tracked as ROUTEDCALLS/ROUTEDTIME. However, these calls are not shown on standard reports.

BCMS Interactions

If the command advances a call to another position (that is, ASAI routing is successful), the call is tracked as “outflow” in the VDN Report.

Announcement Command

announcement

Purpose

Provides the caller with a recorded announcement.

Syntax

announcement <extension>

Valid Entries

Valid announcement extension number

Requirements

Basic Call Vectoring or Call Prompting software must be installed. Also, integrated board, aux trunk or analog (T&R or Lineside DS1) announcement equipment must be installed. Finally, the announcements themselves need to be administered and recorded. See “Recorded Announcements” in the *DEFINITY Enterprise Communications Server Release 6 Administration and Feature Description,* 555-230-522, for more information.

Example

announcement *2982*

Operation

The announcement is played from beginning to end unless an agent becomes available. In such a case, the announcement is interrupted and (if manual answering operation is assigned to the agent, or if calls are delivered to the agent on a manual answering basis) ringback is provided. If the call is queued, the call remains as such while the announcement is played. Any feedback provided before an announcement is continued until the announcement is played.

If an *announcement* command follows a failed *adjunct routing* command, the announcement is interrupted. If the *adjunct routing* command succeeds (that is, the switch receives a destination from the ASAI adjunct), the announcement terminates immediately.

If the announcement's queue is currently full, the call retries the announcement step every five seconds and for an indefinite period of time before any new vector steps are processed.

The *announcement* command step is skipped, and vector processing continues at the next vector step, whenever any of the following conditions exist:

- Requested announcement is busied out, not available, or not administered.
- Integrated board is not installed.
- External aux trunk or analog equipment is not attached.

For a complete description of the types and operation of announcements see "Recorded Announcements" in the *DEFINITY Enterprise Communications Server Release 6 Administration and Feature Description,* 555-230-522.

Answer Supervision Considerations

Unless answer supervision has already been sent, it is sent as soon as the command starts to process the call (even before the announcement starts).

Feature Interactions

For Look-Ahead Interflow, the command may be considered a call acceptance vector command or a neutral vector command.

The command is considered a call acceptance vector command whenever one of the following is true:

- Announcement is available.
- Call is queued for an announcement.
- Announcement is retried.

The command is considered a neutral vector command whenever the announcement is unavailable.

BCMS/CMS Interactions

The command is not tracked on CMS or BCMS.

Busy Command

busy

Purpose

Gives the caller a busy signal and causes termination of vector processing.

Syntax

busy

Requirements

Basic Call Vectoring software must be installed.

Operation

The command takes effect on non-CO trunk calls whether or not answer supervision has been sent. However, if the call is on a CO trunk and answer supervision has not been sent, the busy is not passed back by the CO, and the caller continues to hear ringback from the CO. Calls are dropped approximately 45 seconds after the busy tone is applied.

If ISDN-PRI is involved, the application of the busy tone is enabled via D-channel messaging. The network switching office returns the busy tone to the caller. The facility to the PBX/ACD is dropped, thus making it immediately available for another call.

Answer Supervision Considerations

After the 45 second timeout, an unanswered CO trunk call is answered and then dropped. All other unanswered calls after this timeout are dropped without being answered. For an ISDN call that has not yet queued or been answered, no timeout occurs, and answer supervision is not sent. Instead, a message requesting a busy tone is sent to the network and, subsequently, the trunk is released.

Feature Interactions

For Look-Ahead Interflow or Best Service Routing, the command is considered a call denial vector command in all cases.

CMS Interactions

R3 CMS:

Busy Command

Database Item	Report Heading
BUSYCALLS/BUSYTIME	Calls Forced Busy Calls Busy/Disc
OTHERCALLS/OTHERTIME	Inbound Other Calls
INTIME	Avg Time In Vector

BUSYTIME, OTHERTIME, and INTIME for splits and vectors are tracked according to when the busy tone starts. BUSYTIME, OTHERTIME and INTIME for VDNs are tracked according to when the trunk idles.

R2 CMS: Calls given forced busy are tracked in the FBUSYCALLS and FBUSYTIME database items. These calls are not shown in standard reports. FBUSYTIME is tracked according to when the busy tone starts. VECTIME is recorded for the vector, and it is reported as “Avg Time in Vector.”

BCMS Interactions

A call that is forced busy due to the command is tracked as “OTHER” in the VDN Report.

Check Command

check

Purpose

Checks the status of a split (skill) for possible termination of the call to that split (skill).

Syntax

check best if <condition> [<comparator> <threshold>]

check split <split #> pri <priority level> if <condition> [<comparator> <threshold>]

check skill <skill #> pri <priority level> if <condition> [<comparator> <threshold>]

Valid Entries

split #: *1 through 99* (vs/csi/si), *1 through 600* (r)

skill #: *1 through 99* (vs/csi/si), *1 through 600* (r), *1st, 2nd, 3rd* (VDN)

priority level: *l* (low), *m* (medium), *h* (high), *t* (top)

condition:[1]	comparator:	threshold:
unconditionally	N/A	N/A
rolling-asa	<	*1-999* (1-second increments)
available-agents	>	*0-199* (vs/csi/si), *0-998* (r)
calls-queued	<	*1-200* (vs/csi/si), *1-999* (r)
expected-wait	<	1-9999
oldest-call-wait	<	*1-999* (1-second increments)
staffed-agents	>	*0-199* (vs/csi/si), *0-998* (r)
wait-improved	>	*0–9999* (1-second increments)

1.Only the *unconditional*, *expected-wait*, and *wait-improved* conditionals are allowed when the *best* keyword is used.

Requirements

Basic Call Vectoring software must be installed, and the split (skill) involved must be vector-controlled. Skills replace splits only if Expert Agent Selection (EAS) is optioned. The conditions “rolling-asa” and “expected-wait” are only available if the “Vectoring (G3V4 Advanced Routing)” customer option is enabled. The *best* keyword and the *wait-improved* conditional are only available if the “Vectoring (Best Service Routing)” customer option is enabled.

1-second increments for *oldest-call-wait* and *rolling ASA* require DEFINITY ECS R6.3 or later and CentreVu® CMS R3V6 or later. Earlier versions of these software packages only allow even-numbered entries.

Examples

check best if **expected-wait** < *30*

check best if **wait-improved** > *20*

check best if **unconditionally**

check split *22* pri *h* if **unconditionally**

check split *11* pri *l* if **available-agents** > *5*

check split *11* pri *t* if **calls-queued** < *5*

check split *12* pri *h* if **wait-improved** > *30*

check skill *2nd* pri *m* if **staffed-agents** > *5*

check skill *25* pri *l* if **oldest-call-wait** < *60*

check skill *12* pri *l* if **rolling-asa** < *50*

check skill *1st* pri *m* if **expected-wait** < *20*

Operation

The *check* command checks the status of a split or skill against conditions specified in the command. If the conditions specified in the command are met, the call is terminated to the split (skill). If the conditions are met but no agents are available, the call is queued to the split (skill) and waits for an agent to become available.

Each *check* command may be used with one of the following three keywords: *split*, *skill*, or *best*. The *check split* or *check skill* command requires you to specify the split or skill to be checked. The *check best* command checks the status of

the best split (skill) identified by the immediately preceding series of *consider* steps, then either terminates or queues the call to that split (skill). You don't have to specify the split (skill) in *check best* commands since DEFINITY ECS compares two or more skills and identifies the "best" in the preceding series of *consider* steps.

The command is customized to check for and/or respond to specific conditions. For example, the command can queue/terminate unconditionally. The command can also queue/terminate if any of the following is true:

- Number of available agents is *greater than* the threshold value.
- Number of staffed agents is *greater than* the threshold value.
- Number of calls queued for a specified priority level or higher is *less than* the threshold value.
- Oldest call waiting in queue at the specified priority level or higher has been waiting *less than* the threshold value, which is expressed in seconds.
- Rolling average speed of answer is *less than* the threshold value, which is expressed in seconds.
- Expected wait time is *less than* the threshold value, which is expressed in seconds.
- Expected wait time will be improved by *more than* the threshold value, which is expressed in seconds, by queuing the call to the split (skill) specified. EWT in the specified split (skill) is compared to the call's current EWT. (A call's EWT will be infinite if the call is not in a queue.)

A call may be queued to up to three splits (skills) simultaneously. A call remains queued either until vector processing terminates (via a successful *disconnect, busy*, or *route-to* command, or via an abandoned call), the call is routed to another VDN (by a *route-to number* or *route-to digits* command), or the call reaches an agent. When an agent becomes available in any split (skill) to which the call is queued, the following actions take place:

- Call begins alerting the agent.
- Call is removed from any other queues.
- Vector processing terminates.

If the desired backup split (skills) is one of the splits (skills) to which the call is already queued, the call is requeued at the new priority level, provided that the command conditions are met. The step is skipped, and vector processing continues at the next step if any of the following conditions are true:

- Command conditions are not met.
- Desired split's (skill's) queue is full.
- Desired split (skill) has no queue and also no available agents.
- Desired split (skill) is not vector-controlled.

- Call is already queued to this split (skill) at the specified priority level.
- Call has been previously queued to three different splits (skills).

NOTE:
A *route-to* to another VDN can be used to remove the call from the splits it is queued to if necessary. The steps in the routed-to vector then can be used to queue to other splits.

Answer Supervision Considerations

No answer supervision is returned.

Feature Interactions

The *check* command can access a AUDIX/Message Center/Server split (skill) in cases where a VDN is assigned as a coverage point. To enable this function, the split (skill) must be assigned as a vector-controlled hunt group.

For Best Service Routing and Look-Ahead Interflow, the command can be considered either a call acceptance vector command or a neutral vector command. For more on BSR interactions, see

The command is considered a call acceptance vector command whenever one of the following is true:

- Call terminates to an agent.
- Call queues to a split (skill).
- BSR interflowed call is accepted at remote interflow vector.

The command is considered a neutral vector command when the call neither terminates nor queues.

No COR checking is carried out when a *check* step places a call to a split or skill.

CMS Interactions

Calls answered via the check command are indicated as answered by backup in CMS.

Only R3V6 or later versions of CMS may be used with Best Service Routing.

R3 CMS: Calls queued via a *check split (skill)* command are tracked as CALLSOFFERRED and LOWCALLS/MEDCALLS/HIGHCALLS/TOPCALLS.

The presence of the command in a vector enables the calls serviced by the vector to be vector-directed. When such a call is answered by an agent, the call is tracked as ACDCALLS/ANSTIME, and it is reported as “ACD Calls,” “Split ACD

Calls," and "Avg Speed Ans." If the call is also queued to other splits (skills), OUTFLOWCALLS/OUTFLOWTIME is tracked in the first split (skill) to which the call queues, and "Flow Out" is reported (unless the split (skill) turns out to be the answering split (skill)). DEQUECALLS/DEQUETIME is tracked in the second and third splits (skills) if these splits (skills) are not the answering split (skill), and the call is reported as "Dequeued Calls" and "Dequeued Avg Queue Time." However, if the second or third split (skill) is the answering split (skill), INFLOWCALLS is tracked in the split (skill), and the call is reported as "Flow In."

Whenever the call is answered in a split (skill) accessed by the *check split* command, the BACKUPCALLS data base item is incremented, and the call is reported as "Calls Ans in Backup" and "Calls Handled/Backup." The "Calls Ans in Main" report item is calculated by using the algorithm ACDCALLS - BACKUPCALLS.

If the call abandons after the command queues the call to a split (skill), ABNCALLS/ABNTIME is tracked for the vector, the VDN, and the first split (skill) to which the call is queued. The call is reported as "Aban Call" and "Avg Aban Time." If the call is also queued to other splits (skills), DEQUECALLS/DEQUETIME is tracked in these splits (skills), and the call is reported as "Dequeued Calls" and "Dequeued Avg Queue Time." **R2 CMS**: An ACD call that is directed via the command and that is subsequently answered is tracked as ANSWERED/ANSDELAY and ACDCALLS. The call is reported as "No. ACD Calls," "No. Calls Ans," and "Avg Speed Ans." If the call is also queued to other splits (skills), OUTFLOW is tracked in these splits (skills), and the call is reported as "Flow Out." Calls directed via the command are tracked as ANSBACK, and they are reported as "No. Calls Ans in Backup."

If the call abandons after the command queues the call to a split (skill), ABNCALLS/ABNTIME is tracked for the vector, the VDN, and the first split (skill) to which the call is currently queued. The call is reported as "No. Aban Calls" and "Avg Aban Time." If the call is also queued to other splits (skills), OUTFLOWCALLS/OUTFLOWTIME is tracked in these splits (skills), and the call is reported as "Flow Out."

BSR status poll calls are not counted as interflows. BSR interflows are now tracked as network interflowed calls (NETCALLS) by CMS at the receiving switch. CMS tracks a call's accumulated time-in-VDN as NETINTIME (that is, the NET_TIME value on CMS at switch C combines the time a call has spent in VDNs at any previous locations, as communicated by ISDN information forwarding. The NETINTIME can be added to the time spent in the local switch to provide reports that include the total time the call has spent in the call center network (e.g., total ASA).

For more information on CMS database units and reports, see *CentreVu® Call Management System R3V6 Administration* (585-215-850) and *CentreVu® Call Management System R3V6 Reports* (585-215-851).

BCMS Interactions

The total number of calls to the VDN that are queued via the command and then answered by an agent within a specified time period is tracked as “ACD CALLS” in the VDN Report. The average time that calls spend in a vector before being connected via the command as an ACD call to an agent is tracked as “AVG SPEED ANS” in the same report.

There is no added tracking for calls interflowed by Best Service Routing ™. BCMS tracks these calls as “outflow” in the VDN Report.

Collect Digits Command

```
collect digits
```

Purpose

Allows the user to enter up to 16 digits from a touch-tone phone, or allows the vector to retrieve Caller Information Forwarding (CINFO) digits from the network.

Syntax

collect <# of digits> digits after announcement <extension>

collect ced digits

collect cdpd digits

Valid Entries

of digits: 1 through 16

extension: *none* or valid announcement extension

Requirements

Call Prompting software must be installed. Also, at least one TN744 Call Classifier circuit pack must be in the system unless the command is used only to collect digits returned by a VRU or sent by the network and never to collect digits from a caller. Call Prompting and Vectoring (CINFO) must both be enabled to collect ced or cdpd digits from the network. In addition, CINFO requires ISDN and the AT&T Network Intelligent Call Processing (ICP) service or equivalent.

Example

collect *12* digits after announcement *2982*

collect *ced* digits

Operation

Collecting Digits on the DEFINITY System

The *collect digits* command allows a caller to enter digits from a touch-tone or an internal rotary phone. An optional announcement may be used to request the caller to enter these digits. The announcement can instruct the user to enter an

asterisk (*) if incorrect data is entered. When the caller enters an asterisk, the digits collected for the current *collect digits* command are deleted, digit collection is restarted, and the announcement is not replayed.

In using this command, the maximum number of digits requested of the caller must be specified in the administration of the command. If the caller can enter fewer digits than the maximum specified, the announcement should instruct the caller to terminate the entry with a pound sign (#) digit as an end-of-dialing indicator. If all the digits strings for all the variations of a specific *collect digits* command are terminated with "#," the "#" must be counted as one of the digits. Therefore, the number of digits collected should include any "#" that needs to be collected. Otherwise, the terminating "#" is kept as a dial-ahead digit and is processed by a subsequent *collect digits* command. If fewer digits than the maximum specified are entered, and if the caller does not complete the entry with a pound sign, an interdigit timeout occurs. The timeout terminates the command, and any digits collected prior to the timeout are available for subsequent vector processing.

Generally, processing of the command requires that a TTR be connected. (If the call originates from an internal rotary phone, no TTR is needed.) TTRs accept the touch-tone digits that are entered by Call Prompting users. TTRs are automatically connected as needed by the system.

The connection of the announcement prompt is skipped and the digit collection phase begins whenever one of the following conditions is true:

- Dial-ahead digits exist.
- No announcement is administered for the *collect digits* step.
- Announcement administered for the *collect digits* step does not exist.

Otherwise, an attempt is made to connect the administered announcement. If the announcement to be connected is busy, and if the queue for the announcement is full, or if there is no queue, the calling party continues to hear the current feedback. The system waits five seconds and then tries again to connect the call to the announcement. This process continues until the call is successfully queued or connected to the announcement, or until the calling party disconnects from the call. If the queue for the announcement is not full, the call is queued for the announcement.

If the announcement to be connected is available (either initially or after queuing, or after system retry), any previous feedback is disconnected, and the calling party is connected to the announcement.

While the announcement is playing, or while the call is being queued for an announcement, the caller may enter digits at any time. This causes the announcement to be disconnected or removed from the queue, as appropriate, and the digit collection phase to begin. If the caller does not enter any digits during the announcement phases, the digit collection phase begins when the announcement completes.

As soon as the digit collection phase begins, interdigit timing is started, unless the TTR is already in timing mode (that is, the dial-ahead capability is active and the TTR is not disconnected).

Digits are *collected* either as digits dialed during the *collect digits* command or as dial-ahead digits dialed since a previous *collect digits* command but prior to the current appearance of the command. Digit collection continues for the current command until one of the following conditions exists:

- Number of digits specified is collected.
- Pound sign (#) digit is collected (signifying end of dialing).
- Inter-digit timer expires.

If, during the digit collection phase, a "*" is encountered within a stream of dialed or dial-ahead digits, all digits that are collected for the current *collect digits* step are discarded. If additional dial-ahead digits occur after the asterisk, these digits continue to be processed. If there are no such digits, and if no TTR is connected, vectoring continues at the next vector step. If a TTR is connected, the caller can start entering digits again. In such a case, the announcement is not replayed, and the interdigit timer is restarted.

NOTE:
If an asterisk is entered after the requested number of digits are entered, the asterisk has no effect on the previously entered digits. However, in such a case, the asterisk is treated as a dial-ahead digit for the next *collect digits* command.

When digit collection is completed, and if a TTR is connected (for a touch-tone phone), the interdigit timer is restarted to detect a timeout for releasing the TTR. Vector processing then continues at the next vector step. However, the switch continues to collect any subsequent dialed digits (including the pound sign (#) and asterisk (*) digits) to allow for the dial-ahead capability. These additional "dialed ahead" digits are saved for use by subsequent *collect digits* commands, and they provide the caller with a means to bypass subsequent unwanted announcement prompts. A single "#" digit can be collected and tested by subsequent *route-to...if digits* or *goto...if digits* commands. Alternately, any collected digits (whether collected from callers or CINFO) can be passed to a host via ASAI or forwarded to another site via Information Forwarding. Collection of dial-ahead digits continues until one of the following occurs:

- Vector processing stops or is terminated.
- The sum of the digits collected for the current *collect digits* command and the dial-ahead digits exceeds the switch storage limit of 24. Any additional dialed digits are discarded until storage is freed up by a subsequent *collect digits* command.

NOTE:
Any asterisk (*) or pound sign (#) digits count towards the 24-digit limit, as do any dial-ahead digits entered after the asterisk or pound sign digit.

- The TTR required by the touch-tone phone user to collect digits is disconnected. This occurs under the following conditions:
  - — Successful or unsuccessful *route-to number* step is encountered during vector processing except where the number routed to is a VDN extension.
  - — Successful or unsuccessful *route-to digits* step is encountered during vector processing except where the number routed to is a VDN extension.
  - — Successful or unsuccessful *adjunct routing* step is encountered during vector processing.
  - — Successful or unsuccessful *converse-on* step is encountered during vector processing.
  - — 10 second timeout occurs, during which time the caller does not dial any digits, asterisks (*) or pound signs (#).
  - — A collect ced/cdpd digits step is processed.

NOTE:
When the TTR is disconnected due to a *route-to number, route-to digits, converse-on*, or an *adjunct routing* step, *all dial-ahead digits are discarded*. This means that, following a failed *route-to, converse-on* or *adjunct routing* step, a subsequent *collect digits* step always requires the caller to enter digits.

NOTE:
Dial-ahead digits are available for use only by subsequent *collect digits* commands. The digits are never used by other vector commands that operate on digits (for example, *route-to digits*, *goto...if digits*, etc.). In addition, these digits are not displayed as part of the CALLR-INFO button operation until they are collected via a *collect digits* command.

Collecting Caller Information Forwarding (CINFO) Digits

The collect digits step allows you to collect CINFO Digits from the network. When a *collect ced digits* or *collect cdpd digits* step is processed, the system retrieves the first sixteen ced or cdpd digits from the ISDN User Entered CODE (UEC) Information Element that is associated with the call. It places the digits in the

collected digits buffer. Any digits that were in the collected digits buffer when the ced or cdpd digits are collected, are erased. If a TTR was connected to the call from a previous *collect digits* step, it is disconnected.

If the ced or cdpd digits contain invalid digits (not 0-9, *, #) the digits are not placed in the collected digits buffer. However, the collected digits buffer is still cleared and if a TTR is attached it is disconnected.

If no ced or cdpd digits were received from the network, when the *collect ced digits* or *collect cdpd digits* step is reached, the step is skipped. However, the collected digits buffer is still cleared and if a TTR is attached it is disconnected.

A * in the collected digits is treated as a delete character. Only the digits to the right of the * are collected. A # is treated as a terminating character. Only the # and the digits to the left of the # are collected. If a single # is sent, it is placed in the collected digits buffer.

The number of ced or cdpd digits to collect cannot be specified in the *collect digits* step. If there are 16 or fewer digits, all the digits are collected. If there are more than 16 digits, the first 16 digits are collected and a vector event is generated.

The CINFO ced and cdpd digits can be used with any vector step that uses the digits in the collected digits buffer.

Once ced or cdpd digits are collected, they can be displayed on a two-line display, or using the callr-info button.

Answer Supervision Considerations

Answer supervision is provided as soon as a TTR is connected and processing of the command starts. The command always provides answer supervision to an incoming trunk if supervision has not been previously provided except that a collect ced/cdpd digits step does not return answer supervision.

Feature Interactions

For BSR and LAI, the command is considered a call acceptance vector command except for collect ced/cdpd digits which is neutral.

CMS/BCMS Interactions

Collected digits are passed to CentreVu®CMS when the *collect* step is processed. Digits are not passed to BCMS.

Consider Command

consider

Purpose

Defines the resource (split, skill, or location) that is checked as part of a Best Service Routing™ (BSR) consider series and obtains the data BSR uses to compare resources. After the consider series has been executed, a *queue-to best* or *check best* command can queue the call to the best resource identified. If the *consider* commands are in a status poll vector, a *reply-best* step returns the data for the best resource found to the primary vector on the origin switch.

Syntax

BSR is available in single- and multi-site versions. The allowable syntax of the *consider* command varies according to which version is installed on your switch.

Single-site BSR

consider split <split#> pri <priority level> adjust-by <user adjustment>

consider skill <skill#> pri <priority level> adjust-by <user adjustment>

Multi-site BSR

consider split <split#> pri <priority level> adjust-by <user adjustment>

consider skill <skill#> pri <priority level> adjust-by <user adjustment>

consider location <location #> adjust-by <user adjustment>

Valid Entries

split #: *1 through 99* (vs/csi/si), *1 through 600* (r)

skill #: *1 through 99* (vs/csi/si), *1 through 600* (r), *1st, 2nd, 3rd* (VDN skill)

priority level: *l* (low), *m* (medium), *h* (high), *t* (top).

user adjustment: *0–100*

location #: *1–255*

Requirements

For additional switch requirements, see "Switch Requirements" on page 8-5.

Single-site BSR

- Vectoring (G3V4 Advanced Routing)
- Vectoring (Best Service Routing)

Multi-site BSR

Multi-site BSR requires ISDN-BRI or -PRI connectivity between switches, in addition to

- Vectoring (G3V4 Advanced Routing)
- Vectoring (Best Service Routing)
- Look-Ahead Interflow (LAI)

Examples

consider split *3* pri *m* adjust-by *0*

consider split *111* pri *h* adjust-by *20*

consider skill *11* pri *l* adjust-by *50*

consider skill *2nd* pri *t* adjust-by *100*

consider location *212* adjust-by *30*

Operation

In order to deliver a call to the resource that can provide the best service, *consider* commands collect and compare information. Whether you use single-site BSR, multi-site BSR, or both, consider steps work very much the same.

Each *consider* command collects status data from one split or skill. Splits or skills on the same switch are identified by number. Remote locations must be identified by a location number assigned on the BSR Application form. See *"Multi-site BSR Applications" on page 8-26* for more information.

Consider commands are typically written in a series of two or more steps called a "consider series." The first step in a consider series collects status data from the resource (a split, skill, or location specified by the user in the command) and saves this data to a buffer. The next *consider* step collects status data on its assigned split or skill and compares the data to that already in the buffer. If the

existing data in the buffer indicates the first split can provide better service to the call, the data for the first split/skill remains in the buffer as the “best” data. If the second split/skill can provide better service to the call, its status data replaces the data already in the buffer. Each subsequent step works similarly, collecting data from one resource, comparing it to the “best” data found up to that point, and replacing the best data only if the resource tested by the current step can provide better service to the caller. This series ends when a *queue-to best* or *check-best* step delivers or queues the call, or when a *reply-best* step returns the data for the best resource to a primary vector on the origin switch.

The first consider step in a series shortens the call vectoring 7-step timeout from 1.0 to 0.2 seconds. The timeout is shortened for BSR vectors only (that is, vectors that use *consider* series) in order to reduce real-time delays for call processing and reduce the incidence of race conditions in multi-site BSR applications.

User Adjustments

You may have preferences as to which splits or skills should answer certain types of calls. In both single- and multi-site BSR, the *adjust-by* portion of the *consider* command allows you to program these preferences into your vectors.

If a resource does not have an available agent when its *consider* step tests it, the *consider* step collects the Expected Wait Time (EWT) were the call to be queued to that resource. You can adjust this EWT value, for purposes of calculation only, by assigning a value of 0–100 in the user adjustment. The units of this value are supplied by the switch depending on the conditions whenever that *consider* step executes. For example, in the command *consider split 1 pri h adjust-by 20,* DEFINITY ECS interprets *adjust-by 20* to mean “add 20% to the EWT, but add at least 20 seconds.”[1] For Expected Wait Times of 1–100 seconds, an adjustment of 20 will therefore add 20 seconds. Above 100 seconds, the same adjustment will add 20% to the EWT for the split or skill specified in the *consider* step. Table A-12 shows the results of applying a constant adjustment to a range of Expected Wait Times.

1. If the user adjustment were defined as a number of seconds, BSR would not be efficient when EWT was high. If the user adjustment were defined as a percentage, BSR would not be efficient when EWT was low. Such efficiencies, while always important, become critical in multi-site BSR applications where issues of trunk cost and capacity are involved.

Table A-12. User Adjustments in BSR

EWT of resource (sec.)	User adjustment	Adjustment applied by DEFINITY ECS (sec.)	Adjusted EWT used to select resource
10	20	20	30
60		20	80
100		20	120
200		40	240
300		60	360

User adjustments work in single-site and multi-site BSR. In multi-site BSR applications, user adjustments may be applied at the remote switches in an application as well as at the origin switch. Since a status poll vector uses *consider* steps to evaluate resources on the switch where it resides, the *adjust-by* portion of each *consider* command allows the administrator at each switch to set preferences for the splits or skills at that switch. In BSR applications, any such adjustment for a split or skill is considered by the status poll vector in selecting the best resource on its switch. The adjustment is then returned to the origin switch along with the other data for that resource. When DEFINITY ECS receives this adjustment from the remote switch, it adds it to any adjustment that was assigned to that location in the *consider location* step. Table A-13 shows how user adjustments at an origin and a remote switch are added to yield the adjusted EWT for each location. For EWTs above 100 seconds, remember that DEFINITY ECS interprets the *adjust-by* value as a percentage. Thus, for example, each user adjustment for location 5 increases the actual EWT value by 30% of 200, or 60 seconds.

Table A-13. Samples of Additive User Adjustments

Location	Actual EWT of remote best (sec.)	User adjustment on origin switch	User adjustment on remote switch	Adjustment applied by origin switch (sec.)	Adjusted EWT used in BSR calculations (sec.)
1	20	30	30	60	80
2	50	30	30	60	110

Continued on next page

Table A-13. Samples of Additive User Adjustments — *Continued*

Location	Actual EWT of remote best (sec.)	User adjustment on origin switch	User adjustment on remote switch	Adjustment applied by origin switch (sec.)	Adjusted EWT used in BSR calculation s (sec.)
3	70	30	30	60	130
4	100	30	30	60	160
5	200	30	30	120	320
6	300	30	30	180	480

Events that Clear "Best" Data

As the steps in a consider series execute, the status data for the best resource found is kept in a buffer. This "best data" is unaffected by some call processing events and vector commands, while other events and commands initialize (clear) this buffer. Table A-14 shows you what initializes the best data buffer and what doesn't.

Table A-14. Initialization of BSR "Best Data"

Events and Vector Commands that Clear Best Data	Events and Vector Commands that Do Not Clear Best Data
Execution of any queue-to or check command	Converse command

Continued on next page

Table A-14. Initialization of BSR "Best Data"

Events and Vector Commands that Clear Best Data	Events and Vector Commands that Do Not Clear Best Data
Vector processing terminates: ■ reply-best command executes ■ agent answers ■ successful route-to command ■ successful adjunct routing command ■ successful messaging split/skill command ■ vector disconnect timeout ■ disconnect command ■ busy command ■ vector processing reaches last step without call in queue	Announcement command
	Collect Digits command
	Unsuccessful execution of a messaging split/skill command
	Unsuccessful adjunct routing command
	Goto step/vector with any conditional
	Wait command (with any feedback)
	Unsuccessful route-to command
	Vector processing reaches last step while call is still in queue
	Execution of a consider step (this will either replace the current best data with new data or leave the current data untouched)

Recommendations

Lucent recommends that you follow the guidelines below when using *consider* commands:.

- Don't put a consider series in vector loops.
- Don't put any commands between the steps of a consider sequence that would cause a delay. *Announcement* and *wait* commands, for example, should not be used within a consider sequence. *Goto* commands are OK.
- Arrange your *consider* steps in order of preference.

  The *consider* step that tests the main, or preferred, resource should be the first in the series. The second *consider* step should test the resource that is your second preference for handling the given call type, and so on. To avoid unnecessary interflows, put *consider* steps for local resources before steps that consider remote resources. Arranging *consider* steps in order of preference is recommended for all BSR vectors. It's especially important when the active VDN for the call is using the 1st-found agent strategy: since DEFINITY ECS will deliver the call to the first available

agent found, arranging *consider* steps in order of preference will ensure that calls are delivered to the best of the available resources and that unnecessary interflows are avoided.

Answer Supervision Considerations

All forms of the *consider* command are ISDN neutral and do not return answer supervision.

Feature Interactions

Splits used in *consider* commands must be vector-controlled.

CMS/BCMS Interactions

Only R3V6 or later versions of CMS may be used with Best Service Routing.

BCMS does not log LAI attempts. Therefore, it will not log BSR status polls since they are LAI attempts.

Converse-on Command

```
converse-on split (skill)
```

Purpose

Delivers a call to a converse split (skill) and activates a voice response script that is housed within a Voice Response Unit (VRU).

Syntax

converse-on split <split #> pri <priority level> passing <data_1> and <data_2>

converse-on skill <skill #> pri <priority level> passing <data_1> and <data_2>

Valid Entries

split #: *1 through 99* (vs/csi/si), *1 through 600* (r)

skill #: *1 through 99* (vs/csi/si), *1 through 600* (r), *1st, 2nd, 3rd* (VDN skill)

priority level: *l* (low), *m* (medium), *h* (high), *t* (top).

data_1, data_2: String consisting of any digits *0-9* and/or of one or more asterisks *(*)* and not exceeding a total of six such digits and/or asterisks; the pound sign *(#)*; any of the following keywords: *vdn, ani, digits, qpos, wait, none* (with the exception that *none* cannot be included as an entry for data_1 if data_2 has an entry other than *none*).

Requirements

Basic Call Vectoring software must be installed, and the converse split must be vector-controlled. If the keyword *digits* is included within the command syntax, Call Prompting software must be installed. Moreover, Call Prompting software is necessary to allow for the full functionality of Voice Response Integration (VRI). If the command passing Caller Information Forwarding (CINFO) digits, Vectoring (CINFO) must be enabled. Also, skills replace splits only if Expert Agent Selection (EAS) is optioned. To include the keyword *wait* in the command syntax, the “Vectoring (G3V4 Advanced Routing) customer option must be enabled.

Examples

converse-on split *1* pri *h* passing *none* and *none*

converse-on split *20* pri *m* passing *123456* and *none*

converse-on skill *57* pri *h* passing *vdn* and *ani*

converse-on skill *3rd* pri *t* passing *digits* and *qpos*

converse-on skill *23* pri *h* passing *wait* and *none*

Operation

NOTE:
Refer to Appendix I for details regarding call flows, data passing, collection, and return specifications involving the *converse-on* command.

The *converse-on* command is designed primarily to integrate Voice Response Units (VRUs), principally the CONVERSANT Voice Information System (VIS), with the DEFINITY System. The command effects data passing between the DEFINITY System and the VRU, and it enables the caller to hear the appropriate voice response script housed in the VRU.

If the command is successful, it delivers the call to a predetermined split (skill), which is referred to as the converse split (skill). Once the call is answered by the VRU, the command may or may not pass data to the VRU (depending upon the parameters of the command). Regardless of whether or not data is passed, the caller is then connected to the VRU, which in turn executes the voice response script. If by this time the call has already queued to a nonconverse split (skill), the call retains its position in the nonconverse split (skill) queue. If an agent from the nonconverse split (skill) becomes available to service the call while the voice response script is being executed, the DEFINITY System drops the line to the voice information system and connects the caller to the available agent. The voice information system, in turn, detects the disconnect and terminates the voice response script. Whenever a voice response script is executed, any audible feedback provided by the vector is disconnected, and no further vector steps are executed until the voice response script is executed.

The VRU may or may not eventually return data to the switch. If, once the voice response script is completed, there is no data to be returned from the voice information system to the DEFINITY System, the VRU drops the line to the DEFINITY System, and vector processing is reactivated on the switch.

If there is data to be returned to the switch, the “Converse data return code” is outpulsed before the data to be passed is outpulsed. Once all VRU data is received, it is stored in the Call Prompting digits buffer as dial-ahead digits, and

vector processing is reactivated. Digits returned by the voice information system are not heard by the caller.

Digits returned from the VRU can be:

- Displayed on the answering agent's display set (automatically for 2-line displays, or by using the "CALLR-INFO" button for 1-line displays)
- Treated as an extension in a *route-to digits* step
- Used for vector conditional branching in a step containing a command with the *if digits* parameter
- Tandemed to an ASAI host

The DEFINITY System can be set up to pass information in-band to the voice information system. In such a case, the *converse-on* command can outpulse up to two groups of digits to the voice information system. The digits may serve two major purposes: the digits may notify the voice information system of the application to be executed, and they may share call related data, such as ANI (BN) or caller digits collected by the DEFINITY System. (In many applications, both application selection and data sharing are required.) The touch tone outpulsing rate is adjustable see Appendix I, "Detailed Call Flow and Specifications for Converse—VRI Calls" for details.

Since in many cases the digit strings are of variable length, the DEFINITY System always appends a pound sign (#) character to the end of each digit string. The *Prompt and collect* steps in the voice response script must therefore always be administered to expect "#" as the end-of-string symbol and to include "#" in the digit count.

The sending of "#" prevents excessive delays caused by digit timeouts, and it prevents other problems caused by timeouts. It also ensures that each data field is used to satisfy a single *prompt and collect* step.

Any data passed from the DEFINITY System to a VRU is outpulsed in-band. The user can administer two time delays on the System Parameter Features form: "converse first data delay" and "converse second data delay." These delays may range from 0 to 9 seconds with a default of zero seconds for the converse first data delay and a default of two seconds for the converse second data delay. The delays are needed to give the VRU time to invoke an application and to allocate a touch-tone receiver to receive the passed digits.

NOTE:
No time delays are invoked when the keyword "none" is administered.

If <data_1> is not "none," the converse first data delay timer starts when the call is answered by the VRU. When the timer expires, the <data_1> digits are outpulsed in-band to the VRU. The end-of-string character (#) is then outpulsed.

If <data_2> is not “none,” the converse second data delay timer starts when the end-of-string character (#) from the first digit string is outpulsed. When the timer expires, the <data_2> digits are outpulsed in-band to the VRU. The end-of-string character (#) for the second digit string is then outpulsed. The following values may be administered for <data_1> and <data_2> within the *converse-on* command:

- **Administered digit string**: This string can contain up to six characters consisting of one or more digits (0 through 9) or asterisks (*). The pound sign (#) may not be included in a digit string because it is reserved as the end-of-string character. However, a single “#” may be administered.
- **ani**: If the call is an internal call or an incoming DCS call, this data type causes the extension of the calling party to be outpulsed. If the call is an incoming ISDN-PRI or R2MFC Signaling call with ANI (BN) provided to the DEFINITY System, the calling party number/billing number (CPN/BN) of the calling party is outpulsed to the voice information system. If there is no ANI (BN) to send, the end-of-string pound sign (#) is the only character outpulsed. Any other type of incoming call results in “#” being outpulsed.
- **digits**: This data type can be used only if Call Prompting is optioned. To pass CINFO digits, Vectoring (CINFO) must also be enabled. The digits data type causes the most recent set of digits collected in vector processing, either from the caller or from the network, to be outpulsed. If no digits are available, the end-of-string pound sign (#) is the only character outpulsed.
- **none**: This data type causes no characters to be outpulsed. Also, no end-of-string pound character (#) is outpulsed, and no time delays are invoked.
- **qpos**: This data type causes the value of the queue position of a call in a nonconverse split to be outpulsed. This value is a variable length data item from which between one and three digits can be outpulsed. If the call is not queued, the end-of-string pound sign (#) is the only character that is outpulsed.

  NOTE:
  The use of this keyword is not recommended with multiple split (skill) queuing because any queue position value that is sent may not be meaningful. If the call is queued to multiple nonconverse splits (skills), the value of the caller’s queue position in the first nonconverse split (skill) is sent.

  This data may be used by the voice information system to inform callers of their position in queue or to decide whether to execute a long or short version of a voice response script.
- **vdn**: This data type causes the VDN extension to be outpulsed. In cases where multiple VDNs are accessed, normal VDN override rules determine which VDN extension is outpulsed.

- **wait:** This data type can be used only if the "Vectoring (G3V4 Advanced Routing)" customer option is enabled. It causes the expected wait time of the call in seconds to be outpulsed. See "Expected Wait Time (EWT)" on page 5-2 for a detailed description of expected wait time. If the call is not queued or if it is queued only to splits that are unstaffed or splits where all agents are in AUX work mode, the end-of-string character "#" is the only character outpulsed.

  The value outpulsed is a variable number not padded with zeroes. It is a maximum of four digits always followed by #. The range is 0# to 9999# or a single #.

- #: This is the only character outpulsed. Outpulsing this character causes the corresponding *prompt and collect* command in the voice response script to be skipped.

The DEFINITY System always outpulses a pound character (#) at the end of each digit string. Where "#" is administered, or where the "digits" keyword is administered and the last digit collected from the caller is "#," only one "#" is outpulsed. No "#" is outpulsed when the keyword "none" is administered.

If *data_1* is administered as "none," *data_2* must also be "none."

Answer Supervision Considerations

Answer supervision is returned only once during the life of a call. If a call is answered as a result of a *converse-on* step, answer supervision is sent only if it has not been sent previously. If digits are passed to the VRU, answer supervision is not sent until after the digits are outpulsed.

Feature Interactions

- Abandon Call Search

  If the *converse-on* step places a call to a hunt group, and if the incoming call was placed via a trunk group with Abandon Call Search activated, the system checks that the calling party has not abandoned the call (that is, hung up) before terminating to an agent.

- Adjunct Switch Applications Interface (ASAI)

  Since vector-controlled splits (skills) cannot be ASAI-monitored domains, ASAI cannot be used to supplement the operation of the *converse-on* step.

  If a *converse-on* step places a call to an ASAI-monitored domain, ASAI event messages are sent over the ASAI link.

Whenever a *converse-on* step places an ASAI-monitored call, the ALERTing message sent to the ASAI host includes a Cause IE, Coding Standard 3 value 23 (CS3/23). This informs the ASAI host that the call has not been de-queued from any nonconverse splits (skills).

If a *converse-on* step is executed while an adjunct routing request is outstanding, the route request is canceled.

- Audio Information Exchange (AUDIX)

  If a *converse-on* step calls AUDIX, the call is treated as a direct call to AUDIX. The caller hears the “welcome to AUDIX” message and may retrieve his or her messages in the usual manner.

  If a call is forwarded to or covers to a VDN and is then delivered to an AUDIX hunt group by a *converse-on* step, the call to AUDIX is treated as a redirected call, and the caller may leave a message for the principal.

- Auto-Available Splits (Skills)

  A *converse-on* step may place a call to an auto-available split (skill). Except in cases where the converse split (skill) is ASAI-controlled, auto-available converse splits (skills) are recommended for Voice Response Integration (VRI).

- Basic Call Management System (BCMS)

  BCMS tracks calls placed by a *converse-on* step to a BCMS-measured hunt group. Since with the *converse-on* step it is now possible for a call to be “answered” in more than one split (skill), trunk totals may no longer match split totals. However, VDN totals and trunk totals will match.

- BCMS VDN Reports

  For call tracking in BCMS VDN reports, a *converse-on* step is treated like an *announcement* step. A call is considered “answered” when it is answered by a nonconverse split (skill) but never when it is answered by a converse split (skill).

- Call Coverage

  Call Coverage does not apply because the *converse-on* step may deliver calls only to vector-controlled splits (skills), which do not have coverage paths.

- Call Detail Recording

  For incoming calls to a VDN, the duration of the call is recorded from the time answer supervision is returned. Answer supervision is returned for a successful *converse-on* step. No ineffective call attempt records are generated for *converse-on* steps that fail. Also, no outgoing calls can be placed by a *converse-on* step.

- Call Management System (CMS)

  CMS tracks calls placed by a *converse-on* step to a CMS-measured hunt group or split. *converse-on* vector steps may be administered from CMS 3.1 and later releases. Since with the *converse-on* step it is now possible for a call to be “answered” in more than one split (skill), trunk totals no longer match split totals. However, VDN totals and trunk totals will match.

- Call Park

  Calls placed by a *converse-on* step may not be parked.

- Call Pickup

  Calls placed by a *converse-on* step ringing at an agent station may be picked up if that agent is part of a pickup group. Subsequent transfers are denied.

- Call Prompting

  The Call Prompting customer option must also be enabled to gain full VRI functionality. Without Call Prompting, any data returned by the voice information system cannot be collected and processed by the switch.

  If the *converse-on* step places a call to a split (skill) of live agents, any digits collected previously may be displayed by agents using the callr-info button.

- Call Vectoring—Basic

  The *converse-on* step is an enhancement to the Basic Call Vectoring customer option. This option must be enabled in order to invoke the VRI feature.

- Class of Restriction (COR)

  As is the case for the *queue-to split (skill)* and *check split (skill)* vector steps, no COR checking is carried out when a *converse-on* step places a call to a split (skill).

- Conference

  Any attempt to conference a call placed by a *converse-on* step is denied.

- Coverage Callback

  A call placed by a *converse-on* step does not follow any coverage paths. Therefore, Coverage Callback is not available. Also, if a call reaches a *converse-on* step via a VDN in a coverage path, coverage callback cannot be used.

- Direct Department Calling (DDC)

  A converse split may be administered as a direct department calling split.

- Distributed Communications System (DCS)

  If an incoming DCS call is placed to a vector with a *converse-on split (skill) x pri y passing ani ...* step, the DCS extension of the calling party is outpulsed.

- Priority Levels

  A call placed by a *converse-on* step may be queued at one of four priority levels: low, medium, high or top.

- Hunt Groups

  The *converse-on* step may deliver a call to a vector-controlled hunt group, ACD split (skill), Message Center or AUDIX hunt group.

- Integrated Services Digital Network (ISDN)

  The *converse-on* step may be administered to outpulse to the voice information system the ANI (calling party number/billing number CPN/BN) of the calling party via use of the “ani” keyword.

- Intercept Treatment

  A caller is never given intercept treatment upon execution of a *converse-on* step. Failing to place a converse call successfully results in the failure of the *converse-on* step. Vector processing continues at the next vector step.

- Interflow

  Since a *converse-on* step can place calls only to hunt groups that are vector-controlled, and since the activation of Call Forwarding for a vector-controlled hunt group is blocked, calls placed by a *converse-on* step to a hunt group cannot interflow.

- Intraflow

  Since a *converse-on* step can place calls only to hunt groups that are vector-controlled (that is, without coverage paths), intraflow is not possible.

- Live Agents

  Although not recommended, the switch does not prevent a *converse-on* step from delivering a call to a group of live agents. To the agent, the call looks like any other ACD call. However, certain features, such as call transfer, conference, and supervisor assist are denied.

  The answering agent can display any digits collected prior to executing the *converse-on* step by using the “callr-info” button.

- Look-Ahead Interflow

  If a call placed by a *converse-on* vector step is answered by a VRU, or if such a call queues to a split (skill) on the receiving switch while a Look-Ahead Interflow call attempt is outstanding, the Look-Ahead Interflow call attempt is accepted.

  A *converse-on* step that fails is neutral.

- Message Center

  The *converse-on* step may deliver calls to message hunt groups. Such calls are treated as direct calls to the message.

  If a call is forwarded to a VDN and then delivered to a message split by a *converse-on* step, the call is treated as a redirected call.

- Multiple Split (Skill) Queuing

  A call can be queued to three different splits (skills) and then to a converse split (skill) as a result of a *converse-on* step.

- Music on Hold

  During the data return phase of a *converse-on* step, the caller is temporarily placed on hold. Music on hold, if administered, is suppressed.

- Non-Vector Controlled Splits (Skills)

  A *converse-on* step may not place a call to a nonvector-controlled split (skill).

- Priority Queuing

  The queue priority of a call placed by a *converse-on* step is administrable on the vector step.

- Queue Status

  All queue status display, queue status indication and queue warning wall lamp feature capabilities also apply to calls queued by the *converse-on* command.

- Queuing

  Calls handled by the *converse-on* step queue when they are delivered to busy hunt groups. Call Vectoring audible feedback is not disconnected while a converse call is in queue.

  If a *converse-on* step is executed while a call is queued to a nonconverse split (skill), the call remains in queue for the nonconverse split (skill).

  The queue priority of the call is administrable on the vector step.

- Recorded Announcement

  VRI may be used to increase the system's recorded announcement capacity by off-loading some recorded announcements to the VRU. Callers can be redirected by the *converse-on* step to a group of VRU ports and use data passing to specify the correct announcement to play.

- Redirection on No Answer (RONA)

  If a *converse-on* step places a call to a hunt group with a "no answer timeout" administered, and if the call rings at an agent terminal/port for longer than the administered timeout, the call is redirected, and the agent/port is put into the AUX work state (or logged out if the agent is a member of an auto-available split (skill)).

Thereafter, under RONA, the call is requeued to the split (skill) unless there is no room in the queue or unless this is an auto-available split (skill) whose agents are all logged out. If the call cannot be requeued, the *converse-on* step fails, a vector event is logged, and vector processing is restarted at the next vector step.

- Service Observing

  Calls placed by a *converse-on* step may be service observed. To prevent the observer from hearing tones being outpulsed to the VRU, the observer is not connected to the call until the data passing phase is complete. If data is returned by the VRU, the observer is put in service observing pending mode, and the calling party is temporarily put on hold while the VRU digits are outpulsed. Upon completion of the converse session, and once the VRU hangs up the line, the observer remains in service observing pending mode.

  It is not recommended that a service observing warning tone be administered since the warning tone may interfere with the interaction between the voice information system and the calling party.

- System Access Terminal (SAT)

  converse-on steps may be administered from the SAT terminal.

- System Measurements

  System measurements track converse calls to hunt groups and attendant groups.

- Timed After Call Work (ACW)

  Timed ACW cannot be assigned to auto-available splits (AAS). If a call to a VDN with Timed ACW routes to a converse split, the VDN Timed ACW does not apply.

  If Timed ACW is assigned to a non-AAS split that is a converse split, the Timed ACW of the split does apply.

- Touch-Tone Dialing

  Any touch-tone dialing by the calling party during the digit passing phases of a session involving a *converse-on* step does not result in corruption of data or in the collection of this data in the form of dial-ahead digits by the DEFINITY System.

  Only after the digit passing phase from the DEFINITY System to the voice information system is completed can the calling party enter touch-tone digits in response to a voice information system prompt. Only after the voice information system to the DEFINITY System data return phase is completed and an additional *collect digits* vector step is executed can the calling party enter a touch-tone response to a DEFINITY System prompt.

- Transfer

  A call placed by a *converse-on* step may not be transferred. The only form of transfer allowed is the data passing operation during the data return phase at the end of a voice response script.

  If an illegal attempt to transfer a converse call is made, a vector event is logged, the line to the voice information system is dropped, and vector processing is reactivated at the next vector step.

  If an illegal transfer is attempted by a live agent with a multifunction set, the transfer is denied and the agent may reconnect to the call.

- Transfer out of AUDIX

  If a *converse-on* step delivers a call to an AUDIX hunt group, and if the calling party then attempts to transfer out of AUDIX, the transfer fails, and vector processing is reactivated at the next vector step.

- Uniform Call Distribution (UCD)

  A converse split (skill) may be administered as a Uniform Call Distribution split (skill).

- VDN as a Coverage Point

  If a call covering to a VDN is processed by the *converse-on* command and subsequently reaches a station user (that is, a member of a “converse split (skill)”), and if the “converse split (skill)” agent attempts to activate Consult (coverage), or Coverage Leave Word Calling, any of these coverage attempts is denied because the call is still in vector processing. If the “converse split (skill)” is an AUDIX/Message Center split (skill), the call covered to the VDN is treated like a redirected call to AUDIX/MCS; the original principal and reason for redirection is used in the same manner as a Call Forwarded call to a VDN.

- VDN Override

  If a call that accesses multiple VDNs encounters a *converse-on* step passing “vdn,” normal override rules determine which VDN number is outpulsed to the VRU.

- VDN Reports

  For call tracking in CMS and BCMS VDN reports, a *converse-on* step is treated like an *announcement* step. A call is considered “answered” when it is answered by a nonconverse split (skill) but never when it is answered by a converse split (skill).

- Vector-controlled Splits (Skills)

  A *converse-on* step may place a call to a split (skill) only if that split (skill) is administered as a vector-controlled split (skill).

CMS Interactions

CMS tracks calls placed by a *converse-on* step to a CMS-measured split (skill). *converse-on* vector steps may be administered from R3 CMS and later releases. Since with the *converse-on* step it is now possible for a call to be "answered" in more than one split (skill), trunk totals no longer match split (skill) totals. However, VDN totals and trunk totals will match.

For call tracking in CMS VDN reports, a *converse-on* step is treated like an *announcement* step. A call is considered "answered" when it is answered by a nonconverse split (skill) but never when it is answered by a converse split (skill).

BCMS Interactions

BCMS tracks calls placed by a *converse-on* step to a BCMS-measured split (skill). Since with the *converse-on* step it is now possible for a call to be "answered" in more than one split (skill), trunk totals may no longer match split totals. However, VDN totals and trunk totals will match.

For call tracking in BCMS VDN reports, a *converse-on* step is treated like an *announcement* step. A call is considered "answered" when it is answered by a nonconverse split (skill) but never when it is answered by a converse split (skill).

Disconnect Command

```
disconnect
```

Purpose

Ends treatment of a call and removes the call from the switch. Also allows the optional assignment of an announcement that will play immediately before the disconnect.

Syntax

disconnect after announcement <extension>

Valid Entries

extension: *none* or valid announcement extension

Requirements

Basic Call Vectoring software must be installed. Also, the relevant announcements must be administered and recorded.

Example

disconnect after announcement 2556

Operation

While the command's optional announcement is playing, the call remains in queue and can be connected to an agent. When the announcement completes (or is not specified), the command forces a disconnect, ends the treatment of the call, and removes the call from the switch.

Answer Supervision Considerations

If the switch has not yet sent answer supervision, the switch does so immediately before disconnecting the call, whether an announcement is specified or not. If an announcement is specified, answer supervision is given before an attempt is made to connect the announcement. The exception is for ISDN calls, where the disconnect can occur without answer supervision being sent when an announcement is not played.

Feature Interactions

For Look-Ahead Interflow, the command can be considered either a call acceptance vector command or a call denial vector command.

The command is considered a call acceptance vector command whenever an announcement is included within the command and one of the following is true:

- Announcement is available.
- Call is queued for an announcement.
- Announcement is retried.

The command is considered a call denial vector command whenever one of the following is true:

- No announcement is included within the command.
- Announcement is included within the command, but the announcement is unavailable.

CMS Interactions

R3 CMS:

Disconnect Command

Database Item	Report Heading
DISCCALLS/DISCTIME	Calls Forced Disc
	Calls Busy/Disc
OTHERCALLS/OTHERTIME	Inbound Other Calls
INTIME	Avg Time In Vector

DISCTIME, OTHERTIME, and INTIME for splits and vectors are tracked according to when the announcement starts. DISCTIME, OTHERTIME and INTIME for VDNs are tracked according to when the trunk idles.

R2 CMS: Calls given forced disconnect are tracked in the FDISCCALLS and FDISCTIME database items. These calls are not shown in standard reports. FDISCTIME is tracked according to when the announcement starts. VECTIME is recorded for the vector, and it is reported as “Avg Time in Vector.”

BCMS Interactions

A call that is disconnected via the command is tracked as “OTHER” in the VDN Report.

Goto Step Command

```
goto step
```

Purpose

Allows conditional or unconditional movement (branching) to a preceding or subsequent step in the vector.

Syntax

goto step <step #> if **unconditionally**

goto step <step #> if **digits** <comparator> <digits>

goto step <step #> if **digits** <option> table <table>

goto step <step #> if **ani** <comparator> <digits>

goto step <step #> if **ani** <option> table <table>

goto step <step #> if **ii-digits** <comparator> <digits>

goto step <step #> if **ii-digits** <option> table <table>

goto step <step #> if **time-of-day** is <day> <hour>: <minute> to <day> <hour>: <minute>

Conditions = available-agent, staffed-agents:

goto step <step #> if <condition> in split <split #> <comparator> <threshold>

goto step <step #> if <condition> in skill <skill #> <comparator> <threshold>

Conditions = calls-queued, oldest call-wait:

goto step <step #> if <condition> in split <split #> pri <priority level> <comparator> <threshold>

goto step <step #> if <condition> in skill <skill #> pri <priority level> <comparator> <threshold>

Condition = rolling-asa:

goto step <step #> if <condition> for split <split #> <comparator> <threshold>

goto step <step #> if <condition> for skill <skill #> <comparator> <threshold>

goto step <step #> if <condition> for vdn <vdn> <comparator> <threshold>

Condition = counted-calls:

goto step <step #> if <condition> to vdn <vdn> <comparator> <threshold>

Condition = expected-wait:

goto step <step #> if <condition> for split <split #> pri <priority level> <comparator> <threshold>

goto step <step #> if <condition> for skill <skill #> pri <priority level> <comparator> <threshold>

goto step <step #> if <condition> for call <comparator> <threshold>

goto step <step #> if <condition> for best <comparator> <threshold>

Condition = interflow-qpos:

goto step <step#> if interflow-qpos <comparator> <threshold>

Condition = wait-improved:

goto step <step #> if <condition> for split <split #> pri <priority level> <comparator> <threshold>

goto step <step #> if <condition> for skill <skill #> pri <priority level> <comparator> <threshold>

goto step <step #> if <condition> for best <comparator> <threshold>

Valid Entries

step #: *1-32*

split #: *1 through 99* (vs/csi/si), *1 through 600* (r)

skill #: *1 through 99* (vs/csi/si), *1 through 600* (r), *1st, 2nd, 3rd* (VDN skill)

condition:	comparator:[1]	threshold:
unconditionally	N/A	N/A
rolling-asa	>, >=, =, <>	*0-999* (1-second increments)
	<, <=	*0-999* (1-second increments)
available-agents	>, >=, =, <>	*0-199* (vs/csi/si), *0-998* (r)
	<, <=	*1-200* (vs/csi/si), *1-999* (r)
calls-queued	>, >=, =, <>	*0-199* (vs/csi/si), *0-998* (r)
	<, <=	*1-200* (vs/csi/si), *1-999* (r)
counted-calls	>, >=, =, <>	*0-999*
	<, <=	*1-999*
expected-wait	>, >=, =, <>	*0-9999* seconds
	<, <=	*1-9999* seconds
interflow-qpos	>, >=, =, <>, <, <=	1–9 (position in eligible queue)
oldest call-wait	>, >=, =, <>	*0-999* seconds (1-second increments)
	<, <=	*1-999* seconds (1-second increments)
staffed-agents	>, >=, =, <>	*0-199* (vs/csi/si), *0-998* (r)
	<, <=	*1-200* (vs/csi/si), *1-999* (r)
wait-improved	>, >=, =. <>	*0-9999 seconds*
	<, <=	*1-9999 seconds*

1.Comparators =, >=, <=, <> are only available for these conditions with Vectoring (G3V4 Enhanced).

digits: the following values are accepted:

command	comparator[1]	value[2]
goto step <step#> if digits	<, <=, >, >=	String of 0-9
	=, <>	String of 0-9, #, +, ?, none
	in table, not-in table[3]	1–10 (vs/csi/si), 1–100 (r)
goto step <step#> if ani	<, <=, >, >=	String of 0-9
	=, <>	String of 0-9, #, +, ?, none
goto step <step#> if ii-digits	<, <=, >, >=	Two consecutive digits from string of 0-9
	=, <>	Two consecutive digits from string of 0-9, +, ?, none

1.Comparators >, >=, <, <=, <> are only available for the *goto step if digits* command with Vectoring (G3V4 Enhanced)

2.Wildcards (+, ?) can only be used with Vectoring (G3V4 Enhanced). The + represents a group of digits (0 or more) and can only be used as the first or last character of the string. Only one + can be used in any digit string. The ? represents a single digit. Any number of them can be used at any position in the digit string.

3.“Table” refers to a vector routing table. Vectoring (G3V4 Enhanced) must be enabled for the comparators “in” and “not-in” to be available.

priority level: *l* (low), *m* (medium), *h* (high), *t* (top)

day: *mon, tue, wed, thu, fri, sat, sun, all (that is, “on any day of the week”)*

hour: *00* to *23* (military format)

minute: *00* to *59* (military format)

vdn: assigned vdn extension, *active*, *latest*. Active is the active called VDN as modified by VDN override rules. Latest is the VDN assigned to the vector in which the call is currently being processed.

Requirements

- Basic Call Vectoring software must be installed for all the options.
- Call Prompting software is required for the digits option.
- Skills replace splits only if Expert Agent Selection (EAS) is optioned.

- Wildcard matching and the use of Vector Routing Tables require Vectoring (G3V4 Enhanced).
- To use the *rolling-asa*, *expected-wait*, and *counted-calls* conditions, the "Vectoring (G3V4 Advanced Routing)" customer option must be enabled.
- To use *ani* and *II-digits* routing, the "Vectoring (ANI/II-Digits Routing)" customer option must be enabled.
- 1-second increments for *oldest-call-wait* and *rolling ASA* require DEFINITY ECS R6.3 or later and CentreVu® CMS R3V6 or later.
- To use either the *wait-improved* conditional or the *best* keyword with *expected-wait*, Vectoring (Best Service Routing) must be enabled.
- To use the *interflow-qpos* conditional, Look-Ahead Interflow (LAI) must be enabled.

Examples

goto step *8* if **available-agents** in split *67 < 5*

goto step *12* if **calls-queued** in split *51* pri *t < 17*

goto step *7* if **time-of-day** is *mon 16:30* to *tue 7:30*

goto step *12* if **calls-queued** in skill *3rd* pri *t > 50*

goto step *8* if **available-agents** in skill *1st < 10*

goto step *11* if **oldest-call-wait** in split *26* pri *t >= 20*

goto step *6* if **rolling-asa** for split *9 > 30*

goto step *10* if **counted-calls** to vdn *5372 >= 50*

goto step *4* if **ani** *in* table *10*

goto step *9* if **ii-digits** *= 06*

goto step *11* if **expected-wait** for call *> 30*

goto step *9* if **wait-improved** for split *10* pri m *> 30*

goto step *9* if **wait-improved** for skill *10* pri m <= *30*

goto step *9* if **wait-improved** for best >= *30*

goto step 5 if **interflow-qpos** >= 5

Operation

If the command syntax includes **unconditionally**, the command always branches. The unconditional form of the command is commonly used for skipping vector commands as well as for looping through vector commands.

Otherwise, branching takes place according to one of the conditions that follow:

- The average speed of answer for the indicated split (skill) or VDN meets the constraints defined by the comparator and threshold value.
- Number of available agents in the indicated split (skill) meets the constraints defined by the comparator and the threshold value.
- Number of queued calls in the indicated split (skill) and at the specified priority level (or higher) meets the constraints defined by the comparator and the threshold value.
- The number of active calls in the indicated VDN meets the constraints defined by the comparator and the threshold value.
- The expected wait time at the specified priority level for the indicated split (skill), or for the call meets the constraints defined by the comparator and the threshold value.
- Oldest call-waiting in the indicated split (skill) at the specified priority level (or higher) has been waiting for a period of time within the constraints defined by the comparator and the threshold value, which is expressed in seconds.
- Number of staffed agents in the indicated split (skill) meets the constraints defined by the comparator and the threshold value.
- Digits collected via the *collect digits* command match the criteria defined by the comparator for the specified digit string. Or, the digits are found or not found, depending upon the option chosen, in the specified Vector Routing Table. The “#” digit can be tested against as a single digit.
- The ani digits match the criteria defined by the comparator for the specified digit string. Or, the ani digits are found or not found, depending upon the option chosen, in the specified Vector Routing Table.
- The II-digits match the criteria defined by the comparator for the specified digit string. Or, the II-digits are found or not found, depending upon the option chosen, in the specified Vector Routing Table.
- Time-of-day criteria are met.

NOTE:
The syntax for this condition can be illustrated by a couple of examples, as follows: *mon 8:01 to fri 17:00* means “anytime between 8:01 A.M. Monday through 5:00 P.M. Friday,” and *all 17:00 to all 8:00* means “between 5:00 P.M. and 8:00 A.M. on any day of the week.”

- The Expected Wait Time (EWT) for the call is decreased by a period of time within the constraints defined by the comparator and the threshold value, which is expressed in seconds. The improvement in EWT is defined by calculating the difference between the call's current EWT and its EWT were it to be queued to the resource specified in the command.
- The call's position in the interflow-eligible portion of the queue meets the condition defined by the comparator and the threshold value (representing queue position counting backward from "1," which is the head of the eligible queue).

Answer Supervision Considerations

The call answer is not affected by the command.

Feature Interactions

For Best Service Routing and Look-Ahead Interflow, the command is considered a neutral vector command in all cases. When a call experiences Look Ahead interflow, the ANI value is sent along with the call only for ISDN PRI calls. ANI is not sent for internal or DCS calls.

CMS/BCMS Interactions

The command is not tracked on CMS or BCMS.

The ANI and/or II-digits are passed to CMS when the call first starts vector processing if the following is true:

- Basic Call Vectoring and/or Call Prompting is optioned
- ANI is available from the network, the call is internal, or is received over DCS
- II-digits is available from the network
- The CMS is R3 (R3V5 for II-digits) or a later version

ANI and II-digits are not passed to BCMS.

Goto Vector Command

```
goto vector
```

Purpose

Allows conditional or unconditional movement (branching) to another vector. The goto vector step does not remove a call from queues in which it is already placed.

Syntax

goto vector <vector #> if **unconditionally**

goto vector <vector #> if **digits** <comparator> <digits>

goto vector <vector #> if **digits** <option> table <table>

goto vector <vector #> if **ani** <comparator> <digits>

goto vector <vector #> if **ani** <option> table <table>

goto vector <vector #> if **ii-digits** <comparator> <digits>

goto vector <vector #> if **ii-digits** <option> table <table>

goto vector <vector #> if **time-of-day** is <day> <hour> : <minute> to <day> <hour> : <minute>

Conditions = available-agent, staffed-agents:

goto vector <vector #> if <condition> in split <split #> <comparator> <threshold>

goto vector <vector #> if <condition> in skill <skill #> <comparator> <threshold>

Conditions = calls-queued, oldest call-wait:

goto vector <vector #> if <condition> in split <split #> pri <priority level> <comparator> <threshold>

goto vector <vector #> if <condition> in skill <skill #> pri <priority level> <comparator> <threshold>

Condition = rolling-asa:

goto vector <vector #> if <condition> for split <split #> <comparator> <threshold>

goto vector <vector #> if <condition> for skill <skill #> <comparator> <threshold>

goto vector <vector #> if <condition> for vdn <vdn> <comparator> <threshold>

Condition = counted-calls:

goto vector <vector #> if <condition> to vdn <vdn> <comparator> <threshold>

Condition = expected-wait:

goto vector <vector #> if <condition> for split <split #> pri <priority level> <comparator> <threshold>

goto vector <vector #> if <condition> for skill <skill #> pri <priority level> <comparator> <threshold>

goto vector <vector #> if <condition> for call <comparator> <threshold>

Condition = interflow-qpos:

goto vector <step#> if interflow-qpos <comparator> <threshold>

Condition = wait-improved:

goto vector <vector #> if <condition> for split <split #> pri <priority level> <comparator> <threshold>

goto vector <vector #> if <condition> for skill <skill #> pri <priority level> <comparator> <threshold>

goto vector <vector #> if <condition> for best <comparator> <threshold>

Valid Entries

vector #: *1 through 256* (vs/csi/si), *1 through 512* (r)

split #: *1 through 99* (vs/csi/si), *1 through 600* (r)

skill #: *1 through 99* (vs/csi/si), *1 through 600* (r), *1st, 2nd, 3rd* (VDN skill)

condition:	comparator: [1]	threshold:
unconditionally	N/A	N/A
rolling-asa	>, >=, =, <>	*0-999* (1-second increments)
	<, <=	*0-999* (1-second increments)
available-agents	>, >=, =, <>	*0-199* (vs/csi/si), *0-998* (r)
	<, <=	*1-200* (vs/csi/si), *1-999* (r)
calls-queued	>, >=, =, <>	*0-199* (vs/csi/si), *0-998* (r)
	<, <=	*1-200* (vs/csi/si), *1-999* (r)
counted-calls	>, >=, =, <>	*0-999*
	<, <=	*1-999*
expected-wait	>, >=, =, <>	*0-9999* seconds
	<, <=	*1-9999* seconds
interflow-qpos	>, >=, =, <>, <, <=	1–9 (position in eligible queue)
oldest call-wait	>, >=, =, <>	*0-998* seconds (1-second increments)
	<, <=	*1-999* seconds (1-second increments)
staffed-agents	>, >=, =, <>	*0-199* (vs/csi/si), *0-998* (r)
	<, <=	*1-200* (vs/csi/si), *1-999* (r)
wait-improved	>, >=, =. <>	*0-9999 seconds*
	<, <=	*1-9999* seconds

1.Comparators =, >=, <=, <> (not equals) are only available for these conditions with Vectoring (G3V4 Enhanced).

digits: the following values are accepted:

command	comparator[1]	value[2]
goto step <step#> if digits	<, <=, >, >=	String of 0-9
	=, <>	String of 0-9, #, +, ?, none
	in table, not-in table[3]	1–10 (vs/csi/si), 1–100 (r)
goto step <step#> if ani	<, <=, >, >=	String of 0-9
	=, <>	String of 0-9, #, +, ?, none
goto step <step#> if ii-digits	<, <=, >, >=	Two consecutive digits from string of 0-9
	=, <>	Two consecutive digits from string of 0-9, +, ?, none

1.Comparators >, >=, <, <=, <> (not equals) are only available for the *goto step if digits* command with Vectoring (G3V4 Enhanced)

2.Wildcards (+, ?) can only be used with Vectoring (G3V4 Enhanced). The + represents a group of digits (0 or more) and can only be used as the first or last character of the string. Only one + can be used in any digit string. The ? represents a single digit. Any number of them can be used at any position in the digit string.

3.“Table” refers to a vector routing table. Vectoring (G3V4 Enhanced) must be enabled for the comparators “in” and “not-in” to be available.

priority level: *l* (low), *m* (medium), *h* (high), *t* (top)

day: *mon, tue, wed, thu, fri, sat, sun, all*

hour: *00* to *23* (military format)

minute: *00* to *59* (military format)

vdn: assigned vdn extension, active, latest. Active is the active called VDN as modified by VDN override rules. Latest is the VDN assigned to the vector in which the call is currently being processed.

Requirements

- Basic Call Vectoring software must be installed for all the options.
- Call Prompting software is required for the digits option.
- Skills replace splits only if Expert Agent Selection (EAS) is optioned.

- Wildcard matching and the use of Vector Routing Tables require Vectoring (G3V4 Enhanced).
- To use the *rolling-asa*, *expected-wait*, and *counted-calls* conditions, the "Vectoring (G3V4 Advanced Routing)" customer option must be enabled.
- To use *ani* and *II-digits* routing, the "Vectoring (ANI/II-Digits Routing)" customer option must be enabled.
- 1-second increments for *oldest-call-wait* and *rolling ASA* require DEFINITY ECS R6.3 or later and CentreVu® CMS R3V6 or later.
- To use either the *wait-improved* conditional or the *best* keyword with *expected-wait*, Vectoring (Best Service Routing) must be enabled.
- To use the *interflow-qpos* conditional, Look-Ahead Interflow (LAI) must be enabled.

Examples

goto vector *107* **if unconditionally**

goto vector *8* if **available-agents** in split *67 < 5*

goto vector *21* if **digits** >= *14*

goto vector *8* if **available-agents** in skill *1st > 10*

goto vector *32* if **digits** *in* table *12*

goto vector *9* if **expected-wait** in skill *3* pri *t > 10*

goto vector *99* if **calls-queued** in skill *2nd* pri *t > 5*

goto vector *8* if **rolling-asa** in split *14 > 25*

goto vector *9* if **wait-improved** for split *10* pri m > *30*

goto vector *2* if **wait-improved** for skill *10* pri m <= *30*

goto vector *12* if **wait-improved** for best >= *30*

goto vector 15 if **interflow-qpos** >=5

Operation

If the command syntax includes **unconditionally**, the command always branches. The unconditional form of the command is useful for applications that require the processing of more than 32 commands. Otherwise, branching takes place according to one of the conditions that follow:

- The rolling average speed of answer for the indicated split (skill) or VDN meets the constraints defined by the comparator and threshold value.
- Number of available agents in the indicated split (skill) meets the constraints defined by the comparator and the threshold value.
- Number of queued calls in the indicated split (skill) and at the specified priority level (or higher) meets the constraints defined by the comparator and the threshold value.
- The number of active calls in the indicated VDN meets the constraints defined by the comparator and the threshold value.
- The expected wait time at the specified priority level for the indicated split (skill), or the expected wait time for the call meets the constraints defined by the comparator and the threshold value.
- Oldest call-waiting in the indicated split (skill) at the specified priority level has been waiting for a period of time within the boundaries defined by the comparator and the threshold value, which is expressed in seconds.
- Number of staffed agents in the indicated split (skill) meets the constraints defined by the comparator and the threshold value.
- Digits collected via the *collect digits* command match the criteria defined by the comparator for the specified digit string. Or, the digits are found or not found, depending upon the option chosen, in the specified Vector Routing Table.
- The ani digits match the criteria defined by the comparator for the specified digit string. Or, the ani digits are found or not found, depending upon the option chosen, in the specified Vector Routing Table.
- The II-digits match the criteria defined by the comparator for the specified digit string. Or, the ani digits are found or not found, depending upon the option chosen, in the specified Vector Routing Table.
- Time-of-day criteria are met.

  NOTE:
  The syntax for this condition can be illustrated by a couple of examples, as follows: *mon 8:01 to fri 17:00* means “anytime between 8:01 A.M. Monday through 5:00 P.M. Friday,” and *all 17:00 to all 8:00* means “between 5:00 P.M. and 8:00 A.M. on any day of the week.”

- The Expected Wait Time (EWT) for the call is decreased by a period of time within the constraints defined by the comparator and the threshold value, which is expressed in seconds. The improvement in EWT is defined by calculating the difference between the call’s current EWT and its EWT were it to be queued to the resource specified in the command.
- The call’s position in the eligible portion of the queue meets the condition defined by the comparator and the threshold value (representing queue position counting backward from “1,” which is the head of the eligible queue).

Answer Supervision Considerations

Call answer is not affected by the command.

Feature Interactions

For Best Service Routing and Look-Ahead Interflow, the command is considered a neutral vector command in all cases. When a call experiences Look Ahead interflow, the ANI value is sent along with the call only for ISDN PRI calls. ANI is not sent for internal or DCS calls.

CMS/BCMS Interactions

R3 CMS:

Goto Vector Command

Database Item	Report Heading	Notes
OUTFLOWCALLS/ OUTFLOWTIME GOTOCALLS/ GOTOTIME	Vector Flow Out	
INTIME	Avg Time In Vector	
INFLOWCALLS	Vector Flow In	new vector

R2 CMS: Calls that go to another vector are tracked as OUTFLOW and INFLOW (for the new vector) in the vector tables. These calls are not shown in the standard reports. VECTIME is recorded for the vector and reported as “Avg Time in Vector.”

The ANI and/or II-digits is passed to CMS when the call first starts vector processing if the following is true:

- Basic Call Vectoring and/or Call Prompting is optioned
- ANI is available from the network, the call is internal, or is received over DCS
- The CMS is R3 (R3V5 for II-digits) or a later version
- II-digits is available from the network

ANI and II-digits are not passed to BCMS.

Messaging Command

```
messaging split (skill)
```

Purpose

Allows the caller to leave a message for the specified extension or the active or latest VDN extension (default).

Syntax

messaging split <split #> for extension <extension>

messaging skill <skill #> for extension <extension>

Valid Entries

split #: *1 through 99* (vs/csi/si), *1 through 600* (r)

skill #: *1 through 99* (vs/csi/si), *1 through 600* (r), *1st, 2nd, 3rd* (VDN skill)

extension: extension number, “active,” “latest.” Active is the active called VDN as modified by VDN override rules. Latest is the VDN assigned to the vector in which the call is currently being processed. Active is the default for this field. (Prior to G3V4, “none” was used to specify “active” VDN.)

Requirements

Basic Call Vectoring software must be installed. Also, the split (skill) involved must be an AUDIX split (skill), a remote AUDIX split or skill (DCS-AUDIX), or a Message Server Adjunct (MSA) split (skill). Also, skills replace splits only if Expert Agent Selection (EAS) is optioned.

Examples

messaging split *18* for extension *2000*

messaging skill *45* for extension *active*

Operation

This command causes the caller to be connected to the AUDIX or Message Center split (skill) so that the caller may leave a message for the specified extension (call answering service or “mail”).

If the split (skill) number specified in the command is a valid message service split or skill (such as an AUDIX or a Message Server Adjunct), and if the extension is either a valid assigned extension or is administered as active or latest the system attempts to terminate the call to the message service split (skill) for call answering service.

If the call is queued to the message service split (skill), or if the call terminates to an available message service agent or AUDIX voice port, the caller is connected to ringback (signifying successful termination), and vector processing terminates. Termination is unsuccessful, and vector processing continues at the next vector step if any one of the following is true:

- Split (Skill) queue is full.
- AUDIX link is down.
- All AUDIX voice ports are out of service.
- Message service split (skill) is DCS-AUDIX and all DCS trunks are busy.

If call termination is successful, and if the administered extension (or default VDN) is a message service subscriber, the caller can leave a message for the specified extension.

NOTE:
Agent and/or supervisor stations may be equipped with Automatic Message Wait (AMW) lamps to accommodate the “mail” specified in the *messaging split (skill)* command. The lamps can be assigned for VDNs or extensions used to access the messaging split (skill) and for which messages are to be left. When messages are left for these VDNs or extensions, the assigned AMW lamps light.

If the extension or VDN is not a subscriber of the message service, one of the following may occur:

- If the message service split (skill) is AUDIX, the caller receives ringback until he or she disconnects.
- If the message service split is a MSA, the caller may be answered by a message service agent, but no message is taken since the specified extension (default VDN) is not a MSA subscriber.

Answer Supervision Considerations

If answer supervision has not already been returned, it is returned when the messaging service port or station is connected to the call (that is, when the call is answered by the port or station).

Feature Interactions

The command can use an AUDIX or MSA hunt group in its operation.

If the command specifies a specific “mailbox” extension, the original principal for a call covered by a VDN is not passed to the adjunct, and it does not appear in the display to the answering agent. The specified extension appears in the display.

If the command is accessed via a direct call to the VDN, and if the mailbox is administered as “active” or “latest,” the corresponding active or latest VDN extension mailbox is sent to the messaging adjunct. Additionally, if the call is sent to a DEFINITY Message Service split (skill), the associated VDN name is sent to the messaging adjunct.

If the command specifies “active” or “latest” as the mailbox extension, the original principal for a call covered to or forwarded to a VDN is used as the default mailbox for the call instead of the “active” or “latest” VDN. Accordingly, the original principal extension and the reason for redirection are passed to the messaging adjunct, and they subsequently appear in the display to the answering agent.

AUDIX does not support mixed length numbering plans.

If the command leaves a message for a VDN or for another messaging service extension, the Automatic Message Waiting Lamp (AMWL) associated with the VDN or extension lights steady.

For Look-Ahead Interflow, the command can be considered as either a call acceptance vector command or a neutral vector command.

The command is considered a call acceptance vector command whenever one of the following is true:

- Call terminates to an agent or to an AUDIX port.
- Call queues to a messaging split (skill).

The command is considered a neutral vector command whenever the command fails.

CMS Interactions

R3 CMS: When a queued call successfully goes to the messaging split, OUTFLOWCALLS/OUTFLOWTIME (1st split [skill]) and DEQUECALLS/DEQUETIME (2nd/3rd splits [skills]) are tracked in the split (skill) tables. These calls are reported as split (skill) “Flow Out,” “Dequeued Calls,” and “Dequeued Avg Queue Time.”

Calls that queue via a *messaging split (skill)* command are tracked as CALLSOFFERRED and LOWCALLS (no priority) or MEDCALLS (priority). These calls are shown in the standard reports according to the final disposition of the call.

The presence of the command in a vector enables the calls serviced by the vector to be vector-directed. When such a call is answered by an agent, the call is tracked as ACDCALLS/ANSTIME, and it is reported as “ACD Calls,” “Split ACD Calls,” and “Avg Speed Ans.”

Finally, if the command directs a call to a split (skill), the BACKUPCALLS database item is incremented, and the call is reported as “Calls Ans in Backup” and “Calls Handled/Backup.” The “Calls Ans in Main” report item is calculated by using the algorithm ACDCALLS - BACKUPCALLS.

A call abandoned after the command routes the call to a station or to an attendant is tracked as ABNCALLS/ABNTIME for the messaging split (skill) and in the VDN/vector tables.

R2 CMS: Calls that go to a messaging split (skill) are tracked as OUTFLOW in the split (skill), vector, and VDN tables. The messaging split (skill) also receives an INFLOW. The calls are shown as “Flow Out” and “Flow In” in standard split (skill) reports. VECTIME is recorded for the vector and reported as “Avg Time in Vector.”

BCMS Interactions

A call advanced to another position via the command is tracked as an “outflow” in the VDN Report.

Queue-to Command

`queue-to`

Purpose

Unconditionally queues a call to a split or skill and assigns a queuing priority level to the call in case all agents are busy.

Syntax

queue-to split <split #> pri <priority level>

queue-to skill <skill #> pri <priority level>

queue-to best

Valid Entries

split #: *1 through 99* (vs/csi/si), *1 through 600* (r)

skill #: *1 through 99* (specific: vs/csi/si), *1 through 600* (r), *1st, 2nd, 3rd* (VDN skill)

priority level: *l* (low), *m* (medium), *h* (high), *t* (top)

Requirements

Basic Call Vectoring software must be installed. The split (skill) involved must be vector-controlled. Also, skills replace splits only if Expert Agent Selection (EAS) is optioned. Use of the *best* keyword requires single- or multi-site Best Service Routing™.

Examples

queue-to split *53* pri *t*

queue-to skill 1st pri *t*

queue-to best

Operation

A call sent with this command either connects to an available agent in the resource specified or enter its queue. If single-site BSR is enabled, *queue-to best* queues or delivers a call to the best local split or skill found by a consider series.

If multi-site BSR is enabled, the best resource may be at a remote location; in this case, *queue-to best* interflows the call to the interflow VDN defined for that location on the BSR Application form.

A call may be queued to up to three local splits (skills) simultaneously. A call remains queued either until vector processing terminates (via a *disconnect, busy*, or *route-to* command, or via a dropped or abandoned call) or until the call reaches an agent. When an agent becomes available in any split (skill) to which the call is queued, the following actions take place:

- Call begins alerting the agent.
- Call is removed from any other queues.
- Vector processing terminates.

If the entered split (skill) is one of the splits (skills) to which the call is already queued, the call is requeued at the new priority level. If the priority level specified is the same as the priority level at which the call is queued, the call remains in the same position in queue. The step is skipped, and vector processing continues at the next step if any of the following conditions are true:

- Desired split's (skill's) queue is full.
- Desired split (skill) is not vector-controlled.
- Desired split (skill) has no queue and also no available agents.
- Call has been previously queued to three different splits (skills).

NOTE:
A *route-to* to another VDN can be used to remove the call from the splits it is queued to if necessary. The steps in the routed-to vector then can be used to queue to other splits.

A *queue-to best* command will have the same operation and interactions as the *queue-to split/skill* command when the best resource is a local split or skill. When the best resource is at a remote location, the *queue-to best* command will function as an unconditional *route-to* command (with cov=n) performing Look-Ahead Interflow.

When a *queue-to best* command executes, it initializes the data for the best resource (the "best" data) the consider series found for this call. If no "best" data has been defined by the consider series, a vector event is logged and processing continues at the next vector step. A consider series might not produce "best" data for any of the following reasons:

- all resources considered are unstaffed
- no resource considered has an open queue slot
- "best" data has been initialized before execution of the *reply-best* step (because there are no consider steps in the status poll vector or because the vector contains a prior step that initializes "best" data).

For a list of events and vector commands that initialize “best” data produced by consider series, see “Events that Clear “Best” Data” on page A-43.

If a queue attempt to a local resource fails, a vector event is logged and processing continues at the next vector step. The “best” data is initialized.

If an interflow attempt to a remote resource fails, a vector event is logged and processing continues at the next vector step. If a local split or skill was identified as best at some point in the consider series before the interflow attempt, the call is queued to the local resource. Whether or not the call can be queued locally in this case, the “best” data is initialized and processing continues at the next vector step.

Answer Supervision Considerations

Answer supervision is returned (if not already returned) when the call is connected to an answering agent.

Feature Interactions

The *queue-to* command can access a AUDIX/Message Center/Server split (skill) in cases where a VDN is assigned as a coverage point. To enable this function, the split (skill) must be assigned as a vector-controlled hunt group.

For Best Service Routing and Look-Ahead Interflow, the command can be considered either a call acceptance vector command or a neutral vector command.

The command is considered a call acceptance vector command whenever one of the following is true:

- Call terminates to an agent.
- Call queues to a split (skill).
- BSR interflowed call is accepted at remote interflow vector.

The command is considered a neutral vector command when the call neither terminates nor queues.

No COR checking is carried out when a queue-to step places a call to a split (skill).

CMS Interactions

R3 CMS: Calls queued via a *queue-to split (skill)* command are tracked as CALLSOFFERRED and LOWCALLS/MEDCALLS/HIGHCALLS/TOPCALLS.

Split (skill) calls are reported in the standard reports according to the final disposition of the call.

The presence of the command in a vector enables the calls that are serviced by the vector to be vector-directed. When such a call is answered by an agent, the call is tracked as ACDCALLS/ANSTIME, and it is reported as “ACD Calls,” “Split (skill) ACD Calls,” and “Avg Speed Ans.” If the call is also queued to other splits (skills), OUTFLOWCALLS/OUTFLOWTIME is tracked in the first split (skill) to which the call queues, and “Flow Out” is reported (unless the split (skill) turns out to be the answering split (skill)). DEQUECALLS/DEQUETIME is tracked in the second and third splits (skills) if these splits (skills) are not the answering split (skill), and the call is reported as “Dequeued Calls” and “Dequeued Avg Queue Time.” However, if the second or third split (skill) is the answering split (skill), INFLOWCALLS is tracked in the split (skill), and the call is reported as “Flow In.”

If the call abandons after the command queues the call to a split (skill), ABNCALLS/ABNTIME is tracked for the vector, the VDN, and the first split (skill) to which the call is queued. The call is reported as “Aban Call” and “Avg Aban Time.” If the call is also queued to other splits (skills), DEQUECALLS/DEQUETIME is tracked in these splits (skills), and the call is reported as “Dequeued Calls” and “Dequeued Avg Queue Time.”

BSR status poll calls are not counted as interflows. BSR interflows are now tracked as network interflowed calls (NETCALLS) by CMS at the receiving switch. CMS tracks a call’s accumulated time-in-VDN as NETINTIME (that is, the NET_TIME value on CMS at switch C combines the time a call has spent in VDNs at any previous locations, as communicated by ISDN information forwarding. The NETINTIME can be added to the time spent in the local switch to provide reports that include the total time the call has spent in the call center network (e.g., total ASA).

For more information on CMS database units and reports, see *CentreVu® Call Management System R3V6 Administration* (585-215-850) and *CentreVu® Call Management System R3V6 Reports* (585-215-851).

R2 CMS: *queue-to split (skill)* calls are shown in the standard reports according to the final disposition of the call.

An ACD call directed via the command and subsequently answered is tracked as ANSWERED/ANSDELAY and ACDCALLS. The call is reported as “No. ACD Calls,” “No. Calls Ans,” and “Avg Speed Ans.” If the call is also queued to other splits (skills), OUTFLOW is tracked in these splits (skills), and the call is reported as “Flow Out.” Calls directed via the command are tracked as ANSMAIN, and they are reported as “No. Calls Ans in Main.”

If the call abandons after the command queues the call to a split (skill), ABNCALLS/ABNTIME is tracked for the vector, the VDN, and the first split (skill) to which the call is currently queued. The call is reported as “No. Aban Calls” and “Avg Aban Time.” If the call is also queued to other splits (skills),

OUTFLOWCALLS/OUTFLOWTIME is tracked in these splits (skills), and the call is reported as “Flow Out.”

BCMS Interactions

The total number of calls to the VDN that are queued via the command and then answered by an agent within a specified time period is tracked as “ACD CALLS” in the VDN Report. The average time that calls spend in a vector before being connected via the command as an ACD call to an agent is tracked as “AVG SPEED ANS” in the same report.

There is no added tracking for calls interflowed by Best Service Routing ™. BCMS tracks these calls as “outflow” in the VDN Report.

Reply-best

```
reply-best
```

Purpose

The *reply-best* command is used only in status poll vectors in multi-site Best Service Routing applications, where it returns “best” data for its location to the primary vector on the origin switch.

Syntax

reply-best

Requirements

Basic Call Vectoring software must be installed. In addition, Vectoring (G3V4 Advanced Routing), Look-Ahead Interflow (LAI), and Vectoring (Best Service Routing) are required.

Example

reply-best

Operation

The sole purpose of the *reply-best* step is to return data for the best resource found by the consider series in a status poll vector to the primary vector in a multi-site BSR application. The status poll vector executes in response to an ISDN call from a *consider* step in the primary vector. Each time the status poll vector executes, the *reply-best* step:

- drops the incoming ISDN call without returning answer supervision
- returns status data to the primary vector via the ISDN DISCONNECT message
- initializes (clears) the “best” data
- terminates processing in the status poll vector

If the incoming call is not an ISDN call, the *reply-best* command will drop the call and log a vector event. No status data will be returned to the origin switch.

If the consider series yields no “best” data, the *reply-best* command will drop the incoming ISDN call without returning answer supervision, terminate vector processing, and return an infinite value for EWT in the DISCONNECT message. A consider series might not produce “best” data for any of the following reasons:

- all resources considered are unstaffed
- no resource considered has an open queue slot
- "best" data has been initialized before execution of the *reply-best* step (because there are no consider steps in the status poll vector or because the vector contains a prior step that initializes "best" data.

For a list of events and vector commands that initialize "best" data produced by consider series, see "Events that Clear "Best" Data" on page A-43.

Answer Supervision Considerations

The *reply-best* step does not return answer supervision.

CMS/BCMS Interactions

Operation of the *reply-best* command is not reported or tracked by BCMS or CMS.

Route-to Command

`route-to`

Purpose

Routes calls either to a destination that is specified by digits collected from the caller or an adjunct (*route-to digits*), or routes calls to the destination specified by the administered digit string (*route-to number*).

Syntax

route-to digits with coverage <option>

route-to number <number> with cov <option> if **unconditionally**

route-to number <number> with cov <option> if **digit** <comparator> <digit>

route-to number <number> with cov <option> if **interflow-qpos** <q-pos comparator> <q-pos>

Valid Entries

number: 1 to 16 digits (includes the Abbreviated Dialing (AD) special characters (*~p, ~w, ~m, ~s, ~W), *, #.*)

option: *n* (no), *y* (yes)

comparator: *=, <, <=, >, >=, <>*

q-pos comparator: =, <, <=

q-pos: 1–9 (represents position in interflow-eligible queue, with 1 being the head of the queue)

digit: *0 through 9 or a single "#"*

Requirements

Route-to digits requires Call Prompting software. To collect Caller Information Forwarding (CINFO) digits for a *route-to digits* step, Vectoring (CINFO) must also be enabled.

Route-to number requires Basic Call Vectoring software. For the digit option, Call Prompting software must be installed. Vectoring (CINFO) is required to collect CINFO digits for the digit option. Comparators other than = (equals) are only available with Vectoring (G3V4 Enhanced).

The *interflow-qpos* conditional requires Look-Ahead Interflow (LAI).

Examples

route-to digits with coverage *y*

route-to number *3300* with cov *n* if **unconditionally**

route-to number *473957* with cov *y* if **digit** >= *8*

route-to number *433977* with cov *y* if **interflow-qpos** = *1*

Operation

The *route-to* command attempts to route a call to a set of digits collected from the caller, the network, or from an adjunct, or to route to the destination specified by the administered digit string.

For the route-to number ... if digit command, the call is conditionally routed to a specified destination according to a single digit entered by the caller. If the digit collected in the last *collect digits* command matches the specified comparison in relation to the administered digit, the command attempts to route the call to the specified destination.

The destination for a route-to command can be any of the following:

- Internal extension (for example, split/hunt group, station, etc.)
- VDN extension
- Attendant or Attendant Queue
- Remote extension (UDP/DCS)
- External number, such as a TAC or AAR/ARS FAC followed by a public or private network number (for example, 7-digit ETN, 10-digit DDD, etc.)
- Remote Access Extension
- Service Observing FAC

NOTE:
The VDN's Class of Restriction (COR) is used for calling permissions.

The *route-to digits* command fails if no digits are collected, and vector processing continues at the next vector step.

The *route-to number ... if digit* command fails if more than 1 digit is collected or if the digit comparison fails. Vector processing continues at the next command.

The *route-to number ... if interflow-qpos* command fails if the call is not in the eligible queue established by the *interflow-qpos* condition. Vector processing continues at the next command.

If the *route-to* command is successful, vector processing terminates. Otherwise, vector processing continues at the next vector command.

A *route-to* step in a vector is treated as cov=n for a covered call regardless of the cov setting on the *route-to* command.

If the number expressed in the command is a system extension or an attendant group (and not a VDN), the system considers the step successful if one of the following conditions occurs:

- The endpoint is alerted.
- The endpoint has Call Forwarding or night service (hunt group) enabled, and the (night service) destination forwarded to is alerted; or, if off-premises Call Forwarding (UDP hunt night service), a trunk is seized.

The system then provides ringback to the caller, and vector processing terminates. However, if the call cannot complete successfully (for example, no idle appearance is available), vector processing continues at the next vector command.

If the number is a VDN extension, the following events occur:

- Vector processing terminates within the current vector.
- If the current VDN is administered with override, the new VDN overrides current VDN information.
- Processing of the vector associated with the VDN extension begins.

If the number is an AAR/ARS FAC plus digits, or if it is a remote UDP extension, standard AAR/ARS processing is performed to select the trunk group and outpulse the digits. If a trunk is seized, vector processing terminates, and the calling party hears feedback provided by the far end. Otherwise, the call cannot complete successfully (because no trunks are available, the FRL/COR is restricted, etc.), and vector processing continues at the next vector command.

If the number is a TAC plus digits, and a trunk is seized, vector processing terminates, and the calling party hears feedback provided by the far end. Otherwise, the call cannot complete successfully (because no trunks are available, the COR is restricted, etc.), and vector processing continues at the next vector command.

If the number is any other number (such as an FAC other than an AAR/ARS or Service Observing), the command is unsuccessful, and vector processing continues at the next vector command.

Abbreviated Dialing special characters can also be used in the number field. Each of these characters instructs the system to take a different action when dialing reaches the point where the character is stored. The characters are as follows:

- ~*p* (pause)
- ~*w* (wait)
- ~*m* (mark)
- ~*s* (suppress)
- ~*W* (indefinite wait)

Each special character counts as two digits towards the maximum. The maximum number of digits for the command is 16.

The *route-to digits* command can be used to implement an automated attendant function.

Coverage

The optional coverage parameter determines whether coverage should apply during routing. If coverage applies, and if the digits entered are valid, the following occurs:

- Ringback is provided.
- Vector processing terminates.
- Normal termination and coverage are implemented.

NOTE:
For detailed information about the operation of the route-to command with or without coverage for the different destinations see Table H-1.

Answer Supervision Considerations

Generally, answer supervision is provided when the destination answers the call. The exception to this involves incoming trunk calls routed to another non-ISDN-PRI trunk. Such calls provide answer supervision when the outgoing trunk is seized.

Feature Interactions

When COR checking is applied to a route-to number or route-to digits step, it is the COR of the latest VDN that is used.

The *route-to* command may specify the AAR or ARS access codes. The COR associated with the latest VDN is used to determine the Partitioned Group

Number (PGN) time-of-day routing chart. The PGN determines the choice or route tables used on a particular call.

The command may call the AUDIX extension. If this happens, the call is treated as a direct call to AUDIX, and the calling party may retrieve his or her messages.

If the call covers to a VDN, the command supports a remote AUDIX interface to a local hunt group extension that is assigned as a remote AUDIX hunt group. The "remote AUDIX hunt group" (which has no members and cannot be vector-controlled) forwards the call to the remote AUDIX destination in the same manner as when the hunt group is assigned as a point in the coverage path. A DCS link down condition for a call that covers to a VDN is treated as a direct call to AUDIX.

If the command is directed to a station with bridged appearances, the bridged appearance button lamps are updated.

The following destinations always result in a failure, and vector processing continues at the next step:

- Controlled trunk group
- Code calling FAC
- Facility test call
- TAAS access code
- Priority access code
- Loudspeaker paging access code
- Station Message Detail Recording (SMDR) account code
- Voice message retrieval access code

If the command is executed and Direct Outward Dialing (DOD) is in effect, the COR of the latest VDN is compared with the COR of the called facility to determine if the call is permitted. If access is not permitted, the command fails and vector processing continues. In the case where a COR requiring the entry of account codes is assigned to a VDN, and the command is executed by the associated vector, the command is unsuccessful, and vector processing continues at the next step.

The individual extension number assigned to an attendant console can be used as the command's argument.

A call processed by the command can wait in the individual attendant queue and is subsequently removed from vector processing.

The command can access both public and private networks.

If the command dials the attendant, and if the system is in night service, the call routes to the DID Listed Directory Number (LDN) night destination.

The command can place AAR/ARS calls that implement subnet trunking, which is the routing of calls over trunk groups that terminate in switches with different dial plans.

Authorization codes are disabled with respect to routing via VDNs. In other words, if authorization codes are enabled, and a *route-to* command in a prompting vector accesses AAR or ARS, and the VDN's FRL does not have the permission to utilize the chosen routing preference, no authorization code is prompted for, and the *route-to* command fails.

If the command routes the call without coverage to a display station, the station displays the following: "a = Originator Name to VDN Name."

If the command calls a station that is a member of a pickup group, the call can be picked up by another pickup group member.

Anytime a *route-to with cov n* command initiates a call over ISDN-PRI facilities and Look-Ahead Interflow is optioned, the call will be treated on a Look-Ahead basis. However, if the command is used with the *coverage yes* option in effect, unconditional interflow results.

For Look-Ahead Interflow, the *route-to* command can be considered either a call acceptance vector command or a neutral vector command. The command is considered a call acceptance vector command whenever one of the following is true:

- Command terminates to a valid local destination.
- Command successfully seizes a non-PRI trunk.
- Command execution results in a Look-Ahead Interflow call attempt, and the call is accepted by the far end switch.

The command is considered a neutral vector command whenever one of the following is true:

- Termination is unsuccessful.
- Trunk is not seized.
- Look-Ahead interflow call attempt is denied by the far end switch.

For a call that covers or forwards to a VDN, the *route-to with coverage y* command functions the same way as the *route-to with coverage n* command. For a covered or forwarded call, the coverage option for the command is disabled since such a call should not be further redirected.

A *route-to with cov y* to a station that has call forwarding activated is forwarded.

Service Observing can be initiated with Call Vectoring using the route-to command. See "Service Observing" on page 3-18 for detailed instructions.

NOTE:
Appendix H gives a detailed description of the feature interactions for the *route-to* number with and without coverage command.

CMS Interactions

R3 CMS: Tracking of the *route-to digits* command varies according to the destination successfully routed to, as follows:

Routed to Station or to Attendant

Database Item	Report Heading	Notes
OUTFLOWCALLS/ OUTFLOWTIME	Flow Out Vector Flow Out	1st split
DEQUECALLS/ DEQUETIME	Dequeued Calls Dequeued Avg Queue Time	2nd/3rd splits
INTIME	Avg Time In Vector	
CONNECTCALLS/ CONNECTTIME	Other Calls Connect	answered calls on G3

Routed to Trunk

Database Item	Report Heading	Notes
OUTFLOWCALLS/ OUTFLOWTIME	Flow Out Vector Flow Out VDN Flow Out	1st split
DEQUECALLS/ DEQUETIME	Dequeued Calls Dequeued Avg Queue Time	2nd/3rd splits
INTERFLOWCALLS / INTERFLOWTIME	VDN Flow-Interflow	
INTIME	Avg Time In Vector	

Routed to VDN

Database Item	Report Heading	Notes
OUTFLOWCALLS/ OUTFLOWTIME	Flow Out Vector Flow Out VDN Flow Out	1st split
DEQUECALLS/DE QUETIME	Dequeued Calls Dequeued Avg Queue Time	2nd/3rd splits

Routed to VDN

INTIME	Avg Time In Vector	
INFLOWCALLS	Vector Flow In VDN Flow In	new vector new VDN

Routed to Split or Hunt Group

Database Item	Report Heading	Notes
OUTFLOWCALLS/ OUTFLOWTIME	Flow Out	1st split
DEQUECALLS/ DEQUETIME	Dequeued Calls Dequeued Avg Queue Time	2nd/3rd splits
INTIME	Avg Time In Vector	
CALLSOFFERRED		new split
MEDCALLS/ HIGHCALLS		no priority/priority

NOTE:
For calls that "route to" a split or a hunt group and later intraflow to a station or to an attendant, OTHERCALLS/OTHERTIME are tracked in the vector and in the VDN tables.

Split calls are also shown in the standard reports according to the final disposition of the call.

Calls that route over an ISDN trunk are Look-Ahead Interflow calls. When a call attempts to "route to" an ISDN trunk (Look-Ahead Interflow), the LOOKATTEMPTS database item is tracked and reported as "Look-Ahead Interflow Attempts." If the call successfully routes, LOOKFLOWCALLS/LOOKFLOWTIME are tracked and reported as "Look-Ahead Interflow Completions." Interflow always occurs whenever the *with coverage yes* option is in effect.

The presence of the command in a vector enables the calls that are serviced by the vector to be vector-directed. When such a call is answered by an agent, the call is tracked as ACDCALLS/ANSTIME, and it is reported as "ACD Calls," "Split ACD Calls," and "Avg Speed Ans." If the call is also queued to other splits, OUTFLOWCALLS/OUTFLOWTIME is tracked in the first split to which the call queues, and "Flow Out" is reported (unless the split turns out to be the answering split). DEQUECALLS/DEQUETIME is tracked in the second and third splits if these splits are not the answering split, and the call is reported as "Dequeued Calls" and "Dequeued Avg Queue Time." However, if the second or third split is the answering split, INFLOWCALLS is tracked in the split, and the call is reported as "Flow In."

If the command directs a call to a destination, the BACKUPCALLS data base item is incremented, and the call is reported as “Calls Ans in Backup” and “Calls Handled/Backup.” The “Calls Ans in Main” report item is calculated by using the algorithm ACDCALLS - BACKUPCALLS.

A call abandoned after the command routes the call to a station or an attendant is tracked in the VDN tables as ABNCALLS/ABNTIME.

BSR interflows are now tracked as network interflowed calls (NETCALLS) by CMS at the receiving switch. CMS tracks a call’s accumulated time-in-VDN as NETINTIME (that is, the NET_TIME value on CMS at switch C combines the time a call has spent in VDNs at any previous locations, as communicated by ISDN information forwarding. The NETINTIME can be added to the time spent in the local switch to provide reports that include the total time the call has spent in the call center network (e.g., total ASA).

For more information on CMS database units and reports, see *CentreVu® Call Management System R3V6 Administration* (585-215-850) and *CentreVu® Call Management System R3V6 Reports* (585-215-851).

R2 CMS: Calls that “route to” successfully are tracked as OUTFLOW and reported as “Flow Out.” The vector and VDN tables are not credited with an answer or abandon for calls that route to a split. VECTIME is recorded for the vector, and it is reported as “Avg Time in Vector.”

Calls that “route to” a split are tracked as INFLOW and reported as “Flow In”. Calls that “route to” a trunk are tracked as ROUTEDCALLS/ROUTEDTIME, but they are not shown on standard reports.

BCMS Interactions

A call advanced to another position via the command is tracked as “outflow” in the VDN Report. A call answered by an attendant via the command is also tracked as “outflow.”

There is no added tracking for calls interflowed by Best Service Routing ™. BCMS tracks these calls as “outflow” in the VDN Report.

Stop Command

```
stop
```

Purpose

Halts the processing of any subsequent vector steps.

Syntax

stop

Requirements

Basic Call Vectoring or Call Prompting software must be installed.

Operation

After the *stop* command is processed, any calls already queued remain queued, and any wait treatment (for example, silence, ringback, music) is continued. On the other hand, any calls not queued are dropped under the same scenario.

If a TTR is allocated to the call, and if the *stop* command is encountered, the TTR is disconnected. However, current call processing continues (that is, the call is not dropped). The caller continues to hear the feedback that was provided before the *stop* command was encountered.

NOTE:
An implicit stop is processed following the last administered command in a vector.

Answer Supervision Considerations

The command has no effect on answer supervision.

Feature Interactions

For Look-Ahead Interflow, the command is considered a neutral vector command in all cases except when a call is dropped, then it is considered a denial.

CMS Interactions

R3 CMS: When the command or the end of the vector is encountered, vector INTIME is recorded. This is reported as “Avg Time in Vector.”

R3V4 CMS: VDISCCALLS database item in the VDN tables pegs call that pass all the way through a vector without ever having been queued.

R2 CMS: Nothing is recorded when the command is encountered. When the disposition (that is, “answered” or “abandoned”) of the call is known, VECTIME is recorded and reported as “Total Time in Vector.”

Wait-time Command

```
wait-time
```

Purpose

Delays the processing of the next vector step if a specified delay time is included in the command's syntax. Also provides feedback (in the form of silence, ringback, or music) to the caller while the call advances in queue. The G3V4 Multiple Audio/Music Sources for Vector Delay and Multiple Music Sources on Hold features allow a specified audio or music source to be selected when a call encounters a *wait-time* command. See "Multiple Audio or Music Sources on Delay" on page 3-5 and "Delays for message vectors" on page 3-7 for more information.

Syntax

wait-time <time> <unit> hearing <treatment>

wait-time <time> <unit> hearing <extension> then <treatment 2>

Valid Entries

time: *0 through 999* for seconds; *0 through 480* for minutes; *0 through 8* for hours; *0 through 9 secs* when using the i-silent treatment.

unit: *secs* (seconds), *mins* (minutes), *hrs* (hours)

treatment: silence, ringback, music, i-silent.

treatment 2: silence, ringback, music, continue.

With Multiple Audio/Music Sources, treatment refers to what the caller hears after the wait-time command times out, if the announcement finishes before time out, or if the call cannot be connected to the audio/music source extension. Continue is only a valid treatment with Multiple Audio/Music Sources. It indicates that the caller will continue to hear the alternate audio/music source until another vector command takes effect.

When music is indicated as a treatment, it refers to the system music, not an alternate music source.

The tenant number of the active VDN determines the system music the caller hears. You can allow callers to hear a music source other than the one assigned to the active VDN, however, by directly specifying an extension for an audio source with a command such as *wait-time 30 secs hearing 4301 then music*.

The “i-silent” keyword is for use with adjunct routing-ADR/Lookahead Interflow applications. I-silent provides silence for the specified time, but it is neutral to Look-Ahead Interflow while all other wait treatments (even with 0 secs settings) provide acceptance.

extension: The valid extension number of an alternate audio/music source.

Requirements

Basic Call Vectoring or Call Prompting software must be installed. Also, a music-on-hold port must be provided for the music treatment. Multiple Audio/Music Sources for Vector Delay requires that the “Vectoring (G3V4 Enhanced)” customer option be enabled.

Examples

wait-time *240* secs hearing *music*

wait-time *4* mins hearing *music*

wait-time *2* hrs hearing *silence*

wait-time *120* secs hearing *54795* then *continue*

wait-time *2* mins hearing *54795* then *continue*

wait-time *8* hrs hearing *silence* then *continue*

Operation

The specified feedback is given to the caller, and vector processing waits the specified time before going on to the next step. If the time specified is *0*, feedback is provided without any delay in the processing of the next vector step. The feedback given to the caller continues until any one of the following occurs:

- Subsequent vector step (containing *wait-time* or *announcement*) changes the treatment.
- Vector processing encounters a *disconnect* or *busy* command.
- Call is routed to another location or to a step that includes an announcement (for example, *collect digits*).
- Call is routed to another VDN.
- Call is delivered to a destination (starts ringing at an agent’s terminal).
- Switch receives a destination from the ASAI adjunct.
- Vector disconnect timer expires.

Wait times up to 8 hours are allowed for customers who want to use the ASAI Phantom Call feature to track e-mail and fax messages in split queues.

Answer Supervision Considerations

If the *music* or audio source treatment is included in the command, answer supervision is triggered. If the command is encountered and answer supervision was sent previously, the caller hears the treatment specified in the current command. If, for a CO trunk user, the command with *silence, ringback,* or *i-silent* treatment is encountered prior to answer supervision, the caller continues to hear ringback from the CO.

Feature Interactions

- Music-on-Hold

  When the command is implemented with music as the treatment, the system-wide music-on-hold feature must be administered. Otherwise, the caller hears silence. When Tenant Partitioning is in use, the tenant number of the active VDN determines the system music that is heard.

  Feedback continues while a subsequent vector step queues for an announcement or for a TTR.

- Look-Ahead Interflow

  For Look-Ahead Interflow, the wait-time command is considered a call acceptance vector command in all cases, except i-silent, which is considered a neutral vector command.

  NOTE:
  An implicit wait of 0.2 seconds (with no change in the feedback to the caller) is provided after every seven vector steps if one of these steps does not suspend vector processing. (The following steps, if successful, do not suspend vector processing: *queue-to split, check split, goto step, goto vector* and *wait-time 0 seconds*. The following steps, if unsuccessful, also do not suspend vector processing: *check split, route-to, adjunct routing*, and *messaging split*. The only commands that suspend vector processing are the following: *announcement, wait-time > 0, collect digits*, and *converse-on split*.

- Vector Disconnect Timer

  The timer, which is administered on the Feature-Related System Parameters form, cannot be set for more than 240 minutes. If you want to use *wait-time* commands for periods of more than 240 minutes, set the timer to 0. (Setting the timer to 0 turns it off; it does not limit the duration of wait-time steps.)

CMS/BCMS Interactions

The command is not tracked on CMS or BCMS.

Vectors with *wait-time* steps are only accessible to CMS if the time unit is administered in *secs*.

Criteria for Success/Failure of Call Vectoring Commands

Table A-15 summarizes the success and failure criteria for various vector commands. Before you write or evaluate vectors, it is important to understand the information in this table.

NOTE:
If EAS is enabled, “skill” replaces “split.”

Table A-15. Call Vectoring Command Success/Failure Criteria

Command	Success/Failure Criteria	Vector Processing Disposition
adjunct routing	Fails if any of the following are true: ■ VDN’s COR does not permit routing to the adjunct-supplied destination. ■ TAC/ARS/AAR code is invalid. ■ Specified agent is not logged into the specified split for a direct agent call. ■ Local extension is not in the dialplan. ■ Invalid number was dialed.	Stop *wait-time* or *announcement* step (if present). Then continue vector processing with the next sequential step.
	Otherwise, succeeds.	Route the call and provide feedback.
announcement	Fails if specified announcement is unadministered, not recorded, or busied out.	Continue vector processing with the next sequential step.
	Otherwise, succeeds.	Play the announcement, then continue at the next sequential step.
busy	Always succeeds. (CO without answer supervision trunk callers will not hear the busy tone.)	Exit vector processing, then play the busy tone for 45 seconds before dropping the call. (Unanswered CO trunk calls receive 45 seconds of ringback.)

Continued on next page

Table A-15. Call Vectoring Command Success/Failure Criteria — *Continued*

Command	Success/Failure Criteria	Vector Processing Disposition
check split	Fails if any of the following are true: ■ Vector conditional is false. ■ Split's queue is full. ■ Split is not vector-controlled. ■ Call is already queued at the specified priority to the specified split. ■ Call is already queued to three different splits.	Continue vector processing with the next sequential step.
	Otherwise:	
	Succeeds, and the call is terminated to an agent.	Exit vector processing, and pass control to call processing.
	Succeeds, and the call is queued or requeued in the specified split at the specified priority.	Continue vector processing with the next sequential step.
collect digits	Fails if any of the following are true:	
	■ Call originates from an outside caller who is not using a touch-tone telephone.	Call Prompting timer takes effect, command times out, and vector processing continues at the next vector step.
	■ No TTR is in the system, or the TTR queue is full.	Continue vector processing at the next step.
	■ Caller enters fewer digits than the maximum specified.	Call Prompting timer takes effect, command is terminated, and any digits collected prior to the timeout are available for subsequent processing.
	Otherwise, succeeds.	Continue vector processing at the next step.

Continued on next page

Table A-15. Call Vectoring Command Success/Failure Criteria — *Continued*

Command	Success/Failure Criteria	Vector Processing Disposition
consider location	Fails if any of the following are true: ■ No BSR application administered in active VDN. ■ Location not administered in BSR application. ■ Status Poll VDN number not administered in BSR application. ■ Status Poll VDN number is invalid. ■ Status Poll fails because all trunks are busy.	Continue vector processing with the next sequential step.
	Otherwise:	
	Succeeds, but takes no action if polling of specified location is suppressed.	Continue vector processing with the next sequential step.
	Succeeds, and place status poll call to the status poll VDN.	Suspend vector processing until status poll response received.
consider split	Fails if any of the following are true: ■ VDN skill (1st, 2nd, 3rd) is used in consider step but not administered for active VDN.	Continue vector processing with the next sequential step.
	Otherwise: Succeeds, and the status of the local split is evaluated.	

Continued on next page

Table A-15. Call Vectoring Command Success/Failure Criteria — *Continued*

Command	Success/Failure Criteria	Vector Processing Disposition
converse-on split	Fails if any of the following are true: ■ Converse split queue is full. ■ Converse split is not vector-controlled. ■ Auto-available split is in effect, and all agents are logged out by Redirection on No Answer (RONA).	Continue vector processing with the next sequential step.
	Otherwise: Succeeds, call is delivered to the converse split, and (if administered) digits are outpulsed to the VRU. The caller is connected to the VRU, the voice response script is executed, and (if necessary) digits are outpulsed to the switch.	Continue vector processing with the next sequential step.
disconnect	Always succeeds.	Play the announcement (if specified). Then drop the call.
goto step	Fails if the step condition is not met.	Continue vector processing with the next sequential step.
	Succeeds if the step condition is met.	Continue vector processing with the destination step.
goto vector	Fails if the step condition is not met.	Continue vector processing with the next sequential step.
	Succeeds if the step condition is met.	Continue vector processing with the first nonblank step of the destination vector.

Continued on next page

Table A-15. Call Vectoring Command Success/Failure Criteria — *Continued*

Command	Success/Failure Criteria	Vector Processing Disposition
messaging split	Fails if any of the following are true: ■ Specified split is not an AUDIX split or a Message Server Adjunct (MSA) split. ■ Specified extension is invalid. ■ Messaging split queue is full. ■ Messaging split is not vector controlled and has no working agents (none logged in or all in AUX work mode). ■ Communications link with the messaging adjunct is inaccessible.	Continue vector processing with the next sequential step.
	Otherwise, succeeds.	Terminate vector processing.
queue-to split	Fails if any of the following are true: ■ Split's queue is full. ■ Split is not vector-controlled. ■ Call is already queued at the specified priority to the specified split. ■ Call is already queued to three different splits.	Continue vector processing with the next sequential step.
	Otherwise:	
	Succeeds, and the call is terminated to an agent.	Exit vector processing, and pass control to call processing.
	Succeeds, and the call is queued or requeued in the specified split at the specified priority.	Continue vector processing with the next sequential step.

Continued on next page

Table A-15. Call Vectoring Command Success/Failure Criteria — *Continued*

Command	Success/Failure Criteria	Vector Processing Disposition
reply-best	Fails if any of the following are true: ■ Incoming call is not ISDN ■ Incoming trunk group is not administered for shared UUI or for QSIG Supplementary Service b. Otherwise: Succeeds and returns status data of best resource found in consider series.	Drop the call. Drop the call.
stop	Always succeeds.	Exit vector processing. Control is passed to normal call processing. Any queuing or treatment in effect remains in effect. Call is dropped if not queued.
wait-time	Always succeeds.	Connect the specified treatment and pass control to the delay timer. Any feedback is continued until other feedback is provided.

NOTE:
Complete operational details for the *route-to* commands are provided in Appendix H.

Call Vectoring Management and Monitoring

B

Introduction

Call Vectoring management involves a number of different considerations and tasks. This appendix describes these considerations/tasks. Specifically, the following topics are discussed:

- Call Vectoring feature requirements
- Upgrading to a Call Vectoring environment
- Changing a vector
- Testing a vector
- Identifying links to a vector
- Finding all occurences of a digit string

Implementation Requirements for the Call Vectoring Features

The tables appearing on the next several pages indicate the forms and the hardware required for implementing each of the Call Vectoring features. Details on completing the forms can be found in the *DEFINITY Enterprise Communications Server Release 6 Administration and Feature Description*, 555-230-522.

Table B-1. Basic Call Vectoring Requirements

Feature	Form(s)	Hardware
Basic Call Vectoring	■ Vector Directory Number Form ■ Hunt Group Form ■ Call Vector Form ■ Feature Related System Parameters Form	Announcement capabilities require either: ■ TN750 Integrated Announcement circuit pack(s), **or** ■ External announcement facility (analog announcements). Also, each analog announcement requires a port on a TN742, TN746B, or TN769 Analog Line circuit pack. **NOTE:** International equivalents for these circuit packs include the following: ■ TN467, TN468B (Australia and United Kingdom) ■ TN479 (Japan) ■ TN2135 (Italy) ■ TN2144 (Netherlands) ■ TN2149 (Belgium)

NOTE:
The Basic Call Vectoring and/or Call Prompting option(s) must be enabled on the System-Parameters Customer-Options form before the associated forms and the fields on the forms can be administered.

NOTE:
The TN750 Integrated Announcement circuit pack provides 16 ports for listening to announcements. The system provides for the installation of multiple TN750C Integrated Announcement circuit packs. See “Recorded Announcements” in the *DEFINITY Enterprise Communications Server Release 6 Administration and Feature Description*, 555-230-522, for more details.

Table B-2. Call Prompting Requirements

Feature	Form(s)	Hardware
Call Prompting	■ Vector Directory Number Form ■ Hunt Group Form ■ Call Vector Form ■ Vector Routing Tables Forms (G3V4 Enhanced)	Announcement capabilities require either: ■ TN750 Integrated Announcement circuit pack(s), **or** ■ External announcement facility (analog announcements). Each analog announcement requires a port on a TN742, TN746B, or TN769 Analog Line circuit pack. **NOTE:** International equivalents for these circuit packs include the following: ■ TN467, TN468B (Australia and United Kingdom) ■ TN479 (Japan) ■ TN2135 (Italy) ■ TN2144 (Netherlands) ■ TN2149 (Belgium)

NOTE:
The Basic Call Vectoring and/or Call Prompting option(s) must be enabled on the System-Parameters Customer-Options form before the associated forms and the fields on the forms can be administered.

Table B-3. G3V4 Enhanced Vectoring Requirements

Feature	Form(s)	Hardware
G3V4Enhanced Vectoring	■ Vector Directory Number Form ■ Hunt Group Form ■ Call Vector Form	Requires no hardware in addition to that required for Basic Call Vectoring.

NOTE:
The following options must be enabled on the System-Parameters Customer-Options form before G3V4 Enhanced Vectoring capabilities can be used: Basic Call Vectoring and/or Call Prompting option(s); and "Vectoring (G3V4 Enhanced)."

Table B-4. Advanced Vector Routing Requirements

Feature	Form(s)	Hardware
Advanced Vector Routing	■ Vector Directory Number Form ■ Hunt Group Form ■ Call Vector Form	Requires no hardware in addition to that required for Basic Call Vectoring.

NOTE:
The following options must be enabled on the System-Parameters Customer-Options form before Advanced Vector Routing conditionals can be programmed in a vector step: Basic Call Vectoring and/or Call Prompting option(s); "Vectoring (G3V4 Enhanced)"; and "Vectoring (G3V4 Advanced Routing)."

Table B-5. Vectoring (Best Service Routing)

Feature	Form(s)	Hardware
Single-site Best Service Routing	■ Vector Directory Number Form ■ Call Vector Form	No special hardware required for single-site BSR.
Multi-site Best Service Routing	■ Best Service Routing Application Plan form ■ Vector Directory Number Form ■ Call Vector Form ■ ISDN Trunk forms	Multi-site BSR requires no special hardware other than ISDN BRI/PRI connectivity between switches.

NOTE:
The following options must be enabled on the System-Parameters Customer-Options form before Best Service Routing commands can be programmed in a vector step. For single-site BSR: Basic Call Vectoring; "Vectoring (G3V4 Advanced Routing)"; and "Vectoring (Best Service Routing)." For multi-site BSR: Basic Call Vectoring; "Vectoring (G3V4 Advanced Routing)"; "Vectoring (Best Service Routing)"; and "Look-Ahead Interflow (LAI)."

Table B-6. ANI/II-Digits Requirements

Heading		
ANI/II-Digits Routing	■ Vector Directory Number Form ■ Hunt Group Form ■ Call Vector Form ■ Trunk Group Forms ■ Vector Routing Tables Forms	Requires no hardware in addition to that required for Basic Call Vectoring.

NOTE:
The following options must be enabled on the System-Parameters Customer-Options form: Basic Call Vectoring and/or Call Prompting option(s); "Vectoring (G3V4 Enhanced)"; and "Vectoring (ANI/II-Digits Routing)."

Table B-7. CINFO Requirements

Feature	Form(s)	Hardware
CINFO	■ Vector Directory Number Form ■ Hunt Group Form ■ Call Vector Form ■ Trunk Group Forms ■ Vector Routing Tables Forms	Requires no hardware in addition to that required for Basic Call Vectoring.

NOTE:
The following options must be enabled on the System-Parameters Customer-Options form: Call Prompting; "Vectoring (CINFO)."

Table B-8. Look-Ahead Interflow Requirements

Feature	Forms	Hardware
Look-Ahead Interflow	■ Trunk Group Form (ISDN-PRI) ■ CPN Prefix Table Form	Existing ISDN-PRI hardware can be used for Look-Ahead Interflow ISDN-PRI connectivity to the receiving switch. Interconnecting facilities must be ISDN-PRI with no interworking (that is, call connections that use both ISDN-PRI and non-ISDN-PRI facilities to complete) for the full capabilities of the feature to be operational. Look-Ahead Interflow calls that interwork may interflow successfully, but the ability to do so on an intelligent basis will be lost as will the Look-ahead DNIS information.

NOTE:
The Look-Ahead Interflow, ISDN-PRI, Basic Call Vectoring and (usually) private network access options must be enabled on the System-Parameters Customer-Options form.

NOTE:
Both the sending switch and the receiving switch must have the Basic Call Vectoring and the Look-Ahead Interflow features active. In addition, Look-Ahead Interflow calls can connect ISDN-PRI switch-to-switch using private, public, or SDN facilities.

Table B-9. Adjunct Routing Requirements

Feature	Forms	Hardware
Adjunct Routing	■ Hunt Groups ■ Class of Restriction (for Direct Agent Calls) ■ Call Vector ■ Station ■ Station (ISDN-BRI-ASAI)	**ISDN-BRI Connection** A TN556 ISDN-BRI circuit pack and a TN778 packet control must be in place. The latter provides packet bus control. Also, an adjunct/host processor must be in place to receive the request and select the route. A TN2198 two-wire BRI port circuit pack can be used in place of the TN556. In this case, an NT1 is also required. **MAPD Connection** MAPD hardware is a sandwich of two boards, the TN801 and a Pentium processor, which allows DEFINITY ECS to be connected to Ethernet and TCP/IP networks. The MAPD requires three contiguous slots on DEFINITY ECS: two slots are occupied by the MAPD unit, and the third is reserved for future use.

NOTE:
The Basic Call Vectoring options must be enabled on the System-Parameters Customer-Options form.

NOTE:
The ACD, ASAI Interface, and ASAI Capability Groups must be enabled on the system.

NOTE:
The Packet Bus option (G3r only) must be enabled on the Maintenance-Related System Parameters form before associated ISDN-BRI forms and fields can be administered.

NOTE:
Direct Agent Calls (DACs) remain in queue for the agent if the agent does not have call coverage. Otherwise, DACs follow the agent's coverage path after the no answer time interval expires.

Enabling the Vector Disconnect Timer

Call Vectoring makes available a Vector Disconnect Timer, which can be set for any amount of time between 1 and 240 minutes inclusive. The timer is enabled by selecting the timer field in the Feature-Related System-Parameters form. The timer is started when vector processing is started. Once the timer runs out, the call is dropped. The timer is canceled when vector processing terminates.

Enabling the timer allows queued calls that have not been answered within a determined amount of time to be dropped. For more information, refer to *DEFINITY Enterprise Communications Server Release 6 Administration and Feature Description,* 555- 230-522.

Upgrading to a Call Vectoring Environment

If you are already equipped with ACD and want to use Call Vectoring, the ACD environment must be upgraded to a Call Vectoring environment. This involves installing VDNs, vectors and hunt groups for the desired Call Vectoring feature(s).

The set of guidelines that follows is intended to serve as a general procedure for upgrading to a Call Vectoring environment. For complete details of this process, refer to *DEFINITY Enterprise Communications Server Release 6 Administration and Feature Description,* 555- 230-522.

1. Verify the vector options on the Customer Option Form.
2. Add the VDNs.
3. Evaluate the number of queue slots assigned to each split. Usually, you want to assign enough queue slots to allow all calls processed by Call Vectoring to be queued. (See the considerations for Basic Call Vectoring in Appendix C for more details.)

4. Change hunt-groups to be vector-controlled.
5. Administer the vectors and at least one test hunt group.
6. Test all of the vectors to be installed.
7. Change the trunk groups, night destinations, etc., to use the VDNs.

Changing and Testing a Vector

Vectors currently being used to process calls should not be changed because changes would have an immediate and uncertain effect on the treatment that the calls are receiving. Instead, a new vector should always be written.

In testing the vector, you should not consider the entire vector at once. Rather, you should first figuratively divide the vector into portions, then test each of these portions until the entire vector is tested.

After the new vector is thoroughly tested, the vector should be brought into service by changing the VDN to point to the new vector.

The set of following guidelines is intended to serve as a general procedure for changing and testing vectors. For complete details of this process, refer to *DEFINITY Enterprise Communications Server Release 6 Administration and Feature Description*, 555-230-522.

1. Check that a current version of the translation data is available.
2. Create a new VDN that points to the new vector. This VDN, which is temporary, is necessary to test the new vector.
3. Administer the new vector. Vector commands should be added and tested, one command at a time, starting with the first command. Be sure that each line is correct before proceeding to the next one.
4. Test the new vector with the new VDN. This ensures the new vector will function correctly when the vector is installed.
5. Install the new vector by changing the old VDN's vector assignment so that the VDNs now point to the new vector. Calls that are already being processed by the old vector will continue to be handled by that vector until the vector terminates vector processing.
6. Once all the calls are handled, remove the old vector and the VDN that was used for testing.

Identifying Links to a Vector

One or more VDNs always point to a vector. In addition, some vectors are linked to other vectors by *goto vector* commands or by *route-to* commands that point to a VDN. Before you delete or change a vector, you should identify all the VDNs and vectors that will be affected.

The **list usage vector *nnn*** command finds all the VDNs and vectors that send calls to vector ***nnn***, where "***nnn***" is the assigned vector number.

For example, let's say you want to delete vector 3. To determine what other elements of your system send calls to vector 3:

1. Type **list usage vector *3*** and press RETURN.

   The List Usage Report screen appears.

```
list usage vector 3                                          Page 1

                        LIST USAGE REPORT

Used By
Vector                Vector Number         1                Step 3
VDN                   VDN Number            58883
```

VDN 58883 points to vector 3. In addition, step 3 in vector 1 sends calls to vector 3. When you delete vector 3, you'll need to change this vector and VDN so they point to a different vector or delete them too.

Finding All Occurences of a Digit String

A single extension or an external phone number can be used in several elements in a complex vectoring system. When you modify VDNs or vectors, or when you change the phone numbers used in system elements such as *route-to* commands or Best Service Routing Plans, DEFINITY ECS allows you to find a specific digit string.

The **list usage digit-string *(1–16 digits)*** command finds the specified digit string in vectors, vector routing tables, and Best Service Routing Plans. The digit string can contain the numerals 0–9 and the characters *, #, ~, p, w, W, m, and s.

For example, to find the system elements that route calls to VDN 53338:

1. Type **list usage digit-string *53338*** and press RETURN.

The List Usage Report screen appears.

```
list usage digit-string 53338                                    Page 1

                          LIST USAGE REPORT

Used By
Vector                Vector Number         1                 Step  3
Vector                Vector Number         5                 Step  8
Vector                Vector Number         18                Step  4
Vector                Vector Number         37                Step 10
Best Service Routing Plan Number            1              Location  1
Best Service Routing Plan Number            2              Location  3
Best Service Routing Plan Number            5              Location  1
```

Three Best Service Routing Plans and steps in four different vectors route calls to this VDN. If you delete this VDN or assign a different extension, you'll need to update the extension used by these system elements.

Considerations for the Vectoring Features

C

Introduction

This appendix contains several lists of considerations you should bear in mind when using the Call Vectoring features. These considerations are intended to help you get the highest degree of productivity from Call Vectoring. For Look-Ahead Interflow considerations, see Chapter 7, "Look-Ahead Interflow".

NOTE:
If EAS is optioned, "skill" replaces "split."

Basic Call Vectoring Considerations

The following are considerations you should keep in mind when working with Basic Call Vectoring:

- Make the split queues large enough so that all incoming calls queue and are not dropped. If a queue is too small, a *queue-to split* or a *check split* command might fail to queue a call due to a lack of available queue slots. Accordingly, it is also always a good practice to include in the vector a step that checks a split's queue before queuing occurs and a corresponding step that provides alternate treatment if the queue is full. To check the queue size, you can use a *goto* command (for example, *goto Step 5 if calls-queued in split 20 pri l > 30*). The alternate treatment, which, if needed, is usually accessed by the *goto* command that checks the queue size, can queue the call to a backup split, make an unconditional Look-Ahead Interflow attempt, provide a busy signal, etc.
- A default treatment or a *route-to* destination step should be supplied after a *route-to* command in case the first destination is unavailable.

- Calls should not be queued to an unstaffed split (unless this is intended by the customer) without some alternate treatment.
- Interflow calls should not be permitted to interflow back and forth between a remote switch vector and a local switch. This process could cause a single call to use up all available trunks.
- After an announcement is provided, the audible feedback (such as music) should be re-attached.
- For ease-of-use purposes, each specific vector function or operation should be included in a separate vector and linked via one or more *goto vector* commands.
- In creating a vector, commands can be chosen and arranged in a manner such that answer supervision is delayed as long as possible. This should be done to keep down the service cost.
- The caller should always be provided with initial feedback (usually ringback).
- Direct agent calls merit special attention because such calls can affect call queuing. Although direct agent calls take up a queue slot, they are not always reported as using such a slot on CMS/BCMS reports (discussed in Appendix G). For example, a direct agent call is never counted toward the total of queued calls within a split (that is, the *calls-queued* test condition has no effect on this type of call).
- If it is necessary for a caller to hear an entire CONVERSANT script before talking to an agent, the caller should not be queued until after the *converse-on* step is executed.
- Audible feedback should be provided prior to a *converse-on step* whenever a large number of digits are to be outpulsed to the VRU.

Call Prompting Considerations

The following list includes considerations you should keep in mind when working with Call Prompting:

- To enter the digits requested via a *collect digits* command, outside callers must have a touch-tone telephone. For such callers using rotary dialing, a 10 second inter-digit timeout takes effect, and the *collect digits* command is omitted. As a precaution, a default treatment (for example, *route-to* attendant command, *queue-to split* command) should always be provided in the vector script unless the script is created exclusively for users of touch-tone telephones.
- If a caller does not enter the full number of digits specified in a *collect digits* step, an administered timeout occurs. Thereafter, vector processing continues with subsequent vector steps, and an attempt is made to process the call using the digits that have been collected. If the digits entered do not represent a valid destination, and if Automated Attendant is

being implemented via a *route-to digits* command, the *route-to digits* command fails, and vector processing continues at the next step, which should be a default treatment.

- It may be prudent to take steps in case a *route-to* attendant command fails, such as providing a disconnect announcement.
- From time to time, all of the system's touch-tone receivers might be in use. As a result, you should avoid starting your main vector with a *collect digits* command, since the caller on a DID or tie trunk in this case receives no audible feedback if he or she has to wait for a receiver to become available. Accordingly, it is a good practice to include some treatment (for example, a *wait-time 0 seconds hearing ringback* step) before the initial *collect digits* step. The *wait-time* step is not necessary if the collect step is collecting ced or cdpd digits.

Adjunct Routing Considerations

The following are considerations you should keep in mind when working with Adjunct Routing:

- Depending upon your application, you may want to include a second *adjunct routing* step in your vector in case the first such step fails.
- If you include an *announcement* step immediately after an *adjunct routing* step, be sure the announcement does not contain any information essential to the caller (such as further instructions) since the step following the *adjunct routing* step immediately terminates the moment the switch receives a destination from the ASAI adjunct.
- If you include a *wait-time* step after an *adjunct routing* step, it is a good idea to specify either *ringback* or *music* (and not *silence*) as the feedback. If the caller does not hear any feedback, he or she might give up on the call and hang up.
- The second step after the *adjunct routing* step could (and, in many cases, should) be implemented as a default treatment in case the host application or ASAI link is down. The step containing this default treatment (for example, *route-to number 0 if unconditionally*) executes immediately if the ASAI link is down and if the step is preceded by either a *wait-time* or an *announcement* step. On the other hand, if the host application is down, the default step executes only if the application does not respond with a route within 20 seconds.

Transferring Calls to VDNs Considerations

Care needs to be taken when writing a vector to which callers will be transferred. This is especially true if the vector manipulates or tests data that is delivered with the incoming call, such as ANI, II-digits, or CINFO digits.

To understand why care is needed, it is necessary to understand how a transferred call is treated. There are three main steps in a call transfer.

1. The transferring party hits the transfer button. The caller is put on hold. A second call is created with the transferring party as the originator.
2. The transferring party dials the VDN extension. Vector processing starts. The transferring party, not the caller, hears the initial vector provided feedback, if any.
3. The transferring party hits the transfer button for the second time. The two calls merge. The transferring party is dropped from the call. The caller becomes the originator of the new call. The caller now begins to receive vector provided feedback.

Between transfer steps 2 and 3 there is always a small but finite amount of time during which it is the transferring party who is connected to the vector. Any testing of ANI, II-digits, or CINFO digits during this time window applies to the transferring party and not to the caller. For this reason, it is recommended that vectors not start with an ANI, II-digit, or collect cdpd/ced step. Insert a delay of sufficient length to allow the transferring party to complete the transfer.

A delay is not required before a *collect x digits after announcement* step because a collect announcement is restarted for the caller when the transfer is complete.

VDN Return Destination Considerations

The VDN Return Destination feature allows an incoming trunk call to be placed back in vector processing after all parties, except the originator, drop. This feature is activated through switch administration of the VDN form. It is an optional system feature, and as such, it must be optioned on the System-Parameters/Customer-Options form.

A field on the VDN form allows the user to enter a VDN extension as a Return Destination. In this section, the VDN which has the Return Destination field administered will be called the "VDN with this feature active." The Return Destination VDN (the one specified in the new field) will be referred to as the "Return Destination."

Every incoming trunk call that is processed through a VDN with this feature active will be placed back in vector processing when all parties on the call, except the originator, drop. For this feature, the "originator" is the incoming party that originated the call at the time the call entered the VDN with this feature active.

NOTE:
Incoming calls on DCS ties do not go to VDN Return Destination.

The VDN that the call will be placed in (when the originator is the only remaining party) is determined by the "Return Destination". This VDN may be the same or different than the original VDN.

This feature is used to keep the call active and give the caller the opportunity to signal the need for sequence dialing (by entering a "#"). There are two ways this can happen:

1. When the destination drops on its own (after having answered), the call will go to the Return Destination which will have a *collect digits* vector step. This step will try to collect the "#" sign entered by the caller.
2. When the call is not answered, the caller enters the "#" to request sequence calling (this "#" will be collected by the ASAI-Requested Digit Collection feature). This "#" is reported to the adjunct. The adjunct requests the third_party_drop (or third_party_end_call) for the destination, and at that point the call goes to the Return Destination.

The VDN Return Destination and ASAI-Requested Digit Collection features may be used independently, with the following rules:

1. If there is no ASAI request to collect digits, but a Return Destination is provided: when all parties, except the originator, drop, the switch will route the call with only one party active (the caller) to the Return Destination. At this point, the call enters vector processing for the VDN specified by the Return Destination.

   The caller will keep returning to this same return destination indefinitely until either the caller hangs up or a busy or disconnect vector step is executed. Once a call leaves vector processing for the first time, the return destination will never be changed.
2. If a request is made to collect digits but there is no Return Destination provided: the switch will collect the digits and pass them on to the ASAI adjunct. It will be up to the adjunct to take action. However, if the action taken by the adjunct is to drop one party on the call, the switch will drop the other party as well and clear the call (it cannot retain a call with only one party, if there is no Return Destination for further processing).

User Scenario — Remote Access with Host Provided Security

A customer may use the VDN Return Destination feature to provide a more flexible remote access feature together with host-based call security. The remote user/caller does not have to call back into the switch when multiple destinations need to be reached nor does the caller have to enter his/her identification every time a new destination is desired.

This system consists of three VDN/vector pairs. The first VDN uses the vector shown in Screen C-1.

```
1. collect 6 digits after announcement 1001 ("Please enter
     your identification number and password followed by # sign")
2. adjunct routing link 1221
3. wait-time 6 seconds hearing silence
4. disconnect after announcement 1003 ("We are sorry, but we are
     experiencing technical difficulties at this time, please try
     again later")
```

Screen C-1. Sample Vector for Remote Access

In this scenario, a remote caller calls into the switch by dialing the first VDN. The vector in Screen C-1 prompts the caller to enter an identification number and a password that will be passed, via the adjunct routing vector command, to the host for validation. The host can keep track of invalid attempts or decide to de-activate or activate certain identification numbers based on customer set criteria. If the host is not available, the call will be disconnected after an announcement (vector step 4 above).

```
1. collect 16 digits after announcement 1002 ("Please enter
     the telephone number of your destination, followed by # sign")
2. adjunct routing link 1222
3. wait-time 6 seconds hearing silence
4. disconnect after announcement 1003 ("We are sorry, but we are
     experiencing technical difficulties at this time, please try
     again later")
```

Screen C-2. Sample Return Destination Vector with Disconnect

If the ID and password are valid, the adjunct specifies a route to the second VDN, which uses the vector in Screen C-2. The switch collects digits for the destination that the caller wants to reach (vector step 1 above). The host receives the number entered by the caller (vector step 2 above) and validates the entered number to check if the caller is allowed to reach the specified destination. If so, the host routes the call to the destination. After the called destination disconnects from a call, the caller can remain on the line to be connected to the Return Destination, which points to the same vector.

NOTE:
If the ID or password entered at the first VDN is invalid, then the call can be routed to a third VDN. The vector for this VDN (not shown) consists simply of a *disconnect after announcement* step with an appropriate announcement. The invalid call attempt is logged.

The caller, once connected to the Return Destination, can enter a second destination/phone number to connect to. The host performs the same validation on the destination number as in the first destination and routes the call as

appropriate (destination entered by caller or alternate destination). Note that the host can also provide reports on all the destinations and times reached by each remote user.

In the Return Destination vector, it is recommended that the first vector command give the caller the opportunity to disconnect from the call rather than immediately routing the call to some destination. If the call was immediately routed and then the caller decided to hang-up, the destination that the call was routed to would ring, alerting the called party, but then no one would be on the line at the other end (this could be confusing to customers, and could be misinterpreted as a problem with the feature). Vector commands such as *wait-time*, *collect after announcement*, and *announcement* can provide the caller with the opportunity to disconnect before the call is routed. As an example, an *announcement* command with the recording "Please hang-up to end your call, or remain on the line if you wish to place another call" instructs the caller to disconnect, before the call is routed.

User Scenario — Saving in Trunk Facilities Between Call Centers

You can also use VDN Return Destination to return a call to a local agent after the call is transferred to a remote destination (call). This eliminates the need for the remote agent to transfer the caller back to a local agent and will save in switch trunk facilities, since each time the call is transferred back to a local agent an additional trunk is being used by the call.

For example, calls can be received at the local call through a VDN that has the return destination administered. These calls are delivered to an agent on the local switch. If the local agent transfers the call to a remote destination (because the caller needed to talk to an agent on the remote switch), the call returns to the Return Destination after the remote switch drops the call. The remote switch agent must inform the caller to remain on the line after they are finished and the remote agent just needs to disconnect from the call (hang up).

The Return Destination for this scenario should include an *announcement* vector command at the beginning to inform the caller to disconnect from the call, if they do not want to be reconnected to an agent on the local switch. A sample Return Destination vector will be as shown in Screen C-3.

```
1. announcement 1004 ("Please remain on the line, if you want
     to talk a to another representative")
2. queue-to split 101 pri m
3. announcement 1005 ("All our representatives are busy,
     please wait")
4. wait-time 60 secs hearing silence
5. goto step 3 if unconditionally
```

Screen C-3. Sample Return Destination Vector with Announcement

Advanced Multi-Site Routing

D

Introduction

This appendix supplements the main chapters on Look-Ahead Interflow and Best Service Routing™. The BSR chapter explains the basic concepts and operation of the Best Service Routing feature and provides instructions for implementing single- and multi-site BSR. The chapter on LAI also explains the ISDN trunk group settings and interactions relevant to information forwarding, UCID, and multi-site routing.

Here, we offer an “advanced course” in multi-site applications—especially multi-site BSR—for those who want to develop an in-depth understanding of their design and operation. In addition, this appendix will be useful for readers whose call center networks meet either or both of the following criteria:

- 5 or more switches in the network
- Combination of low- and high-volume locations

Application Architecture in Multi-Site BSR

Multi-site applications may be structured in a variety of ways. In general, however, most applications will fit one of two models: distributed or centralized. When each switch in a network may interflow calls to other switches and receive interflows, this is called a distributed system. A centralized system, by contrast, is one in which all calls are initially delivered to a single call center (the “hub”) and distributed from this site to queues at remote switches. A centralized system requires greater inter-switch trunking, since a greater percentage of calls need to be redirected. However, it may be an appropriate configuration if your organization has a significant investment in VRU and CTI technology at the hub.

Which architecture you choose for an application has direct implications for your choice of user adjustments and polling patterns.

User Adjustments

User adjustments in *consider split* and *consider skill* steps may be set at the user's discretion. In distributed multi-site applications, however, adjustments must be carefully considered because of their potential affect on costs and inter-switch trunk capacity. In centralized applications all calls are redirected anyway so it's OK to use adjustments of "0". In distributed applications, though, a user adjustment of "0" for a *consider location* step is almost never practical or efficient.

In distributed applications, the smaller the adjustment the closer the load balance across the network, but the greater the percentage of calls redirected between switches (and thus the greater the demands on inter-switch trunking). Higher adjustments reduce interflows, but at the cost of allowing greater imbalance in the load between switches. It will take some time and effort to find the best combination of user adjustments in any particular network, but Table D-1 contains recommended ranges for initial user adjustments under different conditions. Adjustments may vary between different call center applications so apply these guidelines for each of your applications separately.

Table D-1. Recommended Initial User Adjustments

Lucent recommends adjustments of…	If the following criteria apply…
10–15	■ You want to balance wait times across the network as much as possible. ■ Trunk facilities between switches are plentiful. ■ Each switch receives more than 1 call every 10–15 seconds (more than 240–360 calls/hour) for this application.

Continued on next page

Table D-1. Recommended Initial User Adjustments — *Continued*

Lucent recommends adjustments of…	If the following criteria apply…
20	■ Balancing wait times across the network is important to you. ■ Adequate trunk facilities are available to support the desired balance. ■ Each switch receives more than 1 call every 20 seconds (more than180 calls/hour) for this application.
30 or higher	■ Gains in agent efficiency are more important to you than balancing wait times across the network. ■ Trunk facilities are scarce. ■ Call interflow is costly. ■ Each switch receives no more than 1 call every 30 seconds (around 120 calls/hour or lower) for this application.

In your first multi-site application, Lucent recommends beginning with a remote adjustment of 30. This can easily be reduced later if inter-switch trunking is under-utilized. On the other hand, if trunk exhaustion is a common occurrence then user adjustments are probably set too low. Care should be taken not to lower remote user adjustments to such an extent that all trunk resources are regularly exhausted. When trunks are exhausted, no further load balancing can take place and the overall balance may deteriorate.

User adjustments should also be set high enough that calls are not interflowed to gain the equivalent of a fraction of a queue position. The following equation will give you the minimum recommended user adjustment for each remote switch:

$$\frac{AverageCallHandlingTime}{NumberOfFullTimeEquivalentAgents} \leq UserAdjustment$$

Adjustments for remote locations will probably be in the range of 10–30 in most distributed applications.

Is there a reliable relationship between user adjustments and the balance in wait times across a network?

Changing conditions can produce significant variations in such a balance, but on average you can predict the balance in wait times for a given user adjustment.

Let's say a user adjustment of 20 is chosen for all remote resources in a network and all the remote sites are polled. When waiting times are short (< 100 secs), the highest and lowest EWTs for this application on the network should stay within a range of approximately 20 seconds (30–50 seconds, for example). When waiting times are long (> 100 secs), the highest and lowest EWTs for the application should stay within a range of approximately 20% (5–6 minutes, for example).

Status Polling in BSR

Status polls are the key element in multi-site BSR applications. Status polls provides the communication links between a switch that wants to interflow a call and the switches that might service that call.

The vectors you write in multi-site applications must balance the costs of time and trunk usage with the benefit of better customer service. BSR is designed to help you achieve this balance, incorporating mechanisms to maximize improvements in customer service while minimizing inter-switch communications with its attendant delays and trunk usage. This section explains those mechanisms and the benefits they provide as you write vectors.

How Long do Status Polls Take?

One *consider location* step polls one remote location. Does this mean that an optimal multi-site BSR application polls every switch in a network? No.

Let's look at an example of a moderately large network, containing 16 switches. The primary vector on switch #1 could be written as shown in Screen D-1. Polling response times are variable. Let's assume that this is a slow response network and that each status poll takes 1 second. The consider series in this vector could add as much as 15 seconds to a call's time in vector processing! In fact, the vector in Screen D-1 is provided as an example of what NOT to do. The benefits of BSR can be obtained much more efficiently.

```
1.   wait time 0 secs hearing ringback
2.   consider skill  1 pri m  adjust-by  0
3.   consider skill  2 pri m  adjust-by 20
4.   goto step 20 if expected-wait for best = 0
5.   consider location 1      adjust-by 30
6.   consider location 2      adjust-by 30
7.   consider location 3      adjust-by 30
8.   consider location 4      adjust-by 30
9.   consider location 5      adjust-by 30
10.  consider location 6      adjust-by 30
11.  consider location 7      adjust-by 30
12.  consider location 8      adjust-by 30
13.  consider location 9      adjust-by 30
14.  consider location 10     adjust-by 30
15.  consider location 11     adjust-by 30
16.  consider location 12     adjust-by 30
17.  consider location 13     adjust-by 30
18.  consider location 14     adjust-by 30
19.  consider location 15     adjust-by 30
20.  queue-to best
21.  announcement 1001
22.  wait time 60 secs hearing music
23.  goto step 21 if unconditionally
```

Screen D-1. Primary Vector on One Switch in a 16-Switch Network

First, even in very large networks you can obtain nearly all of the possible benefits in agent utilization with very few polling connections. In a network of 16 switches, 99% of the total benefits possible with BSR can be obtained if each switch polls just 4 others. (See "How Many Switches Should One Switch Poll?" on page D-7 for more information on this topic.)

Now our vector looks like Screen D-2. Is polling time now cut from 15 seconds to 4 seconds, proportional to the reduction in *consider* steps?

```
1.   wait time 0 secs hearing ringback
2.   consider skill  1 pri m  adjust-by  0
3.   consider skill  2 pri m  adjust-by  0
4.   goto step 9 if expected-wait for call = 0
5.   consider location 5      adjust-by 30
6.   consider location 10     adjust-by 30
7.   consider location 13     adjust-by 30
8.   consider location 15     adjust-by 30
9.   queue-to best
10.  announcement 1001
11.  wait time 60 secs hearing music
12.  goto step 10 if unconditionally
```

Screen D-2. Primary Vector on One Switch in a 16-Switch Network

In fact, polling time in this vector may be around 0.4 seconds per call because of mechanisms in BSR that constantly react to network conditions and resource usage to minimize the number of status polls. These mechanisms, whose combined operation is called “intelligent polling,” also function to make each status poll as productive as possible.

Intelligent Polling

A BSR application will only poll the switches that are likely to provide the best service at any given time. If a remote switch is polled and returns an adjusted EWT greater than that of the current best resource, polling of the remote switch will be suppressed for a period of time proportional to the difference between the two adjusted EWT values. (In other words, polling of a given location is suppressed whenever the adjusted EWT returned by that location is subsequently replaced by a better adjusted EWT from another resource.) The *consider* step for this location will be skipped during this period and vector processing will continue at the next step. When the suppression period is over, the *consider* step will once again poll this location. If the location returns the best adjusted EWT, the next call processed by the vector will also cause this location to be polled. If it is not the best, polling will again be temporarily suppressed, and so on.

If no calls are in queue at the remote location an agent might become available at any moment, and thus BSR will never suppress polling for longer than 5 seconds in such situations. BSR will never suppress polling of any remote location for more than 60 seconds, regardless of the differences between adjusted EWT returned by different switches.

Other conditions can also suppress status polls to a location:

- resource exhaustion (no trunks available, queue full)
- administration errors (badly written vectors, or no application plan)

This feature significantly reduces the average number of status polls placed per call. The greater the call volume, the greater the percentage reduction. Let’s take another look at the vector in Screen D-2.

Let’s assume that the network is operating in a balanced state. EWTs are 30 seconds at all locations, and a call arrives every 3 seconds at each site. Adjusted EWTs are 30 seconds at the origin switch and 60 seconds for each remote switch. After each status poll under these conditions, polling will be suppressed for 30 seconds. Each remote location is polled therefore, by every 10th call. On average, this means that each call polls any one location 0.1 times. Since there are four *consider* steps, each call makes 0.4 polls. Remembering the 1-second polling response time given at the beginning of the example, the average time added to call processing for each call is 0.4 seconds.

The 1st-found available agent strategy, discussed in Chapter 8, “Best Service Routing™”, can cut average polling times further. With the 1st-found strategy,

BSR will skip all subsequent *consider* steps in a series if a resource with an available agent is found and deliver the call to that resource.

Efficient Polling Patterns in Large Networks

Unless you have a small network, you won't benefit by having every switch poll every other switch. This section explains how many remote locations each switch needs to poll, and it provides guidelines for selecting which locations any given switch should poll.

How Many Switches Should One Switch Poll?

It's not necessary to poll every switch in larger networks. Because of BSR's intelligent polling capabilities, you can obtain 99% of the possible benefits in agent utilization with very few polling connections.

For an example, let's look at a laboratory network of 16 switches that Lucent used for simulations of BSR multi-site applications. As Table D-2 shows, approximately 99% of the possible benefits were obtained when any one switch polled 4 others.

Table D-2. Effectiveness of Status Polls in a 16-switch Network

Number of remote sites polled by each switch	ASA across the network (sec.)	Approximate percentage of total benefits obtained
0	192.8	0%
1	26.2	89%
2	10.6	95%
3	7.6	98%
4	6.5	99%
15	4.7	100%

For each switch to poll the other 11 switches in the network would only produce an additional 1% gain in ASA and agent utilization—an improvement which would be more than offset by the cost of additional messaging and trunking.

In most situations, you'll obtain the optimal results with your multi-site BSR applications if you follow the polling guidelines in Table D-3.

Table D-3. Recommended Number of Locations to Poll

If there are this many switches in the network…	Each switch should poll…
2–4	all the other switches
5–10	3 other switches
11–20	4 other switches
21–40	5 other switches
41 or more	6 other switches

Which Remote Switches Should Each Switch Poll?

In networks with fewer than 5 switches, each switch can productively poll all the other switches in the network. In larger networks, each switch need not poll every other switch. But which switches should each switch poll? We'll use the term "polling patterns" to describe the relationships between switches in multi-site BSR applications.

Here are two patterns to avoid. They're simple and seem intuitively obvious, but they don't usually yield the best possible results:

- Mutual polling: As much as possible, 2 switches shouldn't poll each other. This is unavoidable in small networks, but in large networks it can and should be minimized.
- Polling chains: For example, if switch A polls B & C, B polls C & D, and so on, this is a polling chain.

You may want to experiment with polling patterns appropriate to your own network and applications (if you're not constrained by the physical structure of your network). The following table provides a template for creating polling patterns for applications of up to 12 switches. In the majority of situations, these patterns will produce results that are close to optimal. To use this table, first assign a number from 1 to x to each switch in your application. Next, find the column that matches the number of switches in your application. As you read down that column, you'll see which switches each particular switch in the application should poll.

Table D-4. Polling Patterns for Networks of 5–12 Switches

This switch…	Should poll the specific switches shown in the column for your network size							
	5	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	2,4,5	2,4,5	2,4,6	2,4,7	2,4,6	2,4,7	2,4,8,10	2,4,8,9
2	3,5,1	3,5,6	3,5,7	3,5,8	3,5,7	3,5,8	3,5,9,11	3,5,9,10
3	4,1,2	4,6,1	4,6,1	4,6,1	4,6,8	4,6,9	4,6,10,1	4,6,10,11
4	5,2,3	5,1,2	5,7,2	5,7,2	5,7,9	5,7,10	5,7,11,2	5,7,11,12
5	1,3,4	6,2,3	6,1,3	6,8,3	6,8,1	6,8,1	6,8,1,3	6,8,12,1
6		1,3,4	7,2,4	7,1,4	7,9,2	7,9,2	7,9,2,4	7,9,1,2
7			1,3,5	8,2,5	8,1,3	8,10,3	8,10,3,5	8,10,2,3
8				1,3,6	9,2,4	9,1,4	9,11,4,6	9,11,3,4
9					1,3,5	10,2,5	10,1,5,7	10,12,4,5
10						1,3,6	11,2,6,8	11,1,5,6
11							1,3,7,9	12,2,6,7
12								1,3,7,8

In applications of more than 12 switches, Table D-5 provides the formulae you need to figure out the optimal polling pattern.

Table D-5. Polling Pattern Formulae for Large Networks

Number of switches in application	Switch “i” should poll…
13 or 16	i + 1, i + 3, i + 7, i +11
14 or 19	i + 1, i + 3, i + 7, i + 9
15	i + 1, i + 3, i + 7, i + 10
17 or 20	i + 1, i + 3, i + 7, i + 12
18	i + 1, i + 3, i + 7, i + 13
21–23	i + 1, i + 3, i + 7, i + 15, i + 17
24	i + 1, i + 3, i + 7, i + 15, i + 19
25	i + 1, i + 3, i + 7, i + 15, i + 20

To use one of these formulae, first assign a number from 1 to x to each switch in your application. Then, in the left-hand column of the table, find the number of

switches in your application. The corresponding formula in the right-hand column is the one you should use.

In the formulae, “i” is the number of the switch for which you’re calculating a polling pattern. For example, let’s say you want to calculate the polling patterns in an application with 16 switches. The formula to use is

$$i + 1, i + 3, i + 7, i + 11$$

as shown in the first row of the table. Here are the actual results of this formulae for the first 5 switches in this 16-switch application. Notice that the numbers

Switch number…	Should poll switches…
1	2 ,4, 8, 12
2	3, 5, 9, 13
3	4, 6 ,10, 14
4	5, 7, 11, 15
5	6, 8, 12, 16
6	7, 9, 13, 1
7	8, 10, 14, 2

“wrap” (start over at 1) after you’ve polled the last switch in the network: switch 5 polls switch 16 as its fourth poll, and then the polling pattern for switch 6 has switch 1 in the fourth position.

Considerations for Low Volume Splits and Skills

Very small resources (for example, 2–3 agents) have special needs. With BSR, it is easy to obtain a very close balance of wait times across a network of call centers. however, for very small splits or skills, wait times for each call can vary significantly.

To see why this is, let’s take an extreme example of a split with a single agent logged in with one call active and none in queue. Average call handling time is 3 minutes. Now, if a new call arrives in queue, that call could be answered almost immediately—or it might wait for 3 minutes or more. The variation in wait times is perhaps 5–180 seconds.

In general, the fewer agents logged into a split or skill, the greater the variability in wait times because agents become available less often. BSR will naturally favor large resources, steering calls away from smaller resources when there are no available agents or wait times are not the best in the application. This

tendency helps reduce the possibility that an individual caller might have a disproportionately long wait at a small resource.

If your network includes very small splits or skills, you have three options:

- If your operation is not badly affected by a small percentage of calls having variable wait times, simply use BSR normally across the network.
- If your principal concern is that a call does not wait in queue while an agent is available elsewhere, use BSR normally but write primary vectors at smaller locations to perform rapid look-ahead attempts to other resources once the call has been queued. (Rapid LAI vector loops use the *interflow-qpos* conditional, which is an enhancement to LAI. For more information on Look-Ahead Interflow and the *interflow-qpos* conditional, see Chapter 7, "Look-Ahead Interflow".) For an example of this type of vector, "Using LAI as a Backup" on page D-11.
- If you want to answer every caller quickly, then Lucent recommends the following configuration. Do not deliver or queue calls directly to the very small resources. Deliver or queue all incoming calls to larger resources, and use BSR to balance the load across these larger locations. Some or all of the larger locations should then perform rapid look-ahead attempts to one or more of the smaller resources. In this way, the members of the very small resource become an extension of the agent pool at one of the larger call centers. For an example of this design, see "Single-Queue FIFO Hybrid Configuration" on page D-12.

In any network, avoid having several large resources poll or make lookahead attempts to a very small resource. Since the status at the very small resource changes infrequently, frequent polls to that resource are wasteful. A very small resource should receive lookahead attempts or be polled only by other small resources or by one large resource.

Minimizing Variations in Wait Time

When a network contains (or when a call center application combines) large resources and very small resources, BSR and Look-Ahead Interflow (LAI) can be effectively combined. This section presents two sample vectors. The first example shows a primary vector intended for the smaller resources in a network when you want to avoid having a call in queue at one call center while an agent is available at another. This design will reduce wait time variation as well. The second example illustrates a primary vector for larger locations: this example shows you the best way to minimize wait times across a network

Using LAI as a Backup

As noted above, if your principal concern is that a call not wait in queue while an agent is available elsewhere, use BSR at all locations in the network. At smaller

locations, write primary vectors that will perform rapid LAI attempts to other (preferably larger) resources once the call has been queued.

```
1.  wait time 0  secs hearing ringback
2.  consider skill 1st pri m adjust-by  0
3.  consider location 12     adjust-by 30
4.  consider location 22     adjust-by 30
5.  goto step 7 if expected-wait for call < 600
6.  disconnect after announcement 3501 "Due to heavy call volume..."
7.  queue-to skill best
8.  announcement 3500 "Thanks for calling...."
9.  goto step 13 if expected-wait for call < 90
10. wait time 45 secs hearing music
11. announcement 3502 "Still busy..."
12. goto step 9 if unconditionally
13. route-to-number 913031234567 with cov n if interflow-qpos = 1
14. wait time 5 secs hearing music
15. goto step 13 if unconditionally
```

Screen D-3. BSR/LAI Primary Vector for Smaller Resources

Steps 1 to 4 comprise a typical BSR vector. The origin switch considers a local resource and 2 remote resources. Before queuing or routing the call, however, the vector checks the expected wait time for the best resource. If this is 10 minutes or more, the caller receives a ‘busy” announcement. Otherwise, the queue-to best step sends the call to the best resource. Two vector loops follow: one 45-second loop with music and a delay announcement, and one 5-second loop that uses Look-Ahead Interflow. If the call is queued successfully in step 7 the first announcement loop (steps 9-12) executes until the call gets within a certain range of the head of the queue (at which point EWT is less than 90 seconds). At this time, step 9 sends the call to the second loop, where LAI attempts are placed every 5 seconds for the call at the head of the interflow eligible queue (*interflow-qpos=1*). If an agent becomes available at the larger remote resource, any call at the head of the eligible queue at the smaller location is outflowed to the larger resource, normally within a period of 5 seconds.

Single-Queue FIFO Hybrid Configuration

To minimize variations in wait time across a network, the best strategy may be to let only the call centers with the larger resources receive calls. Figure D-1 shows a network of 3 large and 3 small resources (call centers with large splits/skills and call centers very small splits/skills in the same application).

The large locations use BSR and all poll each other, while each location with a small resource (numbered 1, 2, 3) is treated as a satellite of one of the larger locations and only receives calls interflowed from that location. (Mutual polling is not optimal in larger networks, but it’s OK for switches in such a small network to poll each other.) So BSR is used to balance the load between the locations with the larger resources. Then, each large switch executes a rapid LAI vector loop to one small switch to look for available agents. Since calls never queue at the small

switches, the problem of highly variable wait times at the small resources is eliminated. This strategy will also give the best balance in wait times across resources.

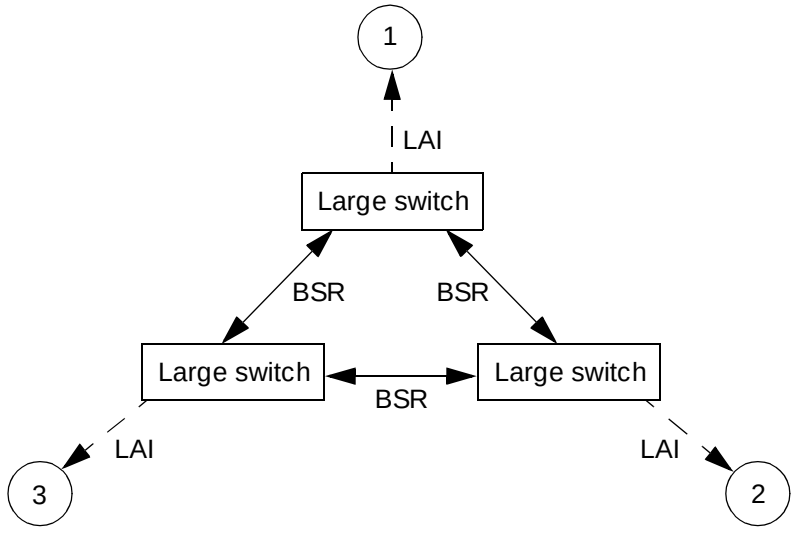


Figure D-1. Hybrid Application Architecture

Screen D-4 shows the primary vector that would be used at the large locations with this strategy. This vector is almost identical to the vector shown in "Using LAI as a Backup" above. The differences are at the application level. In contrast to the previous example:

- Only the locations with the larger resources receive calls.
- The primary vector shown here resides on the larger switches.

Steps 1 to 4 comprise a typical BSR vector. The origin switch considers a local resource and 2 remote resources. Before queuing or routing the call, however, the vector checks the expected wait time for the best resource. If this is 10 minutes or more, the caller receives a 'busy" announcement. Otherwise, the queue-to best step sends the call to the best resource. Two vector loops follow: one 45-second loop with music and a delay announcement, and one 5-second loop that uses Look-Ahead Interflow. If the call is queued successfully in step 7, the first announcement loop (steps 9-12) executes until the call gets within a certain range of the head of the queue. At this time, step 9 sends the call to the second loop, where LAI attempts are placed every 5 seconds (only for the call at the head of the interflow eligible queue). If an agent becomes available at the

smaller resource, any call at the head of the eligible queue at the larger location is outflowed to the smaller resource, normally within a period of 5 seconds.

```
1.  wait time 0  secs hearing ringback
2.  consider skill 1st pri m adjust-by  0
3.  consider location 120    adjust-by 30
4.  consider location 220    adjust-by 30
5.  goto step 7 if expected-wait for best < 600
6.  disconnect after announcement 3501 "Due to heavy call volume..."
7.  queue-to skill best
8.  announcement 3500 "Thanks for calling...."
9.  goto step 13 if expected-wait for call < 90
10. wait time 45 secs hearing music
11. announcement 3502 "Still busy..."
12. goto step 9 if unconditionally
13. route-to-number 913031234567 with cov n if interflow-qpos = 1
14. wait time 5 secs hearing music
15. goto step 13 if unconditionally
```

Screen D-4. Vector Combining BSR and LAI

Similar vector loops can be added to the interflow vectors at each of the large switches. In other words, each vector that processes calls at the larger locations can use rapid LAI loops to interflow calls to its satellite resource. This system maximizes agent utilization and the distribution of call load while evening out wait times across the network.

Advanced Information Forwarding

This section explains ISDN (BRI or PRI) trunk group setting interactions with Information Forwarding, UCID, and Multi-Site Routing.

User information included in the SETUP message for an outgoing call (at the sending DEFINITY) or DISCONNECT message sent back for an incoming call (at the receiving DEFINITY) is based on the trunk group settings at the sending or receiving sites. Pre-R6.3 software transported ASAI User to User Information in a UUI IE without any other data (a "non-shared" codeset 0 UUI IE). Lookahead Interflow data was transported via a codeset 6/7 LAI IE. R6.3 “shared” user information forwarding supports new data items (UCID, collected digits and In-VDN-Time) in addition to “shared forwarding” of LAI Info (VDN-Name and Other-LAI) and ASAI provided user data. “Shared forwarding” over non-QSIG ISDN trunks packs the data items in a codeset 0 UUI IE (called shared UUI), where each item consists of a two-byte header (application ID and data length). “Shared forwarding” over QSIG trunks transports the data items as Manufacturer Specific Information (MSI) in codeset 0 Facility IEs.

Best Service Routing (BSR) and “shared data forwarding” (UCID and other new data items) requires QSIG or the **shared** `UUI IE Treatment` setting with non-QSIG trunk groups on both the sending (outgoing trunks) and receiving

(incoming trunks) at DEFINITY switches. Shared settings are also required on tandem trunk connections through DEFINITY switches that route these calls. In R6.3, LAI Info, UCID, collected digits, In-VDN-Time and ASAI provided user data can be forwarded with a call in the SETUP message (LAI or BSR interflowed call, a tandemed call, for UCID with any outgoing call and for ASAI user data any adjunct routed outgoing call). With R6.3 capabilities, only BSR reply-best data is returned with a BSR poll call and only ASAI user data is returned for a non-poll call in a DISCONNECT message (both types of data will not be included in the same DISCONNECT message). Shared UUI Priority settings do not affect what is put in the DISCONNECT message or data forwarded over QSIG trunks.

The protocol (QSIG or non-QSIG) is set on page 1 of the ISDN trunk group form using the `Supplementary Service Protocol` field. QSIG type as defined for shared MSI is protocol type **b** (another protocol type "d", ECMA QSIG is considered non-QSIG for Information Forwarding). The `Send Codeset 6/7 LAI` field on page 2 indicates whether or not to include an LAI IE in the SETUP message. The codeset used (6 or 7) is determined by the "Codeset to Send TCM, Lookahead" field on page 1. The `Send UCID` field on page 2 indicates whether or not the UCID data item should be included as user information with calls routed over this trunk group. The `Send Codeset 6/7 LAI IE` field is ignored for BSR polls over the trunk group (an LAI IE will never be included with BSR calls).

Non-QSIG Protocol

`UUI IE Treatment` set to **service-provider** includes any application provided UUI in a codeset 0 UUI IE on a non-shared basis. That is, the data portion of the UUI IE only includes user info in the SETUP or DISCONNECT messages as provided by an application such as ASAI without the shared App-ID and length header fields. User data from only one application can be included in non-shared UUI. This setting would be used for non-QSIG trunk groups to pre-R6.3 DEFINITY switches or other situations when service-provider functionality is wanted (e.g., where shared forwarding of the new data items is not required or for trunk groups to other vendor switches or network services that need user information from the trunk group in a non-shared UUI IE such as provided by ASAI). Incoming calls received with shared user information (shared UUI IE) that are routed outgoing over a non-QSIG **service-provider** trunk group will forward only ASAI provided user data in a non-shared UUI IE.

`UUI IE Treatment` set to **shared** allows all applications (with R6.3: LAI, UCID, BSR and ASAI) to include data items in the UUI IE on a shared forwarding basis. The Shared UUI Feature Priorities page settings along with the `Max. Size of UUI Contents` field on page 2 and the features configured for the system determines what actually is included in the UUI IE. This is the normal setting for non-QSIG trunk groups that route calls to R6.3 or later DEFINITY switches over private or public networks when information forwarding is required and must be used for Best Service Routing.

QSIG trunk group

`UUI IE Treatment` set to **service-provider** forwarded ASAI provided user data in a non-shared codeset 0 UUI IE and all other user data in codeset 0 Facility IEs as MSI. In this case the `Max. Size of UUI Contents` field is not relevant and the Shared UUI Feature Priorities page does not show nor apply. This setting would only be used for QSIG trunk groups to pre-R6.3 DEFINITY switches for compatibility with existing ASAI applications (MSI is discarded at a pre-R6.3 endpoint) or when service-provider functionality is wanted (e.g., where shared forwarding of the new data items is not required or for trunk groups to other vendor switches that need user information from the trunk group in a non-shared UUI IE such as provided by ASAI). Incoming calls received with shared data (shared UUI IE) routed out over a QSIG **service-provider** trunk group, will separate any ASAI provided user data included in the shared UUI IE and forward it in a non-shared UUI IE.

`UUI IE Treatment` set to **shared** will forward all user information including ASAI provided user data in codeset 0 Facility IEs as MSI in the SETUP or DISCONNECT message. The UUI IE is never included over a **shared** QSIG trunk group. In this case, the `Max. Size of UUI Contents` field and the Shared UUI Feature Priorities page do not apply. This is the normal setting for QSIG trunk groups to R6.3 or later DEFINITY switches when information forwarding is required and must be used for Best Service Routing.

"Send Codeset 6/7 LAI IE" Option Interactions

The `Send Codeset 6/7 LAI IE` option is independent of the `Supplementary Service Protocol` and `UUI IE Treatment` settings to allow additional flexibility. The R6.3 or later DEFINITY can have a mix of trunk groups set with non-QSIG or QSIG protocol and with **service-provider** or **shared** settings. Calls interflowed over the **shared** non-QSIG trunk groups will contain the data items to be forwarded with the call in the UUI IE while calls interflowed over the non-QSIG **service-provider** trunk groups will not (except for ASAI which can always be sent in UUI). Calls interflowed over R6.3 or later QSIG trunk groups will always have MSI user information (except for ASAI whose transport method depends on the `UUI Treatment` setting).

When a call is LAI interflowed over a non-QSIG **service-provider** trunk group, the `Send Codeset 6/7 LAI IE` option being active will result in just the LAI IE being forwarded with the call in a SETUP message. When interflowed over a non-QSIG **shared** trunk group, setting the `Send Codeset 6/7 LAI IE` to yes includes a codeset 6/7 LAI IE in the SETUP message in addition to the same LAI information included as shared data in the UUI IE. If necessary and appropriate, the LAI information fields (and others) can be set to "blank" on the Priorities page to exclude these data items from the UUI IE. See "Determining User Information Needs" on page 9-7 for details. When interflowed over a QSIG **service-provider** or **shared** trunk group with `Send Codeset 6/7 LAI IE` active, the LAI information will be included as both MSI and in the LAI IE.

However, in this case there is no mechanism to eliminate the duplication of data if the codeset 6/7 LAI IE is required.

These combinations can be used when calls are LAI interflowed to pre-R6.3 DEFINITY switches with existing ASAI applications using ASAI provided UUI that may or may not be using the LAI IE. The MSI will be ignored at the pre-R6.3 switch but this may be usable if the call is further tandemed or interflowed to a R6.3 or later switch depending on the tandem switch outgoing trunk group settings. Note that codeset 6/7 IEs are not defined for QSIG and other vendor switch treatment of calls with a LAI IE is undefined (could be ignored, blocked, misinterpreted, etc.).

If both a codeset 6/7 LAI IE and codeset 0 user information is received at an earlier release DEFINITY, the LAI IE will be used by default and the codeset 0 UUI IE, if sent, will be treated as a regular UUI IE for ASAI. The shared UUI IE format is not supported by pre-R6.3 systems or existing pre-R6.3 ASAI applications.

In general, when the trunk group is set to non-QSIG and shared or to QSIG (service-provider or shared), it is recommended that the `Send Codeset 6/7 LAI IE` option should not also be set to **y** due to the overhead of sending duplicate information. In some cases, this configuration could exceed the SETUP message and/or user information byte count limits for the network and result in the user information being dropped. Also, transport could cost more in networks which charge for user transport by quantity of bytes transported. An administration warning message will be given when this combination is set for the trunk group. In fact this combination is not recommended except in very limited cases where a mix of early and later DEFINITY switches can be reached over the same trunk group (via a public or switched private network) using Lookahead Interflow, and where Best Service Routing or UCID is not active or being used and the data that needs to be forwarded with the call can be limited to that supported by the network.

The `Send Codeset 6/7 LAI IE` option must not be set to **y** with trunk groups (or in DEFINITY switches) where calls will be interflowed over public networks or virtual private networks that do not support codeset 6/7 transport. In these cases, the codeset 6/7 IE will not be forwarded or the calls may not be routed by the network (blocked due to protocol errors). This can happen in some international situations, notably over networks in Germany. Note that if a non-QSIG trunk group has both the **service-provider** option set and the `Send Codeset 6/7 LAI IE` option set to **n**, the LAI info will not be forwarded with the call. This could be a valid combination with Look-Ahead Interflow operation if the remote switch has not been upgraded to R6.3 (or later) and the network does not support or blocks call SETUP messages with codeset 6/7 IEs included. In this case, the originating DNIS (VDN Name) display can be emulated by using dedicated VDNs at the remote switch for interflow from different call center locations and call center applications. However, ASAI applications that require forwarding of the `Other LAI Info` (in-queue-time-stamp and/or queue-priority) can not be supported.

Table D-6. Summary of what is included in the SETUP message

		Supplementary Services Protocol	
UUI IE Treatment	**Send Codeset 6/7 LAI IE**	**Non-QSIG (other than b)**	**QSIG (SS b) [1]**
service-provider	n	ASAI provided user info in codeset 0 UUI IE	ASAI provided user info in a codeset 0 UUI IE and all other user info in codeset 0 MSI
	y	ASAI provided user info in codeset 0 UUI IE & a codeset 6/7 LAI IE	ASAI provided user info in codeset 0 UUI IE, all other user info in codeset 0 MSI and a codeset 6/7 LAI IE [3]
shared	n	All user info in a shared codeset 0 UUI IE	All user info in codeset 0 MSI
	y	All user info in a shared codeset 0 UUI IE & a codeset 6/7 LAI IE [2]	All user info as codeset 0 MSI and a codeset 6/7 LAI IE [3]

1. MSI is sent in codeset 0 Facility IEs.
2. With this combination, the LAI information (VDN-Name and Other-LAI) will be sent in both the UUI IE and in the LAI IE (setting the UUI Priorities for these items to “blank” can eliminate the duplication).
3. With this combination, the LAI information (LAI Name and Other LAI) will be sent both as MSI (in a Facility IE) and in the LAI IE. Note that LAI IE and shared MSI operation with other vendor switches is undefined.

Table D-7. When to Use Specific Trunk Group Options

	Trunk Group Option Settings		
	UUI IE Treatment		Send Codeset 6/7 LAI IE
Situation	**Non-QSIG**	**QSIG**	
Trunk groups over which information forwarding is not required (for LAI, BSR or UCID transport).	service-provider	service-provider	n
Non-LAI interflow or tandem calls to service providers or other vendor switches that do not recognize shared UUI.	service-provider	service-provider	n
LAI to pre-R6.3 switches over networks that block codeset 6/7 IE calls.	service-provider	service-provider	n
LAI to pre-R6.3 switches over networks that allow codeset 6/7 (traditional LAI) with or without ASAI applications that use UUI and/or LAI Info	service-provider	service-provider [2]	y
LAI over public/virtual private network to mixed R6.3 and pre-R6.3 switches, where the R6.3 switches use shared information forwarding. The pre-R6.3 switches may use LAI Info in an ASAI application, but must not use UUI.	shared [1]	shared [2]	y
LAI over public/virtual private network to mixed R6.3 and pre-R6.3 switches. The R6.3 and pre-R6.3 switches may use LAI info or UUI in an ASAI application.	service-provider [3]	service-provider [2]	y
BSR and/or LAI to all R6.3 or later switches [4]	shared	shared	n

1. With this combination, the LAI information (LAI Name and Other LAI) will be sent in both the UUI IE and in the LAI IE.
2. With this combination, the LAI information will be sent both as MSI (in Facility IEs) and in the LAI IE.
3. The LAI IE and ASAI non-shared UUI is supported, but BSR, UCID and other new data items are not.
4. All switches interflowed to must be R6.3 or later with “shared” incoming and outgoing trunk group settings. Tandeming/interflowing through R6.3 or later switches requires “shared” settings. Switches tandemed through can be pre-R6.3 (or other vendor switches that pass codeset 0 UUI or MSI transparently). This is the only combination that supports BSR and new data items information forwarding. In this scenario it is recommended to never set "Send Codeset 6/7 LAI IE" to "y" in order to save SETUP message space and to ensure operation over networks that do not allow codeset 6/7 IEs. *This combination is the recommended setup for Multi-Site Routing.*

Troubleshooting Vectors

E

Introduction

This chapter is intended to serve as a troubleshooting guide for Call Vectoring. The first part of the chapter includes two tables that indicate and explain unexpected operations within Call Vectoring that the customer may encounter. The first table focuses on the Call Vectoring features, while the second table focuses on the Call Vectoring commands. The second part of the chapter contains a table that focuses on *converse-on* command debugging. The third part of the chapter contains procedures for tracking many of the unexpected operations within Call Vectoring that are discussed in the two tables. Finally, the fourth part of the chapter contain procedures for tracing call flow and extended wait time.

NOTE:
If EAS is optioned, “skill” replaces “split.”

Unexpected Feature Operations

Table E-1 indicates and explains unexpected operations within Call Vectoring that you may encounter.

NOTE:
For solutions to these unexpected operations, refer to Chapter 3 through Chapter 10 and to Appendix A, Appendix C, and Appendix H in this guide.

Table E-1. Unexpected Feature Operations

Feature/Area	Customer Observations	Causes
General Vector Processing	Vector stuck.	1,000 steps executed (3000 with enhanced LAI).
		No default treatment in the vector.
	Audible feedback lasts longer than the delay interval.	Last vector step.
		Queuing for an announcement.
		Queuing for a touch-tone receiver for a *collect digits* step.

Continued on next page

Table E-1. Unexpected Feature Operations

Feature/Area	Customer Observations	Causes
Look-Ahead Interflow	Agent receiving phantom call.	Agents on both switches become available simultaneously. Avoid by including at the beginning of the receiving switch vector a short *wait-time* or *announcement* step. Also, use the interflow-qpos conditional (see "How Enhanced Look-Ahead Interflow Works" on page 7-9).
	Remote agent receiving phantom calls when vectoring uses qpos conditional.	Interflow-qpos threshold may be set too low.
	No Look-Ahead Interflow attempts accepted.	No trunks. PRI network failure. Insufficient FRL.
	All Look-Ahead Interflow attempts accepted.	Look-Ahead Interflow attempts are interworking off of one of the following: ■ Interworking off of the PRI network ■ Receiving vector not designed for conditional acceptance ■ *route-to with coverage yes* command was used to interflow
	Look-Ahead DNIS name not displayed or no collected digits received	Look-Ahead Interflow not optioned at the receiving switch. LAI IE or VDN Name (Shared UUI) not forwarding with call. Trunk group settings are not administered to support this data. See sections on Information Forwarding.

Unexpected Command Operations

Table E-2 indicates and explains the unexpected operations the customer may encounter in using the Call Vectoring commands.

Table E-2. Unexpected Command Operations

Command Step	Customer Observation(s)	Cause(s)
adjunct routing	Step skipped (that is, default treatment).	Invalid link extension.
		No trunks available.
		COR/FRL restricted.
		Timeout. (Application did not respond within the time specified in the *wait-time* command and/or within the time length of the recorded announcement.)
		Digit string inconsistent with networking translation.
		ASAI link down.
		Invalid route destination returned from adjunct.
	Busy tone.	Busy local destination has no available coverage points.
	Network reorder or intercept.	Digit string supplied by adjunct inconsistent with public network translation.
		Digit string inconsistent with networking translation.
	Intercept or reorder tone heard.	Vector processing succeeded routing off switch, but a problem has occurred before routing to its final destination.
	All trunks busy on a quiet system.	Two switches treating each other as backup switch.
	Step skipped.	Port Network (PN) link down.

Continued on next page

Table E-2. Unexpected Command Operations — *Continued*

Command Step	Customer Observation(s)	Cause(s)
announcement	Announcement not heard.	Announcement board not present.
		Announcement not administered.
		Announcement not recorded.
		Announcement being rerecorded.
		All ports busied out.
		Announcement restore in progress.
		Link to TN750 down.
	Extra delay before hearing announcement.	Announcement queue full.
		All integrated announcement ports busy.
		Analog announcement busy.
	Vector processing stops.	Analog announcement does not answer.
	Listening to silence after announcement.	Announcement is the last step.
	Incomplete announcement.	Agent becomes available.
		Previous *adjunct routing* step succeeds.
busy	Ringback heard instead of busy tone.	Unanswered CO trunk.
check	Call does not enter queue or terminate to agent.	Step condition not met.

Continued on next page

Table E-2. Unexpected Command Operations — *Continued*

Command Step	Customer Observation(s)	Cause(s)
check and queue-to	Call does not enter queue or terminate to agent.	Queue length specified on the hunt group screen has been exceeded.
		Invalid split.
		Split not vector-controlled.
		Already queued to three different splits.
		No queue.
		Queue or check status indicates space when queue is full due to direct agent calls.
		Best keyword is used but consider series is not defining “best” data.
	Call apparently answered in wrong order.	Call being requeued at different priority.
		Call superseded by higher priority call, including direct agent call.
	Call is not routed to remote best location.	No trunk available.

Continued on next page

Table E-2. Unexpected Command Operations — *Continued*

Command Step	Customer Observation(s)	Cause(s)
collect digits	Announcement not heard while waiting for digits, but network billing indicates that the call was answered.	Announcement board not present.
		Announcement not administered.
		Announcement not recorded.
		Announcement being rerecorded.
		All ports busied out.
		Announcement restore in progress.
		Dial ahead digit exists.
	Collect step and announcement skipped.	TTR TN744 not in system.
		Link to PN that has TN744 is down.
		TTR queue full for TN744.

Continued on next page

Table E-2. Unexpected Command Operations — *Continued*

Command Step	Customer Observation(s)	Cause(s)
collect digits (Continued)	Delay before hearing announcement.	All TTR TN744 ports busy, but space in queue.
		Announcement queue full.
		All integrated announcement ports busy.
		Analog announcement busy.
	Vector stuck.	Analog announcement does not answer.
	Dial-ahead digits not recognized.	Dial-ahead digits entered prior to first collection step.
		Call has been transferred.
		LAI attempt has been made.
		TTR has been released.
		24 digits have already been provided.
		Call Prompting timeout since the last digit was entered.
	Vector processing halted at collect step; announcement heard again upon return.	Call put on hold, transferred, or conferenced.
	Insufficient digits collected; call routed to intercept.	Caller dialed # too soon.
		Caller dialed * without reentering correct digits.
		Call Prompting interdigit time-out.
	Caller information button denied.	No digits were collected.
		Display not in Normal mode.
	Collect announcement not heard and first collected digit incorrect.	System does not contain all TN748C Vintage 5 (or later) boards.
	Incomplete announcement.	Agent becomes available.
		First digit dialed.

Table E-2. Unexpected Command Operations — *Continued*

Command Step	Customer Observation(s)	Cause(s)
consider	Local split/skill best (in Primary vector or Status Poll vector)	If split/skill number is correct, split or skill has no agents logged in, no queue slots available, or all agents are in AUX work.
	Remote location is never best	No BSR application plan assigned to Primary VDN. Location number not assigned in application plan. Missing routing number for Status Poll VDN. No vector assigned to Status Poll VDN. Step in Status Poll vector is initializing “best” data before reply-best step.
converse-on split[1]	VRU script not executed.	Queue full. No queue. Invalid split. Split not vector-controlled. VRU down.
	"Ani" digits not passed.	ANI not available.
	"Qpos" digits not passed.	Call not queued to a nonconverse split.
	No data returned from VRU.	No TTRs available on the DEFINITY System.
	VRU script terminated prematurely.	Agent becomes available. VRU script attempted to transfer the call.
	Wait digits not passed	Call not queued or no working agents in splits where call is queued.

Continued on next page

Table E-2. Unexpected Command Operations — *Continued*

Command Step	Customer Observation(s)	Cause(s)
disconnect	Announcement not heard.	Announcement board not present.
		Announcement not administered.
		Announcement not recorded.
		Announcement being rerecorded.
		All ports busied out.
		Announcement restore in progress.
	Extra delay.	Announcement queue full.
		All integrated announcement ports busy.
		All analog announcements busy.
	Vector stuck.	Analog announcement does not answer.
goto step	Branch is not made to the specified step.	Step condition not met.
		System time not set.
goto vector	Branch is not made to the specified vector.	Step condition not met.
	Vector stuck.	Goto vector with no steps or with all failed steps.

Continued on next page

Table E-2. Unexpected Command Operations — *Continued*

Command Step	Customer Observation(s)	Cause(s)
messaging split	Vector stuck (with ringback).	Extension unknown to AUDIX.
	Step skipped, no message left.	AUDIX link down.
		DCS link to remote AUDIX down.
		All DCS trunks busy.
		Queue for AUDIX voice ports is full.
	Vector stuck (with busy).	Remote AUDIX link down.
	Messages not found.	Message extension is *none* (message is left for VDN that accessed the vector).
	Delay before AUDIX answers.	All AUDIX ports busy, but space in queue.
	Busy tone.	Queue for AUDIX voice ports is full.
	Step skipped.	Split not AUDIX split anymore.
reply-best	Status poll VDN/vector not processing any calls	Incoming call not ISDN. No application plan defined for BSR application. Status Poll VDN routing number missing from or wrong in application plan.

Continued on next page

Table E-2. Unexpected Command Operations — *Continued*

Command Step	Customer Observation(s)	Cause(s)
route-to[2]	Step skipped (that is, default treatment).	Invalid local extension.
		No trunks available.
		COR/FRL restricted.
		Digit string inconsistent with networking translation.
		Busy local destination (route to digits without coverage and route to number).
		No digits collected.
		Step condition not met.
	Network reorder.	Digit string inconsistent with public network translation.
	Intercept or reorder tone heard.	Vector processing succeeded routing off switch, but a problem has occurred before routing to its final destination.
	All trunks busy on a quiet system.	Two switches treating each other as a backup switch.
stop	Call dropped.	Call not queued when vector processing stops.

Continued on next page

Table E-2. Unexpected Command Operations — *Continued*

Command Step	Customer Observation(s)	Cause(s)
wait-time	Audible feedback longer than delay interval.	Queuing for an announcement or for a TTR.
		Stop command executed.
	Audible feedback shorter than delay interval.	Previous *adjunct routing* step succeeds.
		Agent becomes available.
	Music not heard.	No music port administered.
		Music source disconnected or turned off.
	Alternate audio/music source not heard	Announcement board not present.
		Audio/Music source not administered.
		Audio/Music source not recorded.
		Audio/Music source being rerecorded.
		All ports busied out.
		Announcement restore in progress.

1. Refer to the "Converse Command Debugging" section later in this appendix for more details on *converse-on* command debugging
2. Complete operation details for the *route to* commands are presented in Appendix H.

Converse Command Debugging

Table E-3 is intended to help your troubleshooting efforts with the *converse-on* command.

NOTE:
Refer to Appendix I for details on the call flow for converse-VRI calls.

Table E-3. Converse Command Debugging

SYMPTOM	CAUSES	EVIDENCE
PLACING A CALL:		
Converse step skipped.	VRU down (RONA).	Vector event.
	Split queue full	Vector event.
Call stuck in converse.	VRU port doesn't answer, RONA not used.	Check split administration.
	VRU down, RONA leaves call in queue.	Check split status.
DATA PASSING:		
First set of digits not collected.	Converse first delay too short.	Check administration.
	No ANI available.	Vector event.
	No digits collected.	Vector event.
	Call not queued (qpos).	Vector event.
	Expected wait time not available	Vector event.
	VRU timed out awaiting first digit.	VRU error log/trace.
	VRU first digit timeout too short.	Check VRU script.
		Check converse first data delay.
	Faulty hardware.	Diagnostics
Second set of digits not collected.	VRU digit count on first prompt in VRU script does not include "#."	Check VRU script.
	Converse second delay too short.	Check administration.

Continued on next page

Table E-3. Converse Command Debugging — *Continued*

SYMPTOM	CAUSES	EVIDENCE
DATA PASSING: (Continued)	No ANI available.	Vector event.
	No digits collected.	Vector event.
	Call not queued (qpos).	Vector event.
	Expected wait time not available because call is not queued or the splits/skills that the call is queued to are not staffed	Vector Event
	VRU timed out awaiting first digit.	
	VRU error log/trace.	
	VRU first digit timeout too short.	Check VRU script. Check converse second data delay.
	Inter-digit timeout too short on first prompt and collect.	Check VRU script.
	Faulty hardware.	Diagnostics.
Digits incomplete.	Converse data delay too short.	Check administration.
	Faulty hardware.	Diagnostics.
Second set of digits is the same as the first digits passed.	VRUs first prompt timed out.	Check administration.
	Faulty hardware.	Diagnostics.
DATA RETURN:		
No digits returned to the DEFINITY System.	Flash not recognized by switch.	VRU error log/trace.
		Check flash timing on VRU.
	Converse data return FAC not administered.	Check administration.
	VRU does not return FAC.	VRU script. Transfer attempt vector event.
	VRU returns incorrect FAC.	VRU script. Transfer attempt vector event.

Continued on next page

Table E-3. Converse Command Debugging — *Continued*

SYMPTOM	CAUSES	EVIDENCE
	Digit timeout during FAC.	Transfer attempt event.
	Converse data return FAC overlaps with other entries in the dial plan	Check dial plan.
	Faulty hardware.	Diagnostics.
Not all digits returned to the DEFINITY System.	Digit timeout after FAC.	None unless VRU logs being dropped by the DEFINITY System.
	Overflow of Call Prompting buffer	Vector Event.
	Faulty hardware.	Diagnostics.
Collect announcement not heard.	Too many digits returned by VRU.	Check VRU script.
	Faulty hardware.	Diagnostics.

Tracking Unexpected Vector Events

If you have an SAT terminal, you can display unexpected vector events and when you have corrected each problem, then you can clear events from the error log. A vector event is an error that results from resource exhaustion or from faulty vector programming, rather than from a DEFINITY System software error. For example, failures involving the *route-to* command are usually due to an invalid extension entered by the user.

By displaying vector events, you can do the following:

- Diagnose and correct each Call Vectoring problem, as indicated by its corresponding vector event, and thereby
- Eliminate the need for a technician to make on-site visits to do the same.

The following sections explain how you can troubleshoot by tracking unexpected vector events.

Display Events Form

The first step is to initiate the display of vector events. You do this by entering the *display events* command at the *enter command* prompt.

Once the command is entered, the Display Events form appears on the screen as in Screen E-1.

```
display events                                          Page 1 of 1    SPE B
                          EVENT REPORT
        The following option control which events will be displayed.
           EVENT CATEGORY
                        Category:  Vector
           REPORT PERIOD
                      Interval: _a_   From: __/__/__:__  To: __/__/__:__
           SEARCH OPTIONS
                  Vector Number: ___
                     Event Type: ____
```

Screen E-1. Layout of Display Events Form

The following list indicates the options on the form, comments on these options, and also discusses the field(s) within each option.

- **EVENT CATEGORY**. This option is intended to indicate the class of logged events to be displayed. For our purposes, the default value *Vector* automatically appears in this display-only *Category* field. The value *Vector* indicates that only vector events will be displayed.
- **REPORT PERIOD**. This option allows you to specify a report period. This period consists of an *Interval* field, a *From* date/time stamp, and a *To* date/time stamp. Valid entries for the *Interval* field include *(h)our*, *(d)ay*, *(w)eek*, and *(a)ll*. Both stamps consist of a series of numbers that represent a period of time, as follows: *1 through 12* (month), *1 through 31* (day), *0 through 23* (hour), *0 through 59* (minutes). If the field and stamps are populated, only the vector events that occurred within report period specified are displayed. Otherwise, all vector events are displayed regardless of when they occurred.
- **SEARCH OPTIONS**. This option contains two fields, *Vector Number* and *Event Type*.

  Vector Number allows you to specify a vector number. If this field is populated, only vector events that are associated with this vector number are displayed. Otherwise, all vector events are displayed regardless of the vector number with which they are associated.

  Event Type allows you to specify the number associated with a particular type of vector event. This number may range from *0* to *999*. If the *Event Type* field is populated, only vector events of the type indicated are displayed. Otherwise, all vector events are displayed regardless of type.

Display Events Report

After you complete the Display Events form, you can generate the Display Events Report by submitting the display request and hitting the Enter key a second time. A sample report appears in Screen E-2.

```
                                    EVENTS REPORT
Event  Event                Event    Event    First        Last         Event
Type   Description           Data 1    Data 2   Occur        Occur        Cnt
  20   Call not queued       12/5     B        09/28/13:43  09/28/13:43   21
 541   Not a messaging split  Split 89  4C      09/28/13:43  09/28/13:43  136
```

Screen E-2. Display Events Report

The Display Events Report provides details of all the logged vector events that meet the selection criteria supplied by the user. The G3R displays upto 100 vector events and the G3si 50. The following list identifies and discusses the fields in the report.

- **Event Type** contains a unique number between *0* and *999* that identifies the type of vector event that occurred.
- **Event Description** contains text that describes the vector event.
- **Event Data 1** is a 9-character field that contains data in one of two formats:
  - *<number1>/<number2>* (for example, *12/5*), where *<number1>* is the vector number associated with the vector event, and where *<number2>* is the step number associated with the vector event. This format is used for events to which an event type in the range of 0 through 499 is assigned.
  - *Split<number>* (for example, *Split 89*), where *<number>* is the split associated number associated with the vector event. This format is used for events to which an event type in the range of 500 through 999 is assigned.
- **Event Data 2** is an 8-character field that contains additional data encoded as a hex number (for example, *4C*). This number serves as a call identifier. If two or more events with an identical identifier occur at about the same time, it can be concluded that the events were caused by the same call.
- **First Occur** is an 11-character field that contains the date and time when the vector event first occurred (for example, *09/28/13:43*).
- **Last Occur** is an 11-character field that contains the date and time when the vector event last occurred (for example, *09/28/13:48*).
- **Evnt Cnt** (Event Count) contains a number ranging from *1* to *255* that indicates the total number of vector events of this type that have occurred.

Summary of Vector Events

This section contains Table E-4 that does the following:

- Lists the number of each vector event
- Provides a description and an explanation (and sometimes possible causes and solutions) for each event type.

Table E-4. Summary of Vector Events

Event Type	Event Description	Event Explanation
1	Call dropped; call not queued at *stop* step.	Vector processing ended without the call being queued to a split and, as a result, the call cannot be answered. This implies that some default condition was not programmed or that the vector was designed to not always answer the call. Also, call was subsequently dropped.
2	Vector with no steps	The call encountered a vector with no steps administered.
3	1,000 step executed	This can occur due to the following: Incorrect vector programming (for example, including a series of *goto* steps that point to one another) Excessive repetition of a programmed loop during a single call (for example, recurring announcement-wait loop)
4	Administration change	The administration of this step occurred while the step was being executed. The call flow for this call is unpredictable. Vectors should not be changed while calls are active.
5	Call dropped by vector disconnect timer	The call was still in vector processing when the vector disconnect timer expired. The call dropped.
10	Retrying announcement	During an *announcement* step, a *collect digits* step that contains an announcement, or a *disconnect* step, the announcement was not available, and the announcement queue (if specified) was full. The step is retried at regular intervals.

Continued on next page

Table E-4. Summary of Vector Events — *Continued*

Event Type	Event Description	Event Explanation
11	No announcement available	During an *announcement* step, a *collect digits* step that contains an announcement, or a *disconnect* step, the announcement was not available for one of the following reasons: Announcement was not recorded Analog announcement was busied out Integrated announcement board was not installed Integrated announcement ports were busied out Integrated announcement was being recorded or restored
20	Call cannot be queued	A *queue-to split, messaging split,* or *check split* command failed to queue the call. **NOTE:** Event types *520, 521, 522* and *541* may be observed for the same call at the same time.
21	Queued to three splits	The call attempted to queue to four splits. Multiple split queuing allows the call to queue to a maximum of three splits simultaneously. If the call queued to one or more splits, and if it should now be dequeued from those splits and then queued elsewhere, one solution is to route the call to a station (which may be administered without hardware). Once this happens, the call is forwarded to the VDN that controls the next stage of the call.
30	No TTR available	A *collect digits* command failed because TN744 port was not available All queue slots were occupied
31	Dial-ahead discarded	Previously entered dial-ahead digits have been discarded via access of a(n) *adjunct routing, converse-on, route-to number,* or *messaging split* step.

Continued on next page

Table E-4. Summary of Vector Events — *Continued*

Event Type	Event Description	Event Explanation
32	Prompting buffer overflow	The prompting digit buffer already contained the maximum of 24 digits when additional dial-ahead digits were entered by the caller. These additional digits are not stored.
33	ced digits left behind	A *collect ced digits* step collected digits from a UEC IE, and more than 16 digits were sent from the network.
34	cdpd digits left behind	A *collect cdpd digits* step collected digits from a UEC IE, and more than 16 digits were sent from the network
35	ced digits not available	A *collect ced digits* step collected digits from a UEC IE, and no digits were sent from the network, or no digits were present in the UEC IE.
36	cdpd digits not available	A *collect cdpd digits* step collected digits from a UEC IE, and no digits were sent from the network, or no digits were present in the UEC IE.
40	Messaging step failed	A messaging step failed because the Messaging Adjunct was not available. **NOTE:** Event types 540 and 541 may be observed for the same call at the same time.
50	Route -to step failed	A *route-to* step failed to reach the intended destination. **NOTE:** Event types 51 and 52 may provide more specific information regarding the reason for the failure. See Appendix H, ‘‘Operation Details for the Route-to Command’’.

Continued on next page

Table E-4. Summary of Vector Events — *Continued*

Event Type	Event Description	Event Explanation
51	No digits to route-to	The *route-to digits* step was unable to route the call because the previous *collect digits* step failed to collect any digits. This could result from an error in vector programming (for example, a *route-to digits* step appears without a preceding *collect digits* step). More often, however, this results because the caller was unable to enter the required digits (that is, the caller was using a rotary telephone), or because the caller was not provided with enough information to do so (as can be the case for auto-attendant applications).
52	No available trunks	A *route-to* command was unable to reach the specified off-switch destination due to a lack of available trunks.
53	Route-to step failed	The step was unable to seize a trunk because of a hardware problem or glare.
54	LAI retry	Look Ahead Interflow *route-to* step failed because of glare. The route will be retried once.
55	Double coverage attempt	Coverage option on route-to step was ignored because double coverage is not allowed. This may happen when the call has covered to a VDN.
60	Adjunct route failed	An adjunct route failed for one of reasons indicated in event types 61 through 66.
61	Invalid destination	The *adjunct routing* command returned digits that did not represent a valid destination.
62	Adjunct route cancelled	The *adjunct routing* step was cancelled because another "routing" step (such as a *queue-to split* step) was encountered in the vector.
63	Queue before route	The *adjunct routing* command was skipped because the call had already been queued via a *queue-to split* or a *check split* command.

Continued on next page

Table E-4. Summary of Vector Events — *Continued*

Event Type	Event Description	Event Explanation
64	Adjunct link error	The *adjunct routing* command was cancelled for one of the following reasons: Link to the adjunct was down ASAI protocol violation prevented the call from completing Software resources to complete the call were unavailable
65	Agent not logged in	A Direct Agent Call was made to an agent who was not logged into the relevant split. Used for adjunct routing request only.
66	Agent not member of split	A Direct Agent Call was made to an agent who is not a member of the relevant split. Used for adjunct routing request only.
67	Invalid direct agent	A Direct Agent Call was made to an agent extension that is not valid. Used for adjunct routing request only.
70	Busy step for CO trunk	A CO trunk call reached a *busy* step in a vector without having previously received answer supervision. As a result, the caller continues to hear ringback rather than the busy tone.
80	Time not set	A *goto* step with a *time-of-day* conditional was processed, but the switch time was not set.
81	No digits collected	No digits were collected and a comparison was requested against a digit string or “in-table”. The comparison test was considered false and the next step in the vector was executed.
90	Wait step music failed	A *wait-time* step with music was accessed, but the music was not connected. Music may not be administered correctly.
91	Wait step ringback failed	A *wait-time* step with ringback was accessed, but the ringback was not connected.
100	Redirect unanswered call	The call was sent to an agent via a vector, but, due to the Redirection on No Answer (RONA) feature, the call was redirected from the ringing agent.

Continued on next page

Table E-4. Summary of Vector Events — *Continued*

Event Type	Event Description	Event Explanation
101	Redirect of call failed	The call was sent to an agent via a vector, but, due to the Redirection on No Answer (RONA) feature, the call was redirected from the ringing agent. The call could not be redirected.
110	Converse no ANI digits	On a *converse-on* step with passing type *ani*, no information was available to populate the field.
111	Converse no qpos digits	On a *converse-on* step with passing type *qpos*, no information was available to populate the field.
112	Converse no prompt digits	On a *converse-on* step with passing type *digits*, no information was available to populate the field.
113	Converse drop during data	On a *converse-on* step, the converse agent hung up while data was being passed. This may indicate a port failure.
115	ASAI transfer converse	ASAI attempted a transfer of a call that was active at a *converse* step. The transfer failed, and vector processing continued at the next vector step.
116	Converse transfer denied	A transfer of a call that was active at a *converse-on* step was attempted. The transfer either failed or was denied, and vector processing continued at the next vector step.
117	Agent drops converse	While active on a *converse-on* step, an agent became available in a split associated with a *queue-to split* or *check split* step. The call was delivered to the nonconverse agent, and the converse agent was dropped.
125	Data return no digits	On a *converse-on* step, the converse agent activated data return but did not return any digits.
126	Data return timeout	On a *converse-on* step, the converse agent activated data return but timed out while waiting to return digits. Vector processing continued at the next vector step.

Continued on next page

Table E-4. Summary of Vector Events — *Continued*

Event Type	Event Description	Event Explanation
140	Coverage conference denied	Coverage to a VDN in a coverage path was denied because more than one party was active on the call.
150	Invalid EAS hunt group used in the vector step	Either the skill hunt group was removed or the skill hunt group became a non-ACD hunt group.
151	Skill indirection used improperly	Either no VDN skills are administered or the vector command has skill indirection and EAS is not enabled.
160	No vector steps, ANI sent	ANI was sent to CMS for a call that reached a VDN that accessed a vector with no steps defined.
170	ASA - invalid VDN	A *check* or *goto* test requested a comparison of ASA for a VDN that had been removed since the vector was programmed. The comparison test was considered false and the next step in the vector was executed.
200	ANI not avail - digits	A *goto* test requested a comparison of ANI against a digit string and ANI was not available for the call. The comparison test was considered false and the next step in the vector was executed.
210	Routing table not assigned	A *goto* test requested a comparison with a vector routing table that is not assigned or had been removed since the vector was programmed. The comparison test was considered false and the next step in the vector was executed.
211	No entries in routing table	A *goto* test requested a comparison with a vector routing table that has no entries. This is considered as a non-match.
212	ANI not avail - table	A *goto* test requested a comparison of ANI against “in-table” and ANI was not available for the call. The comparison test was considered false and the next step in the vector was executed.

Continued on next page

Table E-4. Summary of Vector Events — *Continued*

Event Type	Event Description	Event Explanation
220	EWT call not queued	A *goto* test for a call or converse data passing requested EWT for a call not in queue. In this case, the wait time was assumed to be infinite and the comparison was based on EWT > largest possible threshold.
221	EWT not sent to VRU	The EWT "wait" time for the call was not sent to the VRU for a *converse-on passing wait* vector step because the call was not queued or the splits/skills that the call was queued to were unstaffed.
222	System clock change	The system clock was changed, therefore any calculations involving time (i.e., ASA and EWT) will be inaccurate.
230	II-digits not avail - digits	A *goto* test requested a comparison of II-digits against a digit string and II-digits were not available for the call. The comparison test was considered false and the next step in the vector was executed.
231	II-digits not avail - table	A *goto* test requested a comparison if II-digits against "in-table" and II-digits were not available for the call. The comparison test was considered false and the next step in the vector was executed.
240	No agent strategy found in VDN	The active VDN for the call, as determined by VDN override, did not have a BSR Available Agent Strategy.
251	Call is not incoming ISDN	Occurs when a *reply-best* command in a status poll vector receives and tries to process a non-ISDN call. Processing in the status poll vector terminated is without a reply being sent.
261	No "best" location found	A *queue-to best, check-best,* or *reply-best* command failed because the call vector was unable to calculate a "best" value or because no local best existed. Vector processing continues at the next step. Vectors in multi-site BSR applications won't attempt to interflow calls in this situation.

Continued on next page

Table E-4. Summary of Vector Events — *Continued*

Event Type	Event Description	Event Explanation
262	Look-Ahead Interflow attempt failed	Interflow of the call failed: no trunk was available, LAI denial, or some other problem. Vector processing continues at the next step. In BSR applications, polling of this resource is temporarily suppressed.
271	No BSR app num in VDN	A *queue-to best, check-best,* or *consider location* command failed because the active VDN for the call as determined by VDN override has no BSR application number assigned. Processing continues with the next vector step. Only occurs in multi-site BSR applications.
272	No BSR appl plan admin'd	A *queue-to best, check best,* or *consider location* command failed because the application number assigned to the active VDN does not have an application plan assigned. Processing continues at the next step.
273	Location not on BSR form	A *consider* command failed because it refers to a location number that is not in the BSR Application form assigned to the active VDN. Vector processing continues at the next step.
274	Status Poll VDN field is blank	A *consider* command failed because the entry for this location on the BSR Application form does not contain a routing number for the status poll VDN.
275	Interflow VDN field is blank	A *queue-to best* or *check-best* command failed because the entry on the BSR Application form for the relevant location does not contain a routing number for the interflow VDN.
276	Agent Status Info Invalid	A *consider location* command failed because the status poll returned invalid data for an available agent (AIT, skill level, or occupancy is missing or out of range). Vector processing continues at the next step. Polling of this location is temporarily suppressed.
277	BSR Status Info Invalid	A *consider location* command failed because the status poll returned invalid EWT data. Vector processing continues at the next step. Polling of this location is temporarily suppressed.

Continued on next page

Table E-4. Summary of Vector Events — *Continued*

Event Type	Event Description	Event Explanation
278	No BSR Data in Response	A *consider location* command failed because the status poll did not return data in the DISCONNECT message. Vector processing continues at the next step. Polling of this location is temporarily suppressed.
279	No response from status poll	A *consider location* command failed because the status poll did not respond within the time allowed or because the status poll could not be performed. Vector processing continues at the next step. Polling of this location is temporarily suppressed.
280	Bad resp from status poll	A *consider location* command failed because it received an invalid response from the status poll such as an LAI acceptance message (such as ALERT or CONNECT). Vector processing continues at the next step. Polling of this location is temporarily suppressed.
281	BSR EWT is infinite	A *consider* command failed because the EWT for the referenced split or skill is infinite. This may be because all agents are logged out or in AUX work, or because no queue slots are available. Vector processing continues at the next step. Polling of this location is temporarily suppressed.
282	BSR status poll attempt failed	A *consider location* command failed because the status poll attempt failed. See other events for specific reason. Vector processing continues at the next step. Polling of this location is temporarily suppressed.
283	BSR poll no trunks	A *consider location* command failed because there were no available trunks. Vector processing continues at the next step. Polling of this location is temporarily suppressed.
284	BSR poll seize fail	A *consider location* command failed because the status poll was unable to connect to a trunk due to a hardware problem. Vector processing continues at the next step. Polling of this location is temporarily suppressed.

Continued on next page

Table E-4. Summary of Vector Events — *Continued*

Event Type	Event Description	Event Explanation
285	BSR poll glare retry	The first status poll attempt for a *consider location* command was unable to connect to a trunk due to a race condition (the same trunk being seized for the outgoing call had an incoming call from the remote end). This status poll will be attempted once more. A second attempt failure will result in event 282.
291	No AITCI storage left	The network does not support the transport of all user data, so some user data was not sent. You can prioritize the user data using the Shared UUI Feature Priorities page of the ISDN Trunk Form. For more information, see Chapter 9, ''Information Forwarding''.
292	Data dropped by other app	
293	No room for reply-best information	The network or shared trunk setting does not support the transport of all data for the best resource. This is unlikely under normal circumstances since only 12 bytes of user information are required. Also see event 298.
294	No room for in-VDN time	The network does not support the transport of all user data. You can prioritize the user data using the Shared UUI Feature Priorities page of the ISDN Trunk Form. For more information, see Chapter 9, ''Information Forwarding''.
295	No room for collected dgt	
296	No room for VDN Name	
297	No room for Other LAI	
298	Reply-best got bumped	The network or shared trunk setting does not support does not support the transport of all data about the best resource. (No other applications share user data included in a DISCONNECT message as of R6.3.)
299	In-VDN time got bumped	The network does not support the transport of all user data. You can prioritize the user data using the Shared UUI Feature Priorities page of the ISDN Trunk Form. For more information, see Chapter 9, ''Information Forwarding''.
300	Collected dgts got bumped	
301	VDN Name got bumped	
302	Other LAI got bumped	
303	Block: send reply-best	The transport of the best data for a *reply-best* command was denied because the trunk group is neither Supplementary Service b or Shared UUI.

Continued on next page

Table E-4. Summary of Vector Events — *Continued*

Event Type	Event Description	Event Explanation
304	No enhanced info is sent	During the execution of a *queue-to best* or *check best* step, information forwarding transport over this trunk was denied because the trunk group is neither Supplementary Service b nor Shared UUI. This event is not logged for LAI (for example, in execution of a *route-to* step) in order to permit backward compatibility. For more information, see ''Unexpected Feature Operations'' on page E-1 as well as Chapter 9, ''Information Forwarding'' and Chapter D, ''Advanced Multi-Site Routing''.
520	Split queue is full	A *queue-to split*, *check split*, or *messaging split* command was executed, but the call did not queue to the split because the queue (if administered) was full. To prevent this condition, use a *goto step...if calls queued in split...>...* before each *queue-to split* or *check split* step so that an alternative treatment may be provided for these cases.
521	Not vector-controlled	The split accessed by a *queue-to split* or *check split* command is not vector-controlled. As a result, the step is skipped.
522	AAS split cannot queue	A *queue-to split*, *check split*, or *messaging split* command was executed on an auto-available split (AAS), but the call did not queue to the split because all the agents were logged out by Redirection on No Answer (RONA).
540	AUDIX link down	AUDIX could not be accessed via a *messaging split* command, because the AUDIX link was down. As a result, the step is skipped.
541	Not a messaging split	The split administered for the *messaging split* command is not a messaging split (that is, it does not have a messaging type administered). As a result, the step is skipped.
542	Can't connect idle agent	The call at the head of the queue can't be connected to an idle agent.

Continued on next page

Table E-4. Summary of Vector Events — *Continued*

Event Type	Event Description	Event Explanation
550	ASA - No staffed agents	A *check* or *goto* test requested a comparison of ASA for a split/skill that has no staffed agents. The comparison was based on ASA > largest possible threshold.
560	EWT no history for split	A *goto* test requested EWT for a split/skill that has not yet acquired history. The wait time in this case is assumed to be the default value.
561	EWT no split queue	A *goto* test requested EWT for a split/skill that has no queue. The wait time is assumed to be infinite. The comparison was based on EWT > largest possible threshold.
562	EWT split queue full	A *goto* test requested EWT for a split/skill whose queue is currently full. The wait time is assumed to be infinite. The comparison was based on EWT > largest possible threshold.
563	EWT split no working agents	A *goto* test requested EWT for a split/skill that has no agents logged in or all logged in agents are in the AUX work mode. The wait time in this case is assumed to be infinite and the comparison was based on EWT > largest possible threshold.
564	EWT split locked	A *goto* test requested EWT for a split/skill that is currently locked. The wait time is assumed to be infinite. The comparison was based on EWT > largest possible threshold.
565	EWT call no working agents	A *goto* test for a call or converse data passing “wait” requested EWT for a call that is queued only to splits/skills that have no agents logged in or that have all logged in agents in AUX work mode. In this case, the wait time was assumed to be infinite and the comparison was based on EWT > largest possible threshold.

Clearing Events

When you have finished your review of the vector event log, you can remove vector and denial events from the error log. You must be a superuser to clear events.

To clear vector and denial events from the error log, type **clear events** at the command prompt and press RETURN. This command clears all vector and denial events from the vector-event buffer space within the error log. It does not delete any other entries in the error log.

Tracing Call Flow

To verify whether your vectoring system is working as you intended, you can observe the operation of both vectors and call vectoring features such as Best Service Routing by using the following commands:

- **list trace vdn <*vdn extension*> [print]**

  This command starts a trace with the next call that arrives at the specified VDN. The trace will run for 15 minutes, or until you cancel the trace, or until the call leaves vector processing, whichever comes first. This command will trace a call through multiple vectors.

- **list trace vec <*vector number*> [print]**

  This command starts a trace with the next call that arrives at the specified vector. The trace will run for 15 minutes, or until you cancel the trace, or until the call is routed to another VDN or vector or leaves vector processing, whichever comes first. This command will not trace a call through multiple vectors.

- **list trace previous [print]**

  This command displays or prints the data from the previous trace. It does not initiate a new trace.

These commands display or print a real-time list of vector processing events for a single call. Vector commands are displayed as they are executed. For BSR commands, the switch also displays relevant data:

- status information, such as *consider location 110 polling suppressed*
- common vector events, such as *consider location 110 NO DATA RETURNED FROM POLL*
- data specified in the vector step such as estimated wait time (EWT) and adjusted EWT, agent occupancy or idle time, or agent skill level in a considered resource

Tracing Expected Wait Time

To display information about events that have changed the expected wail time, use the following command:

list trace ewt low/high/top/medium <*split number*> [print]

This command starts a trace with the next call that arrives for the specified skill and will run until canceled. This command displays or prints a real-time list of processing events for all calls until the command is canceled.

Functional Differences for G2 and DEFINITY ECS Call Vectoring and EAS

F

Introduction

This appendix provides the Call Vectoring functional differences between the Generic 2 and DEFINITY ECS switches. This information should prove helpful to system administrators who administer networks that utilize both switches.

The appendix is organized into three sections. The first section indicates differences between the Call Vectoring commands common to both the Generic 2 and DEFINITY ECS switches. The second section consists of general Generic 2 and DEFINITY ECS differences within Call Vectoring. Finally, the third section indicates differences between the two switches in defining and/or interpreting split flows.

NOTE:
Call Prompting is not supported on G2.

Differences in Command Function

The following sections indicate the differences between the Generic 2 and DEFINITY ECS system versions of the Call Vectoring commands. The commands discussed include the following:

- *queue-to split*
- *check split*
- *goto step*
- *goto vector*
- *route-to number*

- *announcement*
- *wait-time*
- *busy*

Queue-to Split and Check Split

The *queue-to split* command queues the call to the specified split and assigns a queuing priority level.

The *check split* command checks the status of a split for possible termination of the call to that split. When termination is not possible, queuing at the specified priority is attempted. Termination and/or queuing is attempted if the split meets certain conditions that are specified as part of the command.

Table F-1. G2/DEFINITY ECS Differences for Queuing Commands

DEFINITY ECS	GENERIC 2
The call is simultaneously queued to a maximum of three different splits. The indicated split is checked only once, and if the specified condition is met, an attempt to terminate or queue the call is made. Multiple checking of a backup split requires repeating the *check split* command multiple times and/or unconditional *goto step* looping. After the call is queued to three splits, subsequent queue commands in the vector for additional splits fail and are skipped (unless these commands specify a different priority).	The call is queued to one split at a time. Successful queue commands that occur after the call is already queued cause the call to be dequeued from the first split and queued to a new split. Each *check* step executed by vector processing is rechecked in the background every two seconds while the steps that follow are processed. This process continues until the specified conditions are met. The periodic threshold checking of the *check split* commands is implemented to simulate multiple split queuing.
Calls can be queued to vector-controlled splits via Call Vectoring or to ACD splits/hunt groups directly via hunt group/split extensions when vectoring/prompting is active. Vector-controlled splits can be directly accessed via split extensions or via *route-to* commands to the extension ACD splits/hunt groups can also be accessed via *route-to* commands to the extension.	When Call Vectoring is active, calls can be queued to ACD splits only via the *queue to main split* and *check split* Call Vectoring commands.
Calls cannot be queued to splits that already hold the number of queued calls defined by the split queue size on the hunt group screen. Therefore, every queuing command should be preceded by a check step to determine if the queue is full. Also, queue limits should be set as high as possible to ensure the call queues. See Appendix O, "Call Vectoring System Parameters" for a list of queue limits.	No split queuing capacity limits are in effect, and the commands are never skipped.
The *check split* command can test a maximum threshold of 199 calls in vs, csi, and si switches, and 999 calls in r models.	The command can test a maximum threshold of only 99 calls.
The oldest-call-waiting test condition within the *check split* command has a range of 1 through 999 seconds in one-second increments.	The oldest-call-waiting test condition within the command has a range of 0 through 999 seconds in one-second increments.

Continued on next page

Table F-1. G2/DEFINITY ECS Differences for Queuing Commands — *Continued*

DEFINITY ECS	GENERIC 2
An unconditional *check split* command is allowed.	The *check split* command is conditional only.
The rolling-asa, expected-wait, and wait-improved conditions are available with the *check split* command.	These capabilities are not provided.
Queue-to and *check* commands can queue a call to the best resource as determined by a series of *consider* steps.	These capabilities are not provided.

Goto Step and Goto Vector

The *goto step* command allows conditional or unconditional movement (branching) to a preceding or subsequent step in the vector.

The *goto vector* command allows conditional or unconditional movement (branching) to another vector.

Table F-2. G2/DEFINITY ECS Differences for Goto Commands

DEFINITY ECS	GENERIC 2
The commands can test a maximum threshold of 200 calls in vs, csi, and si switches. In r models, the maximum threshold is 999.	The commands can test a maximum threshold of *only* 99 calls.
The oldest-call-waiting test condition within the commands contains a range of 0 through 999 seconds and is checked according to a 1-second increment.	The oldest-call-waiting test condition within the commands contains a range of 0 through 999 seconds and is checked according to a 1-second increment.
The rolling-asa, expected-wait, counted-calls, ani, ii-digits, and interflow-qpos conditions are available with the *goto* commands. Vector routing tables can be checked for the digits, ani and ii-digits conditions.	These capabilities are not provided.
The *goto...if expected-wait* commands can use the *best* keyword and *wait-improved* condition.	These capabilities are not provided.

Route-to Number

This command routes the call to a specific number.

Table F-3. G2/DEFINITY ECS Differences for Route-to Number Command

DEFINITY ECS	GENERIC 2
The actual digit string is used as the destination. The string can contain special characters that may be stored in an AD string, including *~p, ~w, ~W, ~m,* and *~s* (but not * or #). (See the *route-to number* command in the manual pages of Appendix A.) Feature access codes (AAR/ARS) or trunk access codes may be used to route calls externally.	The AD member number is used as the destination. None of the special characters may be used. The special functions are handled by the AAR/ARS pattern routing.
The trunk may be accessed via ARS/AAR, TAC, or UDP.	AAR/ARS is required for non-DCS trunk calls.
Routing to individual attendant extensions is permitted.	The individual attendant extension feature is not available.
Routing to announcement extensions is permitted.	The announcement command is required for all announcement access.
If the command fails, and if the command is the last step in the vector, the command is not retried. If retrying is required, an unconditional *goto step* can be used to loop back to the *route-to* step.	If the command is the last step in the vector, a busy destination targeted by the command is retried every two seconds.
The command with the *interflow-qpos* condition tests the call for interflow eligibility.	This capability is not provided.
Routing to an ACD split extension is allowed even if Call Vectoring is operational.	This capability is not provided.
Routing to a Service Observing FAC is allowed	This capability is not provided.

Announcement

This command indicates that the caller should expect to hear an announcement. Although the announcement strategy G2 differs from that in DEFINITY ECS, each strategy assures that, theoretically, the entire announcement is played from the beginning.

Table F-4. G2/DEFINITY ECS Differences for Announcement Command

DEFINITY ECS	GENERIC 2
Announcement extensions are used.	Announcement numbers are used.
Provides integrated board internal announcements.	Integrated announcement board is not supported.
The system supports auxiliary trunk-connected external announcement devices.	Supports only auxiliary trunk-connected announcement devices.
The maximum number of calls that can be queued and connected to an announcement is limited by preassigned queue slots. See Appendix O, "Call Vectoring System Parameters" for capacities The system allows for multiple integrated announcement boards. R5si can have up to 5 boards, R5r can have up to 10. The queue slot and maximum connected calls capacities have been increased. See the *DEFINITY Enterprise Communications Server Administration and Feature Description,* 555-230-522, for details	Limited only by the number of time slots available on the module to which the announcement channel is connected. The maximum number of time slots is 256.

Wait-time

This command sets a length of time for a call to wait in the queue. The command also specifies one of the following treatments while the call advances in the queue(s):

- Silence
- Ringing
- Music
- I-silent
- Alternate Audio/Music Source)

Table F-5. G2/DEFINITY ECS Differences for Wait Command

DEFINITY ECS	GENERIC 2
The system-wide music-on-hold feature must be active for music treatment on the command. An alternate audio/music source can be administered for a *wait-time* step.	A separate music option is available for Call Vectoring.

Busy

This command terminates vector processing and gives the caller a busy signal.

Table F-6. G2/DEFINITY ECS Differences for Busy Command

DEFINITY ECS	GENERIC 2
A timeout after 45 seconds is provided.	A 20 second timeout is provided for both CO and non-CO trunks.

General Call Vectoring Functional Differences

Table F-7 provides an overview of general differences for Call Vectoring operations between the Generic 2 and DEFINITY ECS switches.

Table F-7. General Call Vectoring Functional Differences

TOPIC	DEFINITY ECS	GENERIC 2
General ACD	Split queue size is administered on a per split basis with a system-wide maximum of calls. Call queue space for the appropriate maximum number of calls must be distributed on a preassigned basis over all assigned hunt groups and (vector-controlled or nonvector-controlled) ACD splits.	There is no limit to the size of individual split queues.
	An agent may be concurrently logged into three splits at a time.	An agent may be logged into only one split at a time.
	The agent hears the same zip tone signal for calls that are queued to the main split as well as for intraflowed/interflowed calls.	One burst zip tone is provided for calls that are queued to the main split. Two burst zip tones are provided for intraflowed calls (via the *check split* command), and three burst zip tones are provided for interflowed calls (via Look-Ahead Interflow).
ACD Split Strategy	A split or a hunt group can be accessed by either a call vector or a group extension. This allows for both vector calls and nonvector calls in a single split's queue.	When Call Vectoring is optioned, splits do not have extensions. All access to splits must go through a Call Vector via *queue to main split* or *check split* commands.
	Non-vector-controlled splits can specify redirection treatment (such as Call Coverage, Call Forwarding, etc.) and announcement treatment.	Only vector-controlled splits are available when Call Vectoring is active.

Continued on next page

Table F-7. General Call Vectoring Functional Differences — *Continued*

TOPIC	DEFINITY ECS	GENERIC 2
VDN Access/ Capacity	COR checking is used for access to a VDN and for routing to a station.	No restriction checking is used to access a VDN. **NOTE:** Both G2 and DEFINITY ECS use the Facility Restriction Level (FRL) associated with the VDN for outgoing trunk calls.
	COR checking is used when routing locally from a vector.	No restriction check is implemented for local routing.
	A maximum of 500 VDNs [R5si+ memory upgrade (R3 CMS)], 100 VDNs [R5si (R3 CMS), R5vs (R3 CMS)], or 20000 VDNs [R5r (R3 CMS)] can be used.	The maximum number of VDNs is limited only by the number of extensions capacity (32K).
Voice Mailbox	*messaging split* command is used.	Calls are routed to a messaging split via a route to another VDN assigned to a vector with a queue to AUDIX.
Miscellaneous	Changes made to vector administration take effect upon submission. These changes can affect current calls.	A “scratch” pad is used for vector changes. Consequently, only new calls that enter the vector receive the treatment specified in the corrected vector. Vector processing for existing calls is completed in the old vector.

Continued on next page

Table F-7. General Call Vectoring Functional Differences — *Continued*

TOPIC	DEFINITY ECS	GENERIC 2
Miscellaneous (continued)	An existing vector can not be copied to another blank vector. (This capability, however, is available via CMS administration.)	These capabilities are provided by the switch administration.
	Either the VDN or the final destination (but not both) is provided in the CDR record.	Variable format CDR (formerly SMDR) records can be used. Consequently, both the VDN and the final destination can be provided. **NOTE:** CDR records allow the VDN to be specified in the calling party field.
	Blank steps are allowed in vectors, and blank vectors (with no steps defined) may exist.	Blank steps or blank vectors are not allowed (CMS also does not support this).
	Trunk groups can be assigned to VDNs *only* via switch administration.	Trunks groups can be assigned to VDNs via CMS administration.
	Vector processing is limited to a maximum of 1,000 step executions for a call (limit increased to 3,000 with *interflow-qpos* in vector). Once this maximum is reached, processing stops. There is an implied wait of 0.2 seconds for every seven executed steps.	Separate 1,000 step counters are provided for execution of *goto step* commands and *check split* retries. If either counter exceeds 1,000, the call is forced disconnected. Only *check split* retries are counted on internal calls.

Differences in Defining/Interpreting Split Flows

Split flows are defined and/or interpreted according to the switch version and the management system involved. The following sections illustrate how split flow interpretation differs within the DEFINITY ECS and G2 switch versions and according to two management systems, including R3 CMS and R2 CMS.

NOTE:
BCMS is not available on G2 (with or without vectoring).

R3 CMS Standards

Table F-8 illustrates how split flows that occur in the DEFINITY ECS and G2 versions of the switch are interpreted vis-a-vis R3 CMS:

Table F-8. R3 CMS Standards for Interpreting Split Flows

Flow Type	Switch Version	Interpretation
Inflow	DEFINITY ECS with vectoring	Calls answered by a split other than a primary split. **NOTE:** A primary split is the first split to which a call queues.
	G2/traditional ACD	Calls that intraflow from one split's queue to another split's queue (that is, calls that queue to a split after having been previously queued to another split).
Outflow	DEFINITY ECS with vectoring	Calls that are dequeued from a primary split via a *route to* or *messaging split* command, or by being answered by an agent in another split to which the call is also queued.
	G2/traditional ACD	Calls that are taken out of a split's queue and then sent to another destination.
Dequeue	DEFINITY ECS with vectoring	Calls that are dequeued from any split other than the primary split in a VDN.
	G2/traditional ACD	(Not used.)

When a call is not answered (due to an outflow, abandon, busy, or disconnect), the call's disposition is tracked for the primary split. On R3 CMS, the other splits to which the call is queued tracks a dequeue when the call outflows, abandons, is given busy treatment, or is disconnected.

If the primary split in a VDN is unmeasured, a(n) outflow, abandon, busy, or disconnect is not tracked for the call. Also, an answer is not tracked if the call is answered by an agent in the primary split.

R2 CMS Standards

For single split queuing, R2 CMS tracks split inflows and outflows according to the definitions provided in the previous section for "G2/traditional ACD."

However, when multiple split queuing is involved, a call can look like two or three separate calls to R2 CMS. As a result, if a call is queued to multiple splits and is then answered by an agent in one of these splits, an *inflow* is not tracked in R2 CMS. However, if a call is *requeued* to one or more splits (via a *route to*

command, for example), an *inflow* is tracked *only in the first split to which the call requeues*.

Also, when multiple split queuing is involved, R2 CMS tracks an *outflow* in those splits to which the call queues and from which it eventually dequeues without being answered there. In effect, then, R2 CMS tracks an outflow in the same situations where R3 CMS tracks a dequeue.

Differences Between G2 and DEFINITY ECS EAS

This section lists the differences between release G2 and DEFINITY ECS for EAS.

- G2.2 does not have logical agent capabilities.
  - — Voice terminals are preassigned to default skill groups (groups ending in zero).
  - — Agents sharing voice terminals must have the same default skill group.
  - — The voice terminal extension is used to provide a name, COR, and coverage path.
- DEFINITY ECS logical agent provides the following:
  - — Any voice terminal can be used as an ACD terminal for any skills.
  - — Agents can be reached by dialing their login IDs.
  - — Name, COR, and coverage path follow the agent to the voice terminal currently logged into.
- G2.2 does not support Direct Agent Calling.
- G2.2 does not support Call Prompting.
- G2.2 login procedure is: dial feature access code, dial login ID twice. DEFINITY ECS login procedure is: dial feature access code, dial login ID, dial optional password.
- G2.2 restricts agents with multiple skills to skills in the same skill tens group (for example, skill 20-29). DEFINITY ECS allows agent to be in any combination of skills.
- G2.2 restricts calls queuing to multiple skills simultaneously to skills in the same skill tens group. This also applies to VDN skills. DEFINITY ECS allows calls to queue to any three skills simultaneously.
- G2.2 administers agents to a default skill and the agents enter their other skills after logging in. DEFINITY ECS administers all of the agents' skills, and the agents are logged into all of their assigned skills during login. DEFINITY ECS agents can change their skills.

- CMS can only change an agent's default skill on G2.2 (when the agent is unstaffed). CMS can change all skills for an agent on DEFINITY ECS (change affected the next time the agent logs in).
- G2.2 does not support skill levels for agents. This also implies G2.2 does not support expert agent distribution (EAD). DEFINITY ECS does support skill levels for agents and EAD.
- On G2.2, when a change is made to a VDN skill preference, only new calls to the VDN will be impacted by the change. On DEFINITY ECS when a change is made to a VDN preference, existing calls will be impacted as they encounter a vector step that references the VDN skill preference.

Interactions Between Call Vectoring/EAS and BCMS/CMS

G

Introduction

Call Vectoring and EAS interact with a management information system that helps to monitor and report on the activity within Call Vectoring and EAS. In most cases, the management system is either the Call Management System (CMS) or the Basic Call Management System (BCMS).

CMS, which resides on an adjunct processor, collects and processes ACD information to generate various reports. BCMS performs the same duties. The main difference between CMS and BCMS is that the latter resides on the customer switch. Also, it should be noted that CMS reporting and data storage capabilities are much more extensive than those of BCMS.

This chapter is intended to illustrate how these management systems interpret and report on activity within Call Vectoring and EAS. Special emphasis is placed on interpreting and reporting on this activity as it occurs within splits during a series of Call Vectoring or EAS events.

NOTE:
The manual pages in Appendix A provide a summary of the CMS/BCMS interactions with each Call Vectoring command (where applicable).

BCMS/CMS Tracking in a Call Vectoring Environment

Tracking is the identifying of various call flows and other actions relevant to call handling. For our purposes, there are three classes of call flows: split flows, VDN flows, and vector flows. Also, we are most concerned with tracking in the Call Vectoring environment. The specific types of call flows and actions in this environment that are tracked by BCMS/CMS include the following:

- Inflows (flow ins)
- Outflows (flow outs)
- Dequeues
- Abandons
- Answers
- Busies
- Disconnects

The split supervisor can use VDN and vector flows to evaluate how effective vector programming is at the site in question. The supervisor can use split flows to determine the manner in which the splits at the site are handling incoming telephone calls.

Defining and Interpreting Call Flows

The manner in which specific call flows are defined and interpreted depends upon the call flow class in question, the management system in effect, and the version of the DEFINITY System being used. Management systems include R3 CMS, R2 CMS, and BCMS.

The following sections define and interpret specific call flows according to these parameters.

Answered and Abandons

The most important tracking items for most VDNs and vectors are the number of calls answered and the number of calls abandoned. R3 CMS provides VDN profiles that show when calls are answered and abandoned. Ten service level intervals are administered for these profiles. These intervals can have smaller time intervals around the time most calls are answered and when most calls abandon to get more detailed information.

This data can be used to determine what an acceptable service level is for most callers. The percentage answered within the administered acceptable service level is also shown on the Call Profile reports. For VDNs, the calculation is ACD calls answered and nonACD calls connected within the service level divided by calls offered to the VDN (including calls that inflow to the VDN).

For split/skill statistics, the calculation is ACD calls answered within the service level divided by calls queued to the split/skill (answered calls, abandoned calls, calls that flow out, calls that dequeue). In most cases the VDN percentage will be higher then the split percentage since calls dequeued from a split/skill are counted as answered, abandoned, or outflows for the VDN.

Changes made to a vector or to staffing will typically affect the VDN call profile. Even the wording of an announcement can affect the abandon profile. It is worthwhile to review the VDN's call profile before and after any change to determine if the change had a positive impact.

Busies and Disconnects

Busy calls and forced disconnects reported on CMS indicate how many calls this VDN/vector turned away. If forced disconnect is used out of business hours, this item would indicate how many customers expected you to be operating during a specific time interval. If busies are given when the queues are full or waiting times are long, the number of busies in an interval might suggest a staffing change is needed. If disconnect is used to deny a lookahead interflow attempt, a large number of denials would indicate a busy time at multiple sites.

VDN Inflows and Outflows

The following section discusses the specific VDN flows vis-a-vis R3 CMS and BCMS.

R3 CMS and BCMS Standards

Table G-1 illustrates how R3 CMS and BCMS interpret specific VDN flows for the DEFINITY System:

Table G-1. R3 CMS and BCMS Standards for Interpreting VDN Flows

Flow Type	Management System	Interpretation
VDN flow in	R3 CMS	Calls that flow into the VDN via a route-to VDN command or by Redirection on No Answer to a VDN.
	BCMS	(Not tracked.)
VDN flow out	R3 CMS	Calls that successfully flow out of a VDN to another VDN or to an external location via a *route-to* command.
	BCMS	Same as for CMS.

Vector Inflows and Outflows

The following section discusses the specific vector flows as recorded by R3 CMS.

R3 CMS Standards

Vector flow in pertains to calls that flow into a vector from another vector via a *route to* or a *goto vector* command. *Vector flow out* pertains to calls that successfully flow out of a vector via a *route to* or a *goto vector* command.

Split Inflows, Outflows, and Dequeues

The following sections discuss the various split flow types vis-a-vis R3 CMS, R2 CMS, and BCMS.

R3 CMS and BCMS Standards

R3 CMS and BCMS are grouped together because both of these systems interpret two split flow types identically. These flows include *inflow* and *outflow.* However, whereas R3 CMS interprets another split flow type, namely *dequeue*, BCMS does not do so because this system does not have a dequeue tracking item. This means that in a situation where R3 CMS tracks a dequeue, BCMS does not because it is unable to do so.

Before we detail how R3 CMS and BCMS interpret split flows, we should discuss the term primary split, since this concept plays a significant role in tracking. Primary split is defined as the first split in a VDN to which a call actually queues. Therefore, this split is not necessarily the first split referenced in the vector.

Another split becomes the primary split if either of the following events occurs:

- Call cannot queue to the originally-targeted split because the split has no queue slots available.
- Call leaves the VDN (via a *route-to* VDN command, for example) and is queued to another split as a result.

If the call leaves vector processing and does not queue to another split (as a result of a *route-to* extension command, for example), there is no new primary split.

With this discussion in mind, let's take a look at Table G-2 to see how R3 CMS and BCMS interpret split flows for the DEFINITY System:

Table G-2. R3 CMS and BCMS Standards for Interpreting Split Flows

Flow Type	Management System	Interpretation
Inflow	R3 CMS	Calls that ring at an agent in a split other than the primary.
	BCMS	(Same as for R3 CMS.)
Outflow	R3 CMS	Calls that are dequeued from a primary split via a *route-to* or *messaging split* command, or by ringing at or being answered by an agent in another split to which the call is also queued.
	BCMS	(Same as for R3 CMS.)
Dequeue	R3 CMS	Calls that are dequeued from and not answered by any split other than the primary split in a VDN.
	BCMS	(Not tracked.)

When a call is not answered (due to an outflow, abandon, busy, or disconnect), the call's disposition is tracked for the primary split as long as the call is still queued when the call abandons, outflows, etc. However, if the call abandons or outflows from ringing, the disposition is recorded for the split for which it was ringing. On R3 CMS, the other splits to which the call is queued track a dequeue when the call outflows, abandons, is given busy treatment, or is disconnected.

If the primary split in a VDN is unmeasured, an outflow, abandon, busy, or disconnect is not tracked for the call. Also, an answer is not tracked if the call is answered by an agent in the primary split.

R2 CMS Standards

When multiple split queuing is involved, a call looks like two or three separate calls to R2 CMS. As a result, if a call is queued to multiple splits and is then answered by an agent in one of these splits, an inflow is not tracked in R2 CMS. However, if a call is requeued to one or more splits (via a *route-to* command, for example), an inflow is tracked only in the first split to which the call requeues.

Also, when multiple split queuing is involved, R2 CMS tracks an outflow in those splits to which the call queues and from which it eventually dequeues without being answered there. In effect, then, R2 CMS tracks an outflow in the same situations where R3 CMS tracks a dequeue.

Examples of Split Flow Tracking

The following sections provide some examples of tracking in R3 CMS, R2 CMS, and BCMS. Each section first presents a scenario of Call Vectoring events. The scenario is then followed by a table in which the tracking for the various splits involved is recorded. Following each "tracking table," an explanation of the tracking procedure is provided.

The scenarios presented include the following:

- Call answered by a primary split
- Call answered by a nonprimary split
- Call abandoned from queue
- Call answered by a primary split after a route to VDN
- Call answered by a nonprimary split after a route to VDN
- Call answered after a route to split

NOTE:
Inflows, outflows, and dequeues are not tracked for splits administered by the *converse-on split* command. However, if a call is answered both by a converse split and (subsequently) by a nonconverse split, an "answer" is tracked for each split. However, a call is really considered "answered" only when it is answered by a nonconverse split. Therefore, traffic measurements for converse splits should be used only to measure converse split traffic and not to calculate the total number of calls.

Call Answered by a Primary Split. The following scenario involves a call answered by the primary split. The scenario is as follows:

1. Call comes into a VDN whose vector queues the call to splits 1, 2 and 3.
2. Call is answered in split 1.

Table G-3 shows the tracking table for this scenario:

Table G-3. Tracking for Call Answered by Primary Split

	Split Tracking		
	1	**2**	**3**
R3 CMS	answer	dequeue	dequeue
BCMS	answer		
R2 CMS	answer	outflow	outflow

Comments:

- **R3 CMS**: Dequeue is tracked in split 2 as well as in split 3 because the call is answered by the primary split (split 1) and is thus dequeued from splits 2 and 3 without being answered in these splits.
- **BCMS**: No dequeue tracking item is available.
- **R2 CMS**: Outflow is tracked in the same situations where R3 CMS tracks a dequeue. Accordingly, outflow is tracked in splits 2 and 3 because the call is dequeued from these splits without being answered in either one of the splits.

Call Answered by a Non-Primary Split. The following scenario involves a call answered by a nonprimary split. The scenario is as follows:

1. Call comes into a VDN whose vector queues the call to splits 1, 2 and 3.
2. Call is answered in split 2.

Table G-4 shows the tracking table for this scenario:

Table G-4. Tracking for Call Answered by Non-Primary Split

	Split Tracking		
	1	**2**	**3**
R3 CMS	outflow	inflow answer	dequeue
BCMS	outflow	inflow answer	
R2 CMS	outflow	answer	outflow

Comments:

- **R3 CMS**: Outflow is tracked in split 1 because the call is answered by an agent in another split to which the call is queued (that is, split 2). Although the call is obviously removed from split 1 after it is answered in split 2, dequeue is not tracked in split 1 because split 1 is the primary split. Inflow is tracked in split 2 because the call is answered in this split and the split is not the primary split. Dequeue is tracked in split 3 because the call is removed from the split without being answered there. When the call is removed from split 3, outflow is not tracked in split 3 because this split is not the primary split.
- **BCMS**: Follows the same scheme as R3 CMS except for the dequeue tracking.
- **R2 CMS**: Outflow is tracked in splits 1 and 3 because the call is dequeued from these splits without being answered there.

Call Abandoned. The following scenario involves a call abandoned by the caller. The scenario is as follows:

1. Call comes into a VDN whose vector queues the call to splits 1, 2 and 3.
2. Call is abandoned.

Table G-5 shows the tracking table for this scenario:

Table G-5. Tracking for Abandoned Calls

	Split Tracking		
	1	**2**	**3**
R3 CMS	abandon	dequeue	dequeue
BCMS	abandon		
R2 CMS	abandon	outflow	outflow

Comments:

- **R3 CMS**: Abandon is tracked in split 1 because this split is the primary split. *Dequeue* is tracked in splits 2 and 3 because the call is dequeued from these splits without being answered in either split.
- **BCMS**: Abandon is tracked in split 1 because this split is the primary split. Tracking is not recorded in splits 2 and 3 because no dequeue tracking item is available.
- **R2 CMS**: Abandon is tracked in split 1 because this split is the primary split. Outflow is tracked in splits 2 and 3 because the call is dequeued from these splits without being answered in either one of the splits.

Call Answered by a Primary Split after a Route To VDN. The following scenario involves a call answered by the primary split after a *route-to* VDN command is executed. The scenario is as follows:

1. Call comes into a VDN whose vector queues the call to splits 1, 2 and 3.
2. Vector executes a *route-to VDN* step.
3. Call is then queued to splits 4, 5 and 6.
4. Call is answered in split 4.

Table G-6 shows the tracking table for this scenario:

Table G-6. Tracking for Call Answered by Primary Split after Route to VDN

	Split Tracking					
	1	**2**	**3**	**4**	**5**	**6**
R3 CMS	outflow	dequeue	dequeue	answer	dequeue	dequeue
BCMS	outflow			answer		
R2 CMS	outflow	outflow	outflow	inflow answer	outflow	outflow

Comments:

Split 1 is the original primary split, because this is the first split to which the call actually queues. However, split 4 becomes the new primary split because:

- Call leaves the original VDN upon execution of the *route-to VDN* step.
- Split 4 is the first split to which the call queues upon execution of this step.
- **R3 CMS**: Outflow is tracked in split 1 because this split is the original primary split, and the call is dequeued from this split via a *route-to VDN* step. Dequeue is tracked in splits 2, 3, 5, and 6 because the call is dequeued from each of these splits without being answered in any one of them.
- **BCMS**: Follows the same scheme as R3 CMS except for the dequeue tracking.
- **R2 CMS**: Outflow is tracked in splits 1, 2, 3, 5 and 6 because the call is dequeued from these splits without being answered in any one of them. Inflow is tracked in split 4 because split 4 is the first split to which the call requeues after the *route to* command is executed.

Call Answered by the Non-Primary Split after a Route To VDN. The following scenario involves a call answered by the nonprimary split after a *route-to* VDN command is executed. The scenario is as follows:

1. Call comes into a VDN whose vector queues the call to splits 1, 2 and 3.
2. Vector executes a *route-to* VDN step.
3. Call is then queued to splits 4, 5 and 6.
4. Call is answered in split 5.

Table G-7 shows the tracking table for this scenario:

Table G-7. Tracking for Call Answered by Non-Primary Split after Route to VDN

	Split Tracking					
	1	**2**	**3**	**4**	**5**	**6**
R3 CMS	outflow	dequeue	dequeue	outflow	inflow answer	dequeue
BCMS	outflow			outflow	inflow answer	
R2 CMS	outflow	outflow	outflow	inflow outflow	answer	outflow

Comments:

- **R3 CMS**: Outflow is tracked in split 1 because this split is the original primary split, and the call is dequeued from this split via a *route-to* VDN step. Dequeue is tracked in splits 2, 3, and 6 because the call is dequeued from each of these splits without being answered in any one of them. Outflow is tracked in split 4 because this split becomes the *new* primary split after the *route-to VDN* step is executed, and the call is subsequently dequeued from this split by being answered in another split (split 5) to which the call is also queued. Finally, inflow is tracked in split 5 because the call is answered in this split, and the split is not the primary split.
- **BCMS**: Follows the same scheme as R3 CMS except for the dequeue tracking.
- **R2 CMS**: Outflow is tracked in splits 1, 2, 3, 4, and 6 because the call is dequeued from these splits without being answered in any one of them. Inflow is tracked in split 4 because this split is the first one to which the call is requeued after the *route-to* command is executed.

Call Answered after a Route To Split. The following scenario involves a call answered after it is routed to a split via a *route-to digits* or *messaging split* command. The scenario is as follows:

1. Call comes into a VDN whose vector queues the call to splits 1, 2 and 3.
2. Vector executes a *route-to digits* (or *messaging split*) step.
3. Call is queued to split 4 and answered by an agent in split 4.

Table G-8 shows the tracking table for this scenario:

Table G-8. Tracking for Call Answered after Route to Split

	Split Tracking			
	1	**2**	**3**	**4**
R3 CMS	outflow	dequeue	dequeue	answer
BCMS	outflow			answer
R2 CMS	outflow	outflow	outflow	inflow answer

Comments:

- **R3 CMS**: Outflow is tracked in split 1 because this split is the original primary split, the call is dequeued from this split via a *route-to digits* (or *messaging split*) step, and the call is answered in split 4, which becomes the new primary split. Dequeue is tracked in splits 2 and 3 because the call is dequeued from each of these splits without being answered in any one of them.
- **BCMS**: Follows the same scheme as R3 CMS except for the *dequeue* tracking.
- **R2 CMS**: Outflow is tracked in splits 1, 2, and 3 because the call is dequeued from these splits without being answered in any of them. Inflow is tracked in split 4 because this split is the first one to which the call is requeued after the *route-to digits* (or *messaging split*) command is executed.

Evaluating Split Performance

By using the information presented to this point, along with the information from various reports (as discussed in the next section), the split supervisor can answer one or more questions concerning split performance and then make adjustments, if necessary. Here are some of the questions the supervisor can answer:

1. How many ACD calls offered to my split were "mine" (that is, were offered to this split as the primary split)?

   NOTE:
   Split "ACD calls" include Direct Agent Calls for BCMS and for R2 CMS, but not for R3 CMS, which tracks Direct Agent Calls separately.

2. How many of "my" ACD calls did "my" split not answer?
3. How many ACD calls that I didn't answer weren't "mine?"

The following sections present the answers to these questions from the perspective of R3 CMS, BCMS, and R2 CMS.

R3 CMS. The following answers reflect the use of R3 CMS:

1. The number of calls offered to "my" (primary) split that were "mine" can be determined via examination of the CMS Split Summary Report. The algorithm is as follows: CALLSOFFERRED - INFLOWCALLS - DEQUECALLS (that is, the total number of calls offered *minus* the number of calls not "mine" that I answered *minus* the number of calls not "mine" that I didn't answer.)
2. The number of "my" calls that "my" split didn't answer can be determined via examination of the CMS VDN Report. The algorithm is as follows: ABNCALLS + BUSYCALLS + DISCCALLS + OUTFLOWCALLS (that is, the number of abandoned calls plus the number of busy calls plus the number of disconnected calls plus the number of calls outflowed from "my" split tagged as a primary split).
3. The number of calls not "mine" that "my" split didn't answer is DEQUECALLS, which is indicated in the CMS Split Summary Report.

BCMS. The following answers reflect the use of BCMS:

1. The number of calls offered to "my" split that were "mine" can be determined via examination of the BCMS Split Report. The algorithm is as follows: ACDCALLS + ABNCALLS + OUTFLOWCALLS - INFLOWCALLS (that is, the total number of calls answered plus the total number of calls abandoned from "my" split tagged as a primary split plus the number of calls that outflowed "my" split tagged as a primary split minus the number of calls answered that were not directed to "my" split tagged as a primary split).

The other two questions cannot be answered because BCMS does not have a dequeue tracking item.

R2 CMS. Customers using R2 CMS connected to G1/G3 with vectoring enabled cannot *necessarily* answer any of the questions. If multiple-split queuing is involved, the OUTFLOWCALLS contains both "my" calls and other splits' calls that outflowed. As a result, the answers to the questions cannot be calculated.

Using BCMS/CMS Reports to Evaluate Call Vectoring Activity

There exists a number of CMS and BCMS reports that allow the customer to evaluate various facets of Call Vectoring activity. Some of these facets include the call flows present within Call Vectoring as well as the speeds at which calls are answered. The sections that follow identify and discuss the CMS and BCMS reports that indicate this activity.

CMS Reports

CMS has real-time and historical reports. Most CMS historical reports are available in four versions: intra-hour, daily, weekday, and monthly. The following list identifies and describes several CMS reports that summarize Call Vectoring activity. For further details on these and other related reports, refer to your CMS documentation.

NOTE:
The reports described in this section are generated in R3 CMS. Corresponding R2 CMS reports do not provide information that reflects capabilities that are new to the DEFINITY System (for example, internal/external call tracking).

- **Split Summary Report** summarizes the call activity for an entire split. Among other information, the report provides the number of calls answered, the total number of flow ins (inflows), flow outs (outflows), dequeues, and abandoned calls.

  The report also indicates the average speed of answer (interval ASA) for calls. This refers to the sum of the queue time and ring time for a call within the answering split *only*. Finally, the report indicates the dequeued average queue time, which is the average time a call waits until it is answered by another split to which the call is also queued.

- **VDN Report** summarizes VDN activity for specific vectors. Among other information, the report provides calls answered, connected, abandoned, the number of VDN Flow Ins/Outs, calls forced busy, and calls forced disconnect. VDN Flow In pertains to calls that flow into a VDN from another VDN via a *route-to* command. VDN Flow Out pertains to calls that successfully flow out of VDN to another VDN or external location via a *route-to* command.

- **Vector Report** summarizes vector activities. Among other information, the report provides the number of calls offered, calls answered, calls abandoned, Vector Flow Ins/Outs, calls forced busy, and calls forced disconnect. Vector Flow In pertains to calls that flow into a vector from another vector via a *route-to* or *goto vector* command. Vector Flow Out pertains to calls that successfully flow out of a vector via a *route-to* or *goto vector* command.

BCMS Reports

BCMS has a real-time split report, split historical reports, real-time VDN reports, and VDN historical reports. The following list identifies and describes several BCMS reports that summarize Call Vectoring activity. For more information on these and other related reports, refer to *DEFINITY Enterprise Communications Server Basic Call Management System (BCMS) Operations* manual 555-230-706.

- **BCMS Split Report** summarizes the call activity for an entire split. The information can be requested either daily or by the administered time period. Among other information, the report provides the total number of flow ins (inflows) and flow outs (outflows), the calls answered and calls abandoned. The report also provides the average speed of answer time for calls handled by the split during the indicated time period.
- **VDN Summary Report** summarizes statistical information for all internally-measured VDNs. The information can be requested by the administered time interval or daily. The "list bcms vdn" report gives multiple time periods or days for a single VDN. The "list bcms summary vdn" report gives a one-line summary per vdn (with data from the specified times or days), but can give the data for numerous vdns.

  The report also indicates the total number of flow outs, specifically, the number of calls that route to another VDN or to a destination external to the switch. However, calls that encounter a *goto vector* command are not shown as outflows. No further measurements are taken on the calls once the calls have outflowed. If an outflowed call later abandons, this is not indicated in the report.

  Among other information, the VDN report provides a total for offered calls, answered calls, abandoned calls, and also one for calls that were either "forced busy" or "forced disconnect."
- **VDN Real-Time Report** provides statistical information including the number of calls currently waiting and the oldest call waiting. The VDN real-time report has the same characteristics as other real-time BCMS reports.

Using CMS in an EAS Environment

CMS R3V2 or a later release is required to support Expert Agent Selection (EAS). The same tracking and database items used within a traditional Call Vectoring environment are used within an EAS environment but there are also new items that are specific to EAS. All existing custom reports should work when you are upgrading to EAS.

Tracking Entities

The following sections explain how the following entities are tracked in an environment with EAS optioned:

- Agents and their skills
- Direct Agent Calls
- Non-ACD Calls
- VDN Skill Preferences

Agents and their Skills

The fields under the "Extn" column in the CMS Real-Time Agent Report show the voice terminal extension that the agent is logged into. These fields can be used to locate the agent or to service observe the agent.

With EAS optioned, the Skill Status Report replaces the Split Status Report. This report indicates the skills logged into and the skill level of each skill. If too many calls are waiting, or if calls are waiting too long (also shown on the Skill Status report), it is possible that not enough agents have the skill administered at a high enough skill level.

An agent may be denied login to some skills if the maximum agents/skill number is met or if the CMS limit on agent/skill pairs logged in has been reached.

The Login/Logout Historical Report also lists up to 15 of the agent's skills and the skill levels for each.

Direct Agent Calls

Waiting Direct Agent calls are not included in the "Calls waiting" and "Oldest Call Waiting" report fields for skills because such calls are not skill calls. However, Direct Agent Calls are included in these two report fields for VDNs.

The Queue/Agent Summary Real-Time Report lists separately the Direct Agent calls waiting in a skill queue. Direct Agent calls are queued to the skill that is administered as the Direct Agent Skill. To manage the skill's queue slots effectively, it is recommended that a skill be dedicated for Direct Agent calls.

Since Direct Agent calls are not skill calls, the skill tables do not track Direct Agent calls; however, the tables do monitor skill queue slots. The agent's time is tracked as "OTHER" in the skill tables. In the agent tables, there are separate Direct Agent call items. The standard CMS agent reports add the Direct Agent calls and the skill ACD calls and report these calls as "ACD Calls." The VDN tables track Direct Agent calls as ACD calls.

Non-ACD Calls

The first measured skill that an EAS agent is logged into is used by CMS to track non-ACD calls unless the agent has an ACD call on hold. If an ACD call is on hold, outgoing non-ACD calls are counted for the skill of the held ACD call.

VDN Skill Preferences

VDN skill preference data is collected to provide information on what groups of agents (skills) are handling calls and on how effectively each skill group handles a particular VDN.

Real-time and historical VDN Skill Preference reports can be used to compare the percentage of calls being answered by the 1st, 2nd, and 3rd VDN

preferences against an objective. If too few calls are being answered by the 1st skill preference, the vector can be adjusted to allow more time for the 1st skill preference group to answer calls; another alternative is to train or hire more agents with the 1st skill preference.

You can use VDN skill preference data to compare the average talk time and average ACW time for agents in the 1st, 2nd, and 3rd skill groups. If these times vary too much across groups, more training may be needed for the backup groups (that is, the 2nd and 3rd skill groups).

VDN skill preference data is tracked according to the skill preferences (1st, 2nd, 3rd) assigned to the VDN. Whenever a vector step either references a 1st, 2nd, or 3rd skill or specifies a skill number that matches the 1st, 2nd, or 3rd skill administered, the new database items are tracked. For example, if VDN 1000 has Skills 21, 22, and 23 administered as the 1st, 2nd, and 3rd skills, respectively, and if the vector associated with VDN 1000 has a "queue to main skill 22" step, tracking occurs for the 2nd VDN skill preference if the call is answered by an agent in Skill 22. Skill preference tracking also occurs for Skills 21 and 23. This allows users who prefer to specify the actual skill number in the vector to take advantage of the tracking for VDN skill preferences.

EAS Administration from CMS

R3V2 and later release CMS can be used to administer vectors as well as skills for agents and VDNs. The ACD Administration: Change Agent Skills CMS screen is used to display and modify the skills and levels assigned to an agent, as well as the assigned Direct Agent skill and call handling preference.

The ACD Administration: Change VDN Skill Preferences screen is used to request a VDN’s skill preferences and to modify the VDN’s skills.

The CMS Vector Contents screen is used to create and modify vectors. CMS supports the Call Vectoring commands that queue calls to the 1st, 2nd, or 3rd VDN skill.

Operation Details for the Route-to Command

H

Introduction

The *route-to* command can be programmed with or without coverage. Table H-1 in this appendix summarizes the operation of the *route-to* command for each of the destination types and conditions associated with the commands.

Table H-1. DEFINITY System Route-To Command Operation

	INTERACTION	
CONDITION	**cov = n ANY STEP**	**cov = y ANY STEP[1]**
Invalid Destination[2]	Goes to next step, else stop	Goes to next step, else stop
VDN Extension[3] - Vector Assigned - Vector Has No Steps	Goes to new vector Stop[4]	Goes to new vector Stop[4]

Continued on next page

Table H-1. DEFINITY System Route-To Command Operation — *Continued*

	INTERACTION	
CONDITION	**cov = n ANY STEP**	**cov = y ANY STEP[1]**
Station Extension Idle (all appearances idle)		
- CF-ALL Active or -CF-DA Applies	Forwards if possible, else next step, else stop[4]	Forwards if possible, else coverage, else busy
- Coverage		
- DA Applies	Rings idle app.	Coverage on DA
- All Applies	Goes to next step, else stop[4]	Coverage
- SAC Applies	Rings idle appearance	Coverage
- None of Above Applies	Rings idle appearance	Call delivered and is allowed to cover
Station Extension Active (with idle 2-way app)		
- CF-ALL Active	Forwards if possible, else next step, else stop[4]	Forwards if possible, else coverage, else busy
- Coverage		
- DA Applies	Rings idle app (no DA timing)	Coverage on DA
- Ext Act Applies	Rings idle appearance	Coverage
- All Applies	Goes to next step, else stop[4]	Coverage
- SAC Applies	Rings idle appearance	Coverage
- None of Above Applies	Rings idle appearance	Call delivered and is allowed to cover
Station Extension Busy (no idle 2-way app)		
- Extension in Hunt Grp (also see ACD Hunt Grp)	Queues if possible, else next step, else stop[4]	Queues if possible,else coverage, else busy
- CF-ALL Active or -CF-DA Applies	Forwards if possible, else next step, else stop[4]	Forwards if possible, else coverage, else busy
- Call Waiting to Analog Sta Would Apply	Goes to next step, else stop[4]	Call waits
- Coverage		
- Ext Act Applies	Goes to next step, else stop[4]	Coverage
- Ext Bsy Applies	Goes to next step, else stop[4]	Coverage
- All Applies	Goes to next step, else stop[4]	Coverage
- SAC Applies	Goes to next step, else stop[4]	Coverage
- None of Above Applies (or hunt, fwd, or cov dest is unavailable)	Goes to next step, else stop[4]	Busy tone given

Continued on next page

Table H-1. DEFINITY System Route-To Command Operation — *Continued*

CONDITION	INTERACTION cov = n ANY STEP	 cov = y ANY STEP[1]
Extension with Incompatible COR	Goes to next step, else stop.	Goes to next step, else stop.
Terminating Extension Group		
- All Members Idle	Rings idle appearance	Call delivered and is allowed to cover
- A Member Active on TEG	Goes to next step, else stop[4]	Coverage, else busy
- No Idle App on Any Member	Goes to next step, else stop[4]	Coverage, else busy
Hunt Group Extension		
- Idle Agent	Rings idle appearance	Call delivered and is allowed to cover
- No Idle Agent		
- Call can't queue	Goes to next step, else stop[4]	
- Call can queue	Call is queued	Busy tone given Call is queued
Extension on Another Node (Uniform Dialing Plan - UDP DCS or non-DCS)		
- Trunk Available	Call delivered	Call delivered
- Trunk Not Available	Goes to next step, else stop[4]	Queues if possible, else reorder
- No DCS Buffer for Routing	Call delivered w/o DCS msg	Call delivered w/o DCS msg
Trunk Access Code (TAC) Destination		
- Trk Grp No Dial Access	Goes to next step, else stop[4]	Routes to local and
- Trunk Available	Call delivered	Call delivered
- Trunk Not Available	Goes to next step, else stop[1]	Queues if possible, else reorder
AAR/ARS FAC Dest. (including Subnet Trkng)		
- Trk Grp No Dial Access	Tries next route	Routes to local attendant
- Trunk Available	Call delivered	Call delivered
- Other Routes Avail	Tries next route	Tries next route
- All Routes Busy		
- No Pattern Queuing	Goes to next step, else stop[4]	Reorder tone given
- Queuing Assigned	Goes to next step, else stop[4]	Queues to pattern

Continued on next page

Table H-1. DEFINITY System Route-To Command Operation — *Continued*

	INTERACTION	
CONDITION	**cov = n ANY STEP**	**cov = y ANY STEP[1]**
Attendant Queue (dial 0) - Idle Atnd	Rings idle appearance	Call delivered and is allowed to cover
- No Idle Atnd - Not In Night Svc	Call is queued	Call is queued
- In Night Svc - Nite Dest. Assigned	Delivered to night svc.	Delivered to night svc.
- Not Assigned	Call is queued	Call is queued
Individual Attendant Access - Atnd Idle	Rings idle appearance	Call delivered and is allowed to cover
- Atnd Busy	Queues if possible else Goes to next step, else stop[4]	Queues if possible, else Busy tone given
CAS Attendant With Caller on Branch - RLT Available	Rings idle appearance	Call delivered and is allowed to cover
- All RLTs Busy	Queues if possible, else next step, else stop[4]	Queues if possible, else busy tone
Inter-PBX Atnd Calling - Trk Grp Controlled	Routes to local atnd	Routes to local atnd
- Trk Available	Call delivered	Call delivered
- Trk Not Available	Goes to next step, else stop[4]	Reorder tone given

Continued on next page

Table H-1. DEFINITY System Route-To Command Operation — *Continued*

CONDITION	INTERACTION cov = n ANY STEP	INTERACTION cov = y ANY STEP[1]
Look Ahead Interflow (LAI) (feat. active & routes over ISDN-PRI facility)[5]		
- B-Channel Not Avail	Goes to next step, else stop[4]	Queues if possible, else reorder
- B-Channel Avail and		
- Receiving Switch		
- Accepts	Interflow succeeds[6]	Call cut-through
- Rejects	Goes to next step, else stop[4]	Call gets busy/disconnect
- Receiving Switch w LAI Acting as Tandem Sees from Remote Receiving Sw.		
- Call Accepted	Interflow succeeds[6]	Call cut-through
- Call Rejected	Goes to next step at receiving switch, else sending switch considers call rejected after 2-minute timeout	Call gets busy/disconnect
if interflow-qpos	Determines if queued call is eligible for interflow	Determines if queued call is eligible for interflow

1. The call is removed from vector processing (that is, the call is taken out of any split queue, and any feedback, such as music or ringback, is removed) for *with coverage y* interactions, even if the destination is not available. The call is treated as though the destination was directly dialed. This includes coverage, forwarding, treatments for calls that cannot be completed (busy, reorder, and intercept) and displays (answering station sees only caller name and number). See note 3 in this chapter for related information. Specifically, the call is taken out of vector processing when the step is reached, regardless of facility or remote switch availability. If the call is subsequently “rejected” by the receiving switch vector, the call is given the treatment defined by the “rejection” command (either busy or forced disconnect). Finally, note that a call routed via an *adjunct routing* command is treated the same way as a call that is routed via a *route-to with coverage y* command.

2. Invalid destinations include the following: empty (for example, zero collected digits) or invalid *route-to* destination number, unassigned extension number, incomplete number of digits for AAR/ARS pattern, non-AAR/ARS feature access code (FAC), maintenance busy station extension, COR of the VDN that prevents access (for example, origination restricted), FRL of a VDN that is lower than required for the AAR/ARS pattern access, no routes assigned to the AAR/ARS pattern, incompatible calling and destination partitions, ACTGA trunk group destination, or an off-net forwarding destination. If a TAC (trunk access code) destination is involved, and if the TAC is for a CO/FX trunk with a *route-to with coverage n* step, the digits entered must match a valid ARS analysis string. If not, the destination is considered invalid. For other trunk types with a *route-to number* or *route-to digits with coverage n* step, the step succeeds when the trunk is seized (that is, vector processing stops). For a *route-to with coverage y* step, the step succeeds if the TAC is assigned.

3. A call that routes to a VDN via the *route-to number with coverage* = “yes unconditionally” command behaves like a directly- dialed call instead of a VDN call. Therefore, the terminating station’s display only shows the originating station information and does not show the VDN information (for other types of VDN calls, the terminating station would see the VDN name).

4. The interaction "Stop" means the following: vector processing is stopped, the call remains queued to a split, and the caller continues to hear feedback initiated by a previous step. In the case where the *route-to* command fails and processing stops (due to a busy station or trunk group destination), retry can be implemented in the vector. Retrying is accomplished by including an unconditional *goto* step as the last step to allow for a loop back to the *route to* command. Use of an intermediate *wait-time* command step with appropriate feedback and delay interval is strongly recommended in order to reduce processor occupancy.

5. With one exception, any *route-to with coverage y* step that routes over ISDN-PRI facilities cancels Look-Ahead Interflow. The exception occurs when a call reaches a vector via coverage to a VDN. Calls that cover to a VDN will not be further forwarded or otherwise redirected. For covered calls, a *route-to* command with coverage set to *y* functions as though coverage were set to *n*. Thus, a *route-to with coverage y* will route covered calls via LAI over ISDN facilities if LAI is enabled.

6. On the sending switch, the call is removed from vector processing (that is, the call is taken out of any split queue and any feedback, such as music or ringback, is removed).

Detailed Call Flow and Specifications for Converse—VRI Calls

I

Introduction

This appendix discusses the detailed call flow for calls involving a *converse-on* vector step and Voice Response Integration (VRI). This call flow is segmented into the following phases:

1. Converse call placement
2. Data passing (optional)
3. VRU data collection (optional)
4. Script execution
5. Data return (optional)
6. Script completion
7. DEFINITY System data collection (optional)

NOTE:
If, during any phase of this call flow, a *converse-on* step is executed while the caller is in the split queue and an agent becomes available to service the caller, the VRU port is dropped, vector processing is terminated, and the calling party is immediately connected to the available agent.

NOTE:
If EAS is optioned, “skill” replaces “split.”

Converse Call Placement

The first action taken by the *converse-on* step is to deliver the call to the converse split. Ringback tone is not heard by the caller. Any audible feedback supplied by vector processing remains until the VRU answers the call and all digits (if administered) have been outpulsed to the VRU. Vector processing is suspended. Callers remain in any nonconverse split queues, and they retain their position in queue while the converse session is active.

If a Call Prompting TN744 TTR is allocated to the call, the TTR is released. Any dial-ahead digits are discarded. However, any digits collected prior to the *converse-on* step are kept.

Calls to busy converse splits are allowed to queue. The priority of the call in queue is administrable within the *converse-on* step. Again, any audible feedback supplied by vector processing continues until the call is answered by the VRU and any data is outpulsed. Calls to busy converse splits have either no queue or a full queue fail. For this scenario, a vector event is logged, and vector processing continues at the next vector step.

Whenever a *converse-on* step places a call to an auto-available split whose agents are all logged out, the call is not queued. Instead, the *converse-on* step fails, a vector event is logged, and vector processing continues at the next vector step.

NOTE:
Usually, this scenario occurs whenever the Voice Response Unit (VRU) goes down, the ports are members of an Auto-Available Split (AAS) and the Redirection on No Answer (RONA) feature has taken all the ports out of service.

The originator's display is not changed by the terminating or answering of a converse call. Also, whenever a call is delivered to a display station via a *converse-on* step, the station displays the following information: "Originator Name to VDN Name." Conventional Call Vectoring rules for Override are in effect.

Valid destinations for converse calls must be vector-controlled and include the following:

- Hunt groups
- ACD (including Auto-Available) splits
- Agent (including Auto-Available) skill groups
- AUDIX hunt groups

NOTE:
Even though AUDIX hunt groups are valid destinations for converse calls, they do not need to be vector-controlled.

Undefined and nonvector-controlled hunt group, split or skill numbers are rejected at administration time.

Any attempt to remove a hunt group, split or skill administered within a *converse-on* vector step is denied until the vector has been changed. Also, any attempt to make a hunt group, split, or skill nonvector-controlled is denied if the hunt group, split, or skill is called by a *converse-on* step.

Data Passing

NOTE:
This phase is optional and is in effect only if the application calls for the DEFINITY System to pass information in-band to the VRU.

The *converse-on* step may outpulse up to two groups of digits to the VRU. The digits can serve two major purposes, as follows:

- Notify the VRU of the application to be executed
- Share call-related data, such as ANI, CINFO or caller digits collected by the DEFINITY System

In many applications both application selection and data sharing are required.

Since in many cases the digit strings are of variable length, the DEFINITY System always appends a pound sign (#) to the end of each digit string. *Prompt and collect* steps in the VRU script must therefore always be administered to expect the pound sign (#) as the end-of-string symbol and to include the pound sign in the digit count.

Sending the pound sign (#) prevents excessive delays and other problems caused by digit timeouts.

The complete outpulse sequence is summarized as follows:

1. VRU answers the call
2. Delay for the time administered in the “Converse first data delay” field in the System Parameters-Features form occurs
3. <data_1> is outpulsed
4. “#” is outpulsed
5. Delay for the time administered in the “Converse second data delay” field in the System Parameters-Features form occurs
6. <data_2> is outpulsed
7. “#” is outpulsed

NOTE:
The length of DTMF tones (digits) and the interdigit pause between tones is administrable on the Feature-Related System Parameters form. The optimum timers for the CONVERSANT are a 100 msec tone and 70 msec pause (administration default). See the *DEFINITY Enterprise Communications Server Release 5 Implementation*, 555-230-302 for details.

Any audible feedback supplied by the DEFINITY System is disconnected only after the outpulse sequence is completed. Also, any touch-tone dialing by the calling party during the data passing phase does not result in data corruption.

The following values may be administered for <data_1> and <data_2> within the *converse-on* command:

- **Administered digit string**: This string can contain up to six characters consisting of one or more digits (0 through 9) or asterisks (*). The pound sign (#) may not be included in a digit string because it is reserved as the end-of-string character. However, a single "#" may be administered.
- **ani**: If the call is a local call or an incoming DCS call, this data type causes the extension of the calling party to be outpulsed. If the call is an incoming ISDN PRI call with ANI (BN) provided to the DEFINITY System, the calling party number/billing number (CPN/BN) of the calling party is outpulsed to the voice information system. If there is no ANI (BN) to send, the end-of-string pound sign (#) is the only character outpulsed. Any other type of incoming call results in "#" being outpulsed.
- **vdn**: This data type causes the VDN extension to be outpulsed. In cases where multiple VDNs are accessed, normal VDN override rules determine which VDN extension is outpulsed.
- **digits**: This data type can be used only if Call Prompting is optioned, and it causes the most recent set of digits collected in vector processing to be outpulsed. If no digits are available, the end-of-string pound sign (#) is the only character outpulsed.
- **qpos**: This data type causes the value of the queue position of a call in a nonconverse split to be outpulsed. This value is a variable length data item from which between one and three digits can be outpulsed. Valid ranges for the value are 1 through 200 in G3si and G3vs and 1 through 999 in G3r. If the call is not queued, the end-of-string pound sign (#) is the only character outpulsed.

  NOTE:
  The use of this keyword is not recommended with multiple split queuing because any queue position value sent may not be meaningful. However, if the call is queued to multiple nonconverse splits, the value of the caller's queue position in the first nonconverse split is sent.

This data may be used by the voice information system to inform callers of their position in queue or to decide whether to execute a long or short version of a voice response script.

- **wait**: This data type sends the expected wait time for a call in vector processing that is queued to at least one split. It is a value from 0 to 9999 seconds (variable length, that is, not padded with zeros) always followed by a # digit. If the call is not queued, or is queued only to splits with no working agents, only the # is outpulsed.
- **"#"**: This is the only character outpulsed. Outpulsing this character causes the corresponding prompt and collect command in the voice response script to be skipped.
- **"none"**: This data type causes no characters to be outpulsed. Also, no end-of-string pound character (#) is outpulsed, and no time delays are invoked.

The DEFINITY System always outpulses a pound character (#) at the end of each digit string. Where "#" is administered, or where the "digits" keyword is administered and the last digit collected from the caller is "#," only one "#" is outpulsed. No "#" is outpulsed when the keyword "none" is administered.

If <data_1> is administered as "none," <data_2> must also be "none."

Any data to be passed to the VRU from the DEFINITY System is outpulsed in-band. Two time delays on the System Parameter-Features form ("Converse first data delay" and "Converse second data delay") are administrable by customers. These delays may range from 0 through 9 seconds, with a default of zero seconds for the converse first data delay and a default of two seconds for the converse second data delay. The delays may be needed to give the VRU time to invoke an application and allocate a touch-tone receiver to receive the passed digits.

If <data_1> is not "none," the converse first data delay timer starts when the call is answered by the VRU. Once the timer expires, the data_1 digits are outpulsed in-band to the VRU, followed by the end-of-string pound sign (#).

If <data_2> is not "none," the converse second data delay timer starts when the end-of-string pound sign (#) from the first digit string is outpulsed. Once the timer expires, the data_2 digits are outpulsed in-band to the VRU, followed by the end-of-string pound sign (#).

No time delays are invoked when the keyword "none" is administered.

NOTE:
The outpulsing of digits is not heard by the caller.

If the VRU hangs up during the data passing phase, the DEFINITY System will log a vector event, reactivate vector processing at the next vector step, and ensure the VRU port is accessible for future calls.

Once all digits have been passed to the VRU, any audible feedback is disconnected.

NOTE:
At this point, control has effectively been passed to the VRU.

To ensure the robust operation of the VRU data passing operation, be sure to implement the following recommendations:

- Include the *prompt and collect* command in the VRU script for each data field passed in the *converse-on* step.
- Administer each *prompt and collect* command to recognize the "#" character as the end-of-string character.
- Ensure the number of digits expected is one greater than the number of digits passed to allow for the "#" character, which terminates every converse data field.

  Also, ensure no announcement is played in these *prompt and collect* steps.
- Ensure the first digit timeout in the *prompt and collect* steps is five seconds greater than the corresponding converse data delay. (For example, if the *converse-on* step passes two data fields, and if the converse first data delay is 0 secs and the converse second data delay is 4 secs, the first digit timeouts for the two *prompt and collect* commands should be at least 5 and 9 seconds, respectively.)
- Ensure the interdigit timeout in the *prompt and collect* steps is at least five seconds.
- Administer the converse first data delay to give a VRU under a heavy load sufficient time to allocate a DTMF touch-tone receiver after answering the call.
- Administer the converse second data delay to give a VRU under a heavy load sufficient time to complete any tasks between the first and second *prompt and collect* command. (For example, the VRU can invoke a new application if the first data field passed is used to identify the application script to be executed.)
- In general, for *converse-on* steps pass data to the VRU, ensure the VRU script does not execute any commands between the time the call is answered and the time when the first *prompt and collect* command is executed.

VRU Data Collection

When digits are passed from the DEFINITY System to the VRU, the first VRU script commands executed are *answer phone* and *prompt and collect*. No announcement is programmed for the *prompt and collect* command, and the

pound sign (#) is programmed as the end-of-string sign. If two sets of digits (that is, <data_1> and <data_2>) are passed by the DEFINITY System, there will be two *prompt and collect* commands on the VRU to receive them.

If the first digit string (<data_1>) passed to the VRU is for application selection, the CONVERSANT Script Builder *exec* command invokes the appropriate script. If a second digit string (<data_2>) is also used to pass an argument to this selected application, the first command in the *exec*'ed script is a *prompt and collect* command with no announcement prompt programmed and with the pound sign (#) programmed as the end-of-string character.

The "Converse second data delay" is used to give the VRU time to invoke the selected application before the <data_2> digit string is outpulsed.

The application developer should ensure the administered *converse first data delay* and *converse second data delay* timers allow sufficient time for the VRU to successfully collect all outpulsed digits, even during periods of heavy call volume. Loss of digits from <data_2> is an indication the converse second data delay timer needs to be increased.

Script Execution

During script execution, digits input by the calling party in response to *prompt and collect* commands are collected by the VRU but are not collected by the DEFINITY System as dial-ahead digits. Also, audible feedback is determined by the VRU.

If an agent from a nonconverse split becomes available to service the call while the VRU script is being executed, the VRU port is dropped from the call, and the caller is immediately connected to the agent. Any digits collected prior to executing the *converse-on* step are still available and may be displayed using the CALLR-INFO button.

The entire call is dropped if the caller abandons during the execution of a *converse-on* step.

Data Return

This phase is optional and is in effect only if the application calls for the VRU to return information to the DEFINITY System before returning control to vector processing.

Digits returned by the VRU are treated as dial-ahead digits. The rules for collecting and processing VRU-returned digits are identical to those for collecting and processing Call Prompting digits (see Chapter 4, "Call Prompting").

VRU data return is done in a manner similar to an analog transfer. Specifically, the VRU does an analog switchhook flash, outpulses DTMF digits, and then hangs up. If converse data is returned, the DTMF digits comprise two parts. The first sequence of digits is the converse data return feature access code administered on the Feature-Access-Codes form. The second sequence of digits is the sequence to be passed by the VRU. These digits are collected later during vector processing.

The CONVERSANT VRU offers a built-in external function called “converse_ data.” This function allows applications developers to perform this operation in a convenient and robust fashion.

To ensure the robust operation of the VRU data return operation, be sure to follow these recommendations:

- Set the analog flash timing to 600 msecs.
- Ensure DTMF tones last at least 70 msec and interdigit pauses last at least 50 msec. This results in an outpulsing rate up to 8.33 digits per second.
- (CONVERSANT only) Use the “converse_data” external function to return data to the DEFINITY System.
- Hang up line to DEFINITY after outpulsing digits. Assume that DEFINITY will wait between 1.2 and 1.5 secs to determine that the hang-up is a disconnect.

For applications involving VRUs other than CONVERSANT VRUs, be sure to follow these recommendations:

- After the flash, ensure the VRU performs dialtone detection (stutter dialtone) for a sufficient period of time to ensure accurate detection (typically 0.6 to 1.0 secs) before outpulsing the converse data return feature access code.
- If no dialtone is received before the timeout, ensure the VRU does two more retries of the analog flash. Also, if no dialtone is detected after two retries, ensure the VRU logs an error.
- Whenever dialtone is detected, ensure the digits of the converse data return feature access code are outpulsed.
- After the converse data return feature access code is outpulsed, the returned digits can be outpulsed without waiting for the second dial tone.
- After the VRU digits are outpulsed, the line to the DEFINITY is dropped.

Assuming an outpulse rate of 8 digits per sec (0.125 secs per digit), a 3-digit feature access code and stutter dial tone detection time of 0.6 secs, the maximum of 24 digits passed to DEFINITY should take about 6 secs (1.2 secs disconnect plus 8 secs plus 0.125 secs per digit).

The TN744 Call Classifiers required by the Call Prompting feature are not required for returning digits in-band from the VRU to the DEFINITY System.

Instead, general purpose TTR boards (TN748s) are used. As long as dial-ahead digits are available, any *collect digits* steps following a *converse-on* step do not require a TN744 to be allocated to the call.

If no general purpose TTRs are immediately available, and if the call queues for a TTR, no dial tone is provided. For this scenario, the VRU does not outpulse any digits until a TTR is available and dial tone is provided.

If there are no general purpose TTRs available on the DEFINITY System, and if there is no space in the TTR queue, the operation fails. Usually, the VRU logs an error and then quits, and vector processing continues at the next vector step. Existing system measurements reports indicate when the system is configured with an insufficient number of TTRs.

The “Converse Data Return Code” can be followed by a maximum of 24 digits. The VRU touch-tones the code and the digits in-band. However, the code and the digits are not heard by the caller. The digits are stored in the DEFINITY System as Call Prompting dial-ahead digits. If “x” digits are collected by vector processing before the *converse-on* step is executed, the maximum number of digits that can be returned is reduced to “24-x.” Any additional digits returned by the VRU are discarded. The data return is completed once the VRU hangs up.

The digit string returned by the VRU can consist of the digits (0 through 9) and pound signs (#). The pound sign (#) is interpreted by the *collect digits* step as an end-of-string character. If the digit string being returned is of variable length, the VRU can terminate the string with a pound sign (#) to avoid the ten second timeout delay that occurs when the digits are collected. If the digit string being returned is “multi-part” (that is, to be collected by multiple *collect digits* steps), and if some of the parts are of variable length, the pound sign (#) can be used to terminate each of the variable length parts.

NOTE:
An asterisk (*) may be included as part of the converse data return code. However, since the asterisk is interpreted as a “delete” character by the DEFINITY System, it makes little sense to use it as a returned digit. If it is used as such, all characters returned prior to the asterisk are discarded.

During the data return phase, the caller is temporarily put on hold. Music-on-hold, if administered, is suppressed. Since the caller hears silence during this phase, feedback should be provided to the caller as soon as possible after the *converse-on* step is executed.

Any touch-tone digits dialed by the calling party during the data return phase are discarded. These digits do not cause data corruption, and they are not collected as dial-ahead digits by the DEFINITY System.

If an interdigit timeout occurs during the data return phase, the switch logs a vector event, keeps the digits already returned, drops the VRU, and reactivates vector processing at the next vector step.

If the timeout occurs before the converse data return code is returned, the operation is the same except that no discarded digits will be available.

Script Completion

The VRU script returns control to vector processing on the DEFINITY System by simply hanging up the line. In cases where no data is returned to the DEFINITY System, this is done usually by executing the *quit* command. In cases where data is returned, this occurs whenever the VRU hangs up on completion of the VRU data return operation.

The last set of digits collected before the *converse-on split* step is executed is still available and may be displayed by an answering agent on the nonconverse split by using the CALLR-INFO button.

A VRU script can be programmed to continue running after hanging up the voice line. This after-call work is usually very short, and it may involve either a final message to a host or a final update to a local database. For this scenario, the VRU port (channel) is still associated with the running script even though there is no longer a voice connection.

From the DEFINITY System's point of view, the agent (port) is available for the next call. If a call is delivered to this port, the VRU does not answer the call until the previous script has completed. As long as the VRU script's after call work is short in duration, this poses no significant problem for the VRI feature. However, high volume VRI applications with lengthy after call work periods should be avoided, especially if such periods are so lengthy they approach the administered timeout period on the DEFINITY System for the Redirection on No Answer (RONA) feature. In such a case, RONA might think the VRU ports are faulty and might therefore start to take these ports out of service.

DEFINITY System Data Collection

NOTE:
This phase is in effect only if the VRU returns information to the DEFINITY System.

Once the VRU script has completed and vector processing is reactivated, the returned digits are collected and processed by vector commands in the usual manner. Since the digits must be collected by a *collect digits* command, data may be returned and processed only if the Call Prompting option is enabled.

The data returned can consist of multiple parts. For example, the VRU could return a stream of seven digits in which a single digit success/fail code is followed by a six-digit account code. For this scenario, the *converse-on* step would be followed by a sequence of vector steps including two *collect digits* steps. The first *collect digits* step would collect one digit and then check the

result code; the second *collect digits* step would collect the six-digit account code.

Any touch-tone digits dialed by the calling party during the data collection phase are discarded, do not cause data corruption, and are not collected as dial-ahead digits by the DEFINITY System.

If VRU data is returned, the calling party is able to touch-tone a response to a DEFINITY System prompt only after the data collection phase is completed and another *collect digits* step is executed. This is true because each executed *collect digits* step does not allocate a TTR when dial-ahead digits are present. Since VRU-returned digits are treated as dial-ahead digits, a TTR is attached to the call only after all returned digits are collected and another *collect digits* step is encountered. Only at this point can the caller hear an announcement for the *collect digits* command and successfully enter digits.

Security Issues

J

Introduction

There are a number of security issues vis-a-vis Call Vectoring that should be noted. For example, Call Vectoring and Call Prompting can be used to help prevent unauthorized users from gaining access to the switch via the Remote Access feature. This appendix explains how this is done.

NOTE:
For more information on security issues, refer to the *GBCS Products Security Handbook*, 555-025-600.

Remote Access

Abuse of remote access on the switch is one of the main methods by which unauthorized users obtain telephone services illegally. This section explains how a number of Call Vectoring features can be used to prevent unauthorized use of the remote access feature. No new development is required for any of these services.

Two methods are available, as follows:

- Front-ending remote access (that is, reaching the remote access extension via Call Vectoring).
- Replacing the function of the remote access extension by one or more call vectors.

Front-Ending Remote Access

Via this method, authorized external callers are given a VDN extension to call instead of the remote access extension, which is kept private. The corresponding call vector can then implement a number of security checks before routing callers to the remote access extension. Routing can be done via a *route-to number* or *route-to digits* step.

The following advantages are possible via this method.

- Call Vectoring can introduce a delay before the dial-tone is provided to the caller. Immediate dial-tone is often one criterion searched for by a hacker's programs when the hacker is trying to break into a system.
- A recorded announcement declaring that the use of the switch services by unauthorized callers is illegal and that the call is subject to monitoring and/or recording can be played for the caller.
- Call Prompting can be used to prompt for a password. In such a case, the call is routed only if there is a match on the password.
- Use of the remote access extension can be limited to certain times of the day or certain days of the week.
- Real-time and historical reports on the use of the remote access feature can be accessed from BCMS and/or CMS.
- Different passwords can be used on different days of the week or at different times during the day.
- Many VDNs that call the remote access extension can be identified. Accordingly, individuals or groups can be given their own VDN with unique passwords, permissions and reports. Any abuse of the system or security leak can then be attributed to an individual or a group.
- The caller can be routed to a VRU using the *converse-on* step where more sophisticated security checking, such as speaker recognition, can take place.
- Anyone failing any of the security checks can be routed to a "security" VDN that routes the caller to security personnel with a display set or to a VRU. Such a call would show "security" and possibly also the attempted password on the display. If the call is passed to a VRU, the VDN, the ANI and/or the prompted digits can be captured. BCMS/CMS reports on this security violation VDN will give information on how often and when security violations occur.

Replacing Remote Access

For this method, the remote access extension is not used. One or more VDNs are designed to access call vectors that can employ all of the security checks described in the previous section. The same reports and monitoring/recording

capabilities described in the previous section can also be used. Instead of routing to the remote access extension, the vector collects digits from the caller and then routes to the given destination if there is a match on the password.

Again, multiple VDNs can be created for individuals or groups with different security checks and different permissions and/or restrictions. Destination numbers provided by callers can be screened by the vectors and denied if the user does not have permission to access that destination. For example, an individual user could be restricted to placing calls to numbers beginning with area codes “303” and “908.”

EAS

With EAS, agents’ voice terminals can be locked when the terminal is not staffed. This is accomplished by assigning the voice terminals a Class of Restriction that does not allow outbound calls or it could be restricted from toll calls.

EAS agents have an optional password of up to nine digits to log in. This password is not displayed on DCP terminals when the agent is entering the password on the dial pad.

Limiting Outside Access Using VDN COR Restrictions

A VDN has a Class Of Restriction (COR). Calls processed by the vector carry the permissions and restrictions associated with the COR of the Vector Directory Number.

For example, if a vector in the switch is written to collect digits, and then to route to the digits dialed, the restrictions on what calls can be placed are determined by the COR of the latest VDN. Also, checks can be made on the digits that are dialed, using *goto _ if digits* vector commands (for example, *goto _ if digits* in table) to disallow routing to undesired destinations. The *collect digits* step can also be limited to collect only the number of digits required (for example, only collecting five digits for internal dialing).

An incoming caller can access Trunk Access Codes, some Feature Access Codes, or most other sets of dialed digits. In order to deny incoming callers access to outgoing facility paths, the COR of the Vector Directory Number must be configured to disallow outgoing access. This should include; lowering the Facility Restriction Level in the COR to the lowest acceptable value (FRL=0 provides the most restricted access to network routing preferences), assigning a Calling Party Restriction of "Toll" or "Outward", denying Facility Test Call capability, and blocking access to specific COR’s assigned to outgoing Trunk Groups using the Calling Permissions section of the Class of Restriction Screen.

Review the Classes of Restriction assigned to your VDNs. If they are not restricted, consider assigning restrictions on the VDN and/or using *goto* tests on those digits to prevent callers exiting the system via the vector.

Vector Initiated Service Observing

The following restrictions can be used with vector initiated Service Observing to guard against unauthorized use.

- Call prompting commands can be used in Service Observing vectors to provide passcode protection, and to limit access to observing specific destinations or verified caller entered digits.
- Time of Day/Day of Week checks can be incorporated in Service Observing vectors.
- A vector can be created to be used exclusively for Service Observing.
- For a VDN to be observed as the result of a route-to command, the VDN must have a COR that allows it to be observed.
- The calling permissions of the COR assigned to the Service Observing VDN in conjunction with the “can be observed” settings of the COR assigned to the destination determine what agents, stations, or VDNS can be observed.

Voice Response Integration

When a converse step is used to access a VRU application that returns data for a collect digits step, the opportunity for toll fraud exists when the VRU application fails to return any data. To avoid this type of toll fraud be certain that one of the following is true:

- If the collected digits are used to route calls internally, be certain that the Class of Restriction (COR) for the Vector Directory Number (VDN) does not allow calls to route externally.
- If it is necessary to use the collected digits to route calls externally, use a password to verify that the collected digits have been passed by the VRU application. For example, in the following vector Screen J-1 the VRU application returns a three-digit password followed by the eight-digit external number. The vector routes calls without the correct password to a different vector and routes calls with the correct password to the collected digits.

```
converse-on split 10 pri m passing none and none
collect 3 digits after announcement none
goto vector 23 if digits <> 234
collect 8 digits after announcement none
route-to digits with coverage n
```

Screen J-1. Voice Response Integration Security Example

Setting Up a Call Center

K

Introduction

Call center managers need some key indicators to measure ACD performance at their site. Usually, in setting up a call center, several factors involving call management are considered. The following list identifies and defines the most common of these factors, and it provides a typical question that might be asked. In addition, an insurance company example will be used to discuss the different options in this chapter.

- **Volume**

  Number of calls going in or out of the ACD. (How many calls did Split 1 answer?)

- **Productivity**

  Call volume per unit of time. (How many calls did Split 1 answer between 8 a.m. and 9 a.m.?)

- **Utilization**

  Overall use of the phone center. (What was my agent occupancy?)

- **Accessibility**:

  Availability of lines and agents when customers call the ACD (this is an area that CMS can probably most clearly define and help improve). (Were lines busy when customers called or did they have to wait too long?)

- **Quality of Service**:

  Accuracy of information, a pleasant manner, responsiveness to caller concerns, successful completion of business, and efficient time utilization (not all measured directly by CMS). (Was the caller given good service?)

This chapter explains how to set up a call center for customers with Call Vectoring and/or Expert Agent Selection (EAS).

Call Vectoring/Non-EAS Option

To set up a call center that has Call Vectoring but not EAS, do the following:

1. Determine your call center's objectives. Think about how you want your call center to handle calls and also about what you want your call center to achieve. See Non-EAS Worksheet #1.

   A company's basic goals are to increase profits and market share and to decrease costs. The purpose of setting up a call center is to monitor these goals using CMS reports. It is best to have more than one objective. (Some customers set and then live by only one objective.) Call center objectives must then be created to meet the goals. These objectives must be communicated to the Split Supervisor or to the Administrator managing the call center.

   The following list provides an example set of call center objectives:

   - Establish the following measured entities:
     - Average Speed of Answer = 15 seconds
     - Abandon Rate <= 3%
     - Average Talk Time = 2 1/2 minutes
     - ACD calls per agent = 80 to 90 per day
     - Number of calls in queue = 6
     - Percentage of calls answered within the service level = 95%
     - Agent occupancy > 90%
     - Percentage of trunks busy < 3%
   - Generate revenue through the call center.
   - Train agents to back up each other.
   - Adequately train agents to provide service that meets customer expectations.
2. Review your existing operation and determine your customer/call center needs (see Non-EAS Worksheet #2: Current Split Operation Worksheet and Table K-1).
3. On the switch, assign a unique Hunt Group number and Call Distribution method to each caller need. This number will be your split number (see Non-EAS Worksheet #3: Customer Needs Worksheet and Table K-1).
4. Assign DNIS (Dialed Number Identification Service) (that is, the number dialed) as a Vector Directory Number (VDN) (see Table K-1).

As an option, you can assign one VDN for a main number and use Call Prompting to route the call to the proper split.

Table K-1 illustrates the guidelines given up to this point.

Table K-1. Customer/Call Center Needs Guidelines

Customer/Call Center Needs	Split Number (Hunt Group)	Call Distribution[1]	VDN
New policy	1	UCD	555-6543
Questions about policy, Rate Quotes, Billing	2	UCD	555-6432
Spanish speaking for policy, service, and claims	3	DDC	555-6321
Claims	4	UCD	555-6210

1. Options include Direct Department Calling (DDC) and Uniform Call Distribution (UCD).

Notice that this call center has only one split for all Spanish calls. However, resources permitting, you could create a New Policy split, a Service split, and a Claims split, each containing agents who speak Spanish. As an alternative, you could use one main VDN to point to a Call Prompting vector designed to route the calls to the splits.

5. On the switch, assign extensions to the agents' physical terminal locations (see Table K-2).
6. In CMS: Dictionary: Login Identifications, assign each agent a unique loginID (see Table K-2).

   Agents are known to CMS by the login ID. If assigned, reports refer to an agent by name, not by l1ogin ID.

Table K-2 illustrates the assignments described in the previous two items:

Table K-2. Extension/LoginID Assignments

Agent Name[1]	Extension	LoginID[1]
Randy Tyler	1231	2000
Cathy Smith	1232	2001
Carla Silva	1238	2002

1. = assigned in CMS Dictionary

NOTE:
When you are adding names to extensions on the switch, the agent name should be the same name as the loginID assigned to CMS.

7. On the switch, assign agent extensions to splits (see Table K-3).

   More than four splits can be assigned to an agent; however, the agent can log into a maximum of four splits. An agent assignment to splits can be changed in CMS: ACD Administration: Move Extensions Between Splits if the agent is logged off.

   Table K-3 illustrates the assignment of agent extensions to splits:

Table K-3. Agent Extension/Split Assignments

Split (Hunt Group)	Agent Extensions
1 - Sales	1231, 1232, 1233, 1234, 1235, 1236, 1237, 1238, 1239
2 - Service	1231, 1232, 1234, 1238, 1239, 1240
3 - Spanish	1238, 1240, 1245
4 - Claims	1238, 1239, 1240, 1241, 1242

8. On the switch or in CMS: ACD Administration: VDN Assignments, assign a vector to each VDN (see Table K-4).

   A VDN can point to only one vector. However, a vector can have more than one VDN pointing to it.

Table K-4 illustrates VDN/vector assignments.

Table K-4. VDN/Vector Assignments

VDN	Vector
6543	1 (Sales)
6432	2 (Service)
6321	3 (Spanish)
6210	4 (Claims)

9. On the switch or in CMS: ACD Administration: Vector Contents, write your vectors. See Non-EAS Worksheet #4: Vector Design Worksheet.

   Your vectors should match your call center objectives. To meet these objectives, you must make a number of relevant decisions (for example, you may decide how soon you want to enlarge an agent pool or what kind of treatment the caller should receive). If your VDN and vector reports do not satisfy your call center objectives, you must consider your alternatives (for example, you may deem it necessary to train agents or to increase the amount of time elapsed from when a call queues to one split and then to another split).

   The following lists indicate the actions produced by two different vectors:

   Actions Produced by Vector #1:

   1. Tell the caller to select one of the following prompts:
      - 1 = Sales
      - 2 = Service
      - 3 = Spanish
      - 4 = Claims
      - Nothing or 0 = Service
   2. Queue the call.
   3. Provide an announcement to the caller.

10. **Actions Produced by Vector #2:**

    a. Queue the call to the correct service at a medium priority.

    b. If no agents are available, provide a message and then play music.

    c. If the call is not answered within 10 seconds, provide a second message and then play music.

    d. If the call is not answered within 7 more seconds, queue the call to the Service split.

e. If the call is not answered within 7 more seconds, queue the call to the Spanish split at a high priority.

NOTE:
A *check split* command queues the call to up to three splits if the conditions are met. If the conditions are not met, the *check split* command may not get read again (if the vector step in which it appears is not executed again).

11. In CMS: Dictionary, assign names to the splits, VDNs, and vectors.
12. Once your system is up and operational, you will need to monitor it to ensure you are meeting your call center objectives. CMS can be used to monitor many of your objectives. Some objectives will need to be monitored and have adjustments made in real time. For example, if the number of calls waiting, average speed of answer, or percent answered within a service level is not meeting your objectives, you might want to immediately move some agents, direct calls to another vector, or look-ahead interflow some calls. Other items such as agent occupancy and percent all trunks busy may only need to be monitored daily to look for trends.

Non-EAS Worksheet #1
Call Center Objectives Worksheet

What Are My Call Center Objectives?

Figure K-1. Non-EAS Worksheet #1: Call Center Objectives Worksheet

Non-EAS Worksheet #2
Current Split Operation Worksheet

Split _____

Primary Backup _____ **Secondary Backup** _____ **Tertiary Backup** _____

List Individually You Customer/ Caller Needs and Your Agent Skill Sets in this Split	Do You Have Agent Expertise? (Yes/No)	Do You Want to Separate Skill Set with EAS? (Yes/No)
1.		
2.		
3.		
4.		
5.		
6.		

Split _____

Primary Backup _____ **Secondary Backup** _____ **Tertiary Backup** _____

List Individually You Customer/ Caller Needs and Your Agent Skill Sets in this Split	Do You Have Agent Expertise? (Yes/No)	Do You Want to Separate Skill Set with EAS? (Yes/No)
1.		
2.		
3.		
4.		
5.		
6.		

Figure K-2. Non-EAS Worksheet #2: Current Split Operation Worksheet

Non-EAS Worksheet #3
Customer Needs Worksheet

Customer/Call Center Needs	Split Number (Hunt Group)	Call Distribution	VDN

Figure K-3. Non-EAS Worksheet #3: Customer Needs Worksheet

Vector # _____ **Name** _____ **Description** _____

Assigned VDNs _____ **Assigned Trunk Groups** _____

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____
8. _____
9. _____
10. _____
11. _____
12. _____
13. _____
14. _____
15. _____
16. _____
17. _____
18. _____
19. _____
20. _____
21. _____
22. _____
23. _____
24. _____
25. _____
26. _____
27. _____
28. _____
29. _____
30. _____
31. _____
32. _____

Figure K-4. Non-EAS Worksheet #4: Vector Design Worksheet

Call Vectoring/ EAS Option

To set up a call center that has both Call Vectoring and EAS, do the following:

1. Determine your call center's objectives. Think about how you want your call center to handle calls and also about what you want your call center to achieve. See EAS Worksheet #1: Call Center Objectives Worksheet.

   A company's basic goals are to increase profits and market share and to decrease costs. The purpose of setting up a call center is to monitor these goals using CMS reports. It is best to have more than one objective. (Some customers set and then live by only one objective.) Call center objectives must then be created to meet the goals. These objectives must be communicated to the Split Supervisor or to the Administrator managing the call center.

   The following list provides an example set of call center objectives:

   - Establish the following measured entities:
     - Average Speed of Answer = 15 seconds
     - Abandon Rate <= 3%
     - Average Talk Time = 2 1/2 minutes
     - Expected Wait Time < 30 seconds
     - ACD calls per agent = 80 to 90 per day
     - Number of calls in queue = 6
     - Percentage of calls answered within the service level = 95%
     - Agent occupancy > 90%
     - Percentage of trunks busy < 3%
   - Generate revenue through the call center
   - Train agents to back up each other
   - Adequately train agents to provide service that meets customer expectations

2. Review your existing operation and determine your customer/call center needs and your business needs. Also, determine if all of these needs require skills. See EAS Worksheet #2: Current Split Operation Worksheet.

   The call center could have many more skills than those indicated in Table K-5. However, we'll assume that the insurance company represented in the table services only certain states on the West Coast. The point is that EAS allows you to expand on your customer/call center needs. An example of a business need is a temporary promotion.

3. Assign a skill hunt group and call distribution method to each set of needs. See EAS Worksheet #3: Customer needs Worksheet.

   With EAS, most hunt groups tend to use EAD to allow callers to reach the most expert agent. However, UCD might be used if all agents assigned a skill are equally trained and if you want equal call distribution to the agents.

4. In CMS: Dictionary: Splits/Skills, assign names to the skills.

   Table K-5 illustrates the guidelines provided up to this point.

Table K-5. Guidelines

Customer/Call Center Needs	Skill Name	Skill Number	UCD/ EAD	COR
New policy — West	Sales West	1	EAD	
New policy — West/Coast	Sales Coast	10	EAD	
New policy — West/OR	Sales OR	11	EAD	
New policy — West/CA	Sales CA	12	EAD	
New policy — West/AZ	Sales AZ	13	EAD	
New policy — West/Mountains	Sales Mountains	20	EAD	
New policy — West/CO	Sales CO	21	EAD	
New policy — West/TX	Sales TX	22	EAD	
New policy — West/NM	Sales NM	23	EAD	
Questions (Rate Quotes, Billing)	Service	30	EAD	
Questions (Rate Quotes, Billing)	Service Coast	31	EAD	
Questions (Rate Quotes, Billing)	Service Mountains	32	EAD	
Claims	Claims	2	EAD	
Spanish Speaking	Spanish	50	UCD	
Spanish Speaking Sales	Spanish Sales	51	EAD	
Spanish Speaking Service	Spanish Service	52	EAD	
Spanish Speaking Claims	Spanish Claims	53	EAD	

Continued on next page

Table K-5. Guidelines

Customer/Call Center Needs	Skill Name	Skill Number	UCD/ EAD	COR
Sales/Service	Sales/Service	3	EAD	
Service/Claims	Service/Claims	4	EAD	
Super Group/All Skills	Super	5	EAD	

5. On the switch, administer the VDNs. On the switch or in CMS: ACD Administration, change the VDN Skill Preferences and assign up to three skills to each VDN. See EAS Worksheet #6: Skill Preferences Worksheet. Then assign a VDN Skill Preference (either 1st, 2nd, or 3rd) to each VDN to establish which skills are “primary,” “secondary,” or “tertiary.”
6. On the switch or in CMS, assign a vector to each VDN.

   Table K-6 and Table K-7 illustrate a sample of the VDNs for the two previous numbered items.

Table K-6. Sample VDNs

Main VDNs	1st Skill	2nd Skill	3rd Skill	Vector
6543	Sales AZ	Sales Coast	Sales West	1
New policy AZ	13	10	1	
6432	Sales CA	Sales Coast	Sales West	1
New policy CA	12	10	1	
6321	Sales OR	Sales Coast	Sales West	1
New policy OR	11	10	1	
6210	Service Coast	Service/ Claims	Service/Claim	2
Questions Coast	31	30	4	
6123	Claims	Service/ Claims	Spanish Claims	2
Claims	2	4	53	

VDN 6234

The caller can enter a generic VDN and be prompted. Table K-7 illustrates prompting for Spanish callers, where callers would be prompted for type of service they require.

Table K-7. Sample Prompting

Prompting Digit	VDN Accessed From Vector	1st Skill	2nd Skill	3rd Skill	Vector
1	6651	Spanish Sales	Spanish		3
		51	50		
2	6652	Spanish Service	Spanish		3
		52	50		
3	6653	Spanish Claims	Spanish		3
		53	50		

The previous tables contain only a few examples of the VDNs (and, accordingly, the VDN Skill Preferences assigned to the VDNs) that the call center could assign. Be sure to determine the VDNs and VDN Skill Preferences (primary and backup skills) that you require. Also, be sure to determine if the VDNs can share a vector (as is illustrated byTable K-7) or if some VDNs require a separate vector (as illustrated by Table K-6). Table K-7 demonstrates that individual VDNs can be accessed from within a single vector. This is accomplished via Call Prompting digits.

Note that the switch link does not have to be taken down to assign skills and Skill Preferences. Also, if you change these assignments once they are administered, the changes take effect immediately, even if there are calls in queue. As a result, calls in queue can be assigned another VDN skill, and they can be queued to another skill hunt group.

7. Administer a dial plan that differentiates between physical extensions and loginIDs. Assign a loginID to each agent. Also, determine if the agent needs the Direct Agent feature. (Will agents be taking callbacks or transferring to each other, and do you want those to be treated as ACD calls?) If so, determine the login and logout coverage for each agent (which can be a VDN, a skill hunt group, or AUDIX). In CMS, assign to each agent a loginID (to enable the printing of the agent names on the reports). See EAS Worksheet #4: Individual Agent Skill Worksheet Generic 3.

Table K-8 illustrates dial plan administration:

Table K-8. Plan Administration

Agent Name	LoginID	Direct Agent	Login Coverage Pt1/Pt2/Pt3	Logout Coverage Pt1/Pt2/Pt3
Randy Tyler	2000	Yes	2/6543/AUDIX	AUDIX
Cathy Smith	2001	Yes	4/6012/AUDIX	AUDIX
Carla Silva	2002	No	--------	-------
Trish Carara	2003	No	--------	--------

The loginID assigned on the switch and in CMS to the agent is the Logical Agent ID. The ID name assigned in the switch should match the ID name assigned in the CMS Dictionary.

It is strongly recommended that an agent with Direct Agent status be assigned a “Direct Agent skill” as the primary skill (see Table K-9). This way, Direct Agent calls will not be sharing queue slots with other skill calls.

8. Determine which agents you want to answer calls in each skill hunt group. On the switch, assign skills to each agent, and assign each skill a skill level.

   If you want to give preference to some call types, assign the skill at a higher skill level. See EAS Worksheet #5: Agent Skills Worksheet Generic 3.

   Table K-9 illustrates skill status assignments:

Table K-9. Skill Status Assignments

Agent Name	Skill	Skill Assigned
Randy Tyler	12 — Sales CA	Level 1
	10 — Sales Coast	Level 2
	22 — Sales TX	Level 2
Cathy Smith	21 — Sales CO	Level 1
	23 — Sales NM	Level 1
	1 — Sales West	Level 2
Carla Silva	30 — Service	Level 1

Continued on next page

Table K-9. Skill Status Assignments

	4 — Claims	Level 2
	40 — Spanish	Level 2
Trish Carara	53 — Spanish Claims	Level 1
	4 — Claims	Level 1
	8 — Service/Claims	Level 2

NOTE:
Table K-9 indicates a small sample of agents in the call center.

Agent skill assignments can be modified from within CMS: ACD Administration: Change Agent Skills. The agent must log out (if he or she is already logged in) and then log back in for the changes to take effect.

9. On the switch or in CMS: ACD Administration: Vector Contents, write your vectors. See EAS Worksheet #7: Vector Design Worksheet.

   Your vectors should match your call center objectives. To meet these objectives, you must make a number of relevant decisions (for example, you must decide how soon you want to enlarge an agent pool or what kind of treatment the caller should receive). If your VDN and vector reports do not satisfy your call center objectives, you must consider your alternatives (for example, you may deem it necessary to train additional agents or to increase the amount of time elapsed from when a call queues to one skill hunt group and then to another skill hunt group).

   The following list indicates the actions produced by a vector:

   a. Queue the call to the 1st main skill hunt group (Sales).
   b. If no agents are available, provide a message and then play music.
   c. If the call is not answered within 10 seconds, provide a second message and then play music.
   d. If the call is not answered within 7 more seconds, queue the call to the 2nd main skill hunt group (Service).
   e. If the call is not answered within 7 more seconds, queue the call to the 3rd main skill hunt group (Claims).

10. In CMS: Dictionary, assign names to the skills, VDNs, vectors, and loginIDs.
11. Once your system is up and operational, you will need to monitor it to ensure you are meeting your call center objectives. CMS can be used to monitor many of your objectives. See for more details. Some objectives will need to be monitored and have adjustments made in real time. For

example, if the number of calls waiting, average speed of answer, or percent answered within a service level is not meeting your objectives, you might want to immediately move some agents, direct calls to another vector, or look-ahead interflow some calls. Other items such as agent occupancy and percent all trunks busy may only need to be monitored daily to look for trends.

EAS Worksheet #1
Call Center Objectives Worksheet

What Are My Call Center Objectives?

Figure K-5. EAS Worksheet #1: Call Center Objectives Worksheet

EAS Worksheet #2
Current Split Operation Worksheet

Split _____

Primary Backup _____ **Secondary Backup** _____ **Tertiary Backup** _____

List Individually You Customer/ Caller Needs and Your Agent Skill Sets in this Split	Do You Have Agent Expertise? (Yes/No)	Do You Want to Separate Skill Set with EAS? (Yes/No)
1.		
2.		
3.		
4.		
5.		
6.		

Split _____

Primary Backup _____ **Secondary Backup** _____ **Tertiary Backup** _____

List Individually You Customer/ Caller Needs and Your Agent Skill Sets in this Split	Do You Have Agent Expertise? (Yes/No)	Do You Want to Separate Skill Set with EAS? (Yes/No)
1.		
2.		
3.		
4.		
5.		
6.		

Figure K-6. EAS Worksheet #2: Current Split Operation Worksheet

EAS Worksheet #3
Customer Needs Worksheet Generic 3

Customer/Call Center Needs	Split Name	Split Number	UCD/EAD	Class of Restriction

When defining your skills, be aware of how you can combine skills for backup purposes.

Figure K-7. EAS Worksheet #3: Customer needs Worksheet

EAS Worksheet #4
Individual Agent Skill Worksheet Generic 3

Agent Name	Login ID	Direct Agent?	Class of Restrict	Logged-in Coverage			Logged-out Coverage		
				Point 1	Point 2	Point 3	Point 1	Point 2	Point 3

Figure K-8. EAS Worksheet #4: Individual Agent Skill Worksheet Generic 3

EAS Worksheet #5
Agent Skills Worksheet Generic 3

	Agentname	Login ID	First Skill (Skill Level)	Second Skill (SL)	Third Skill (SL)	Fourth Skill (SL)	Class of Restriction
1.							
2.							
3.							
4.							
5.							
6.							
7.							
8.							
9.							
10.							
11.							
12.							
13.							
14.							
15.							
16.							
17.							
18.							
19.							
20.							
21.							
22.							
23.							
24.							
25.							
26.							
27.							
28.							
29.							
30.							
31.							
32.							

Figure K-9. EAS Worksheet #5: Agent Skills Worksheet Generic 3

EAS Worksheet #6
VDN Skill Preferences Worksheet

VDN	VDN Name	Class of Restriction	Skill Preferences			Vector
			1st Skill Number	2nd Skill Number	3rd Skill Number	

Figure K-10. EAS Worksheet #6: Skill Preferences Worksheet

EAS Worksheet #7
Vector Design Worksheet

Vector # _____ **Name** _____ **Description** _____

Assigned VDNs _____ **Assigned Trunk Groups** _____

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____
8. _____
9. _____
10. _____
11. _____
12. _____
13. _____
14. _____
15. _____
16. _____
17. _____
18. _____
19. _____
20. _____
21. _____
22. _____
23. _____
24. _____
25. _____
26. _____
27. _____
28. _____
29. _____
30. _____
31. _____
32. _____

Figure K-11. EAS Worksheet #7: Vector Design Worksheet

Converting a Call Center to EAS

L

Introduction

The procedures in this Appendix provide guidelines for upgrading a call center from a non-EAS ACD environment to an EAS ACD environment. The primary activities involved in this conversion are:

- **Step 1:** Pre-EAS cutover administration for the switch
- **Step 2:** Pre-EAS cutover administration for CMS (if CMS is going to be used in the Call Center)
- **Step 3:** Pre-EAS cutover administration for AUDIX
- **Step 4:** Pre-EAS cutover administration for Messaging Server
- **Step 5:** Pre-EAS cutover administration for ASAI
- **Step 6:** EAS cutover

For more information on various approaches that may be used to cut over to EAS, refer to the CMS Administration documents.

Before the transition to EAS takes place, decisions must be made concerning:

- Which area of the current dial plan is to be used for EAS agent login IDs. EAS agent login IDs cannot conflict with already defined extension numbers (for example, an EAS agent login ID cannot be the same as a station extension number).
- Whether the current incoming call routing through VDNs and vectors will remain the same after the EAS upgrade, or whether new VDNs and/or vectors are required.
- How incoming call traffic is to be handled during EAS cutover.

Once these decisions are made, the pre-EAS cutover administration activities can be started in preparation for the conversion of the call center to EAS.

NOTE:
Even though EAS administration changes are being made, non-EAS ACD call handling and agent operations are unaffected. When the cutover to EAS is completed, all non-EAS ACD call handling and agent operations will cease.

Step 1: Pre-EAS Cutover Administration for the System

Perform the following activities to prepare the DEFINITY System for the cutover to EAS:

1. At administration terminal display the System-Parameters Customer-Options form and verify that the ACD, Expert Agent Selection?, and Vectoring (Basic) fields are set to y. If you will be using the increased capacities of EAS-PHD, verify that this option is set to y.
2. If you haven't already done so, display the Feature Access Code form and administer the ACD Agent Feature Access Codes (for example, "Login," "Logout," and "Auto-In") as required for call center agent operations.
3. Using the CDR System Parameters form, administer whether the EAS login ID, or the terminal extension where the EAS agent is logged in, should appear on CDR reports by setting the Agent Login ID - Record? field to y or n, respectively. This field affects the CDR tracking for incoming calls only; outgoing calls made by a logged-in EAS agent are always recorded by CDR using the agent's login ID.
4. If new VDNs are desired for the EAS environment, using the VDN administration form, administer the VDN Skills and other VDN information for the VDNs used to route calls to EAS agents. If the "1st," "2nd," and/or "3rd" skill options are to be used in the vectors or for CMS tracking associated with these VDNs, then administer the 1st Skill, 2nd Skill, and 3rd Skill fields as required.
5. If new vectors are desired for the EAS environment, using the Vector administration form, administer the vectors associated with the VDNs added in the previous step. As part of the EAS feature, the "1st," "2nd," or "3rd" skill options may be used in the vector step fields where a skill hunt group is entered (rather than entering an absolute skill hunt group number). Refer to Chapter 11, "Expert Agent Selection" for more information concerning vector programming for the EAS feature.

6. If new skill hunt groups are required, using the Hunt Group administration form, administer the desired skill hunt groups.

   NOTE:
   Entering a y in the Skills? field automatically causes the ACD? and Vector fields to be set to y. With EAS optioned, it is not possible to administer members for a skill hunt group.

7. If coverage paths are to be administered for EAS agents, using the Coverage Path administration form, set up the coverage paths to be assigned to EAS agent login IDs.

   NOTE:
   There is a difference between coverage treatment for an EAS "Direct Agent" call (where both the calling party and called login ID have the Direct Agent Calling COR option set to y), and an EAS "personal" call (where either the calling party or called login ID does not have the Direct Agent Calling COR option set to y).

   NOTE:
   A Direct Agent call is routed to an EAS agent as an ACD-type call, and therefore its coverage behavior is considerably different from the coverage for a normal station call. For example, if an EAS agent is not available for an ACD call when a Direct Agent call is made to that agent, the Direct Agent call is queued to the Direct Agent Skill administered on the Agent Login ID form (after initiating a ring-ping and then fluttering the active work-mode button at the agent's terminal). On the other hand, a personal call to an EAS agent is not an ACD-type call, and its coverage behavior is similar to the coverage treatment for a call to a station extension. For example, a personal call to an EAS agent who is busy on any call appearance will result in the call being sent to an idle call appearance at that agent's terminal.

   Depending on the type of coverage criteria desired for Direct Agent and personal calls to EAS login IDs, administer the desired coverage path criteria as follows:

   - To provide coverage for a non-ACD "personal" call to an EAS login ID when the agent is logged in and active on any call appearance, set the Active? coverage criteria to y. The Active? coverage criteria does not apply for a Direct Agent call to an EAS login ID.
   - To provide coverage for calls to an EAS login ID when the agent is logged out, set the Busy? coverage criteria to y. Busy coverage will also be applied to a logged-in EAS agent when:
     — A Direct Agent call is made to the EAS agent and there are no available queue slots in the agent's first skill hunt group;

     or

— A personal call is made to an EAS agent and the agent's station has no idle call appearances.

- To provide coverage for calls to an EAS login ID when the agent is logged in but does not answer after a certain number of ring cycles, set the Don't Answer? coverage criteria to y, and enter a number for the desired ring time-out in the Number of Rings field.
- To provide immediate coverage for calls to an EAS login ID whether the agent is logged in or logged out, set the All? coverage criteria to y.
- To provide coverage for calls to EAS login IDs when the call is to a logged-in agent who has activated the Send All Calls or Go To Cover features, set the DND/SAC/Goto Cover? coverage criteria to y.

8. Up to three coverage paths for different types of call coverage criteria may be linked together by administering the Next Path Number field on the Coverage Path form. If the criteria for the first coverage path are not met, then the criteria for the second linked coverage path are checked by the system, and so on. This can be used to provide different coverage paths for calls to an EAS login ID when the associated agent is logged in or logged out.

NOTE:
If a call to a logged-in EAS login ID is a "personal" call and coverage goes into effect, the redirected call maintains a "simulated bridged appearance" at that agent's terminal. The agent may still answer the call after redirection takes place by going off-hook on this line appearance. However, if a call to a logged-in EAS login ID is a Direct Agent call, the redirected call does not maintain a simulated bridged appearance at the agent's terminal. The agent may not then answer the call after redirection takes place.

NOTE:
If the Redirection on No Answer (RONA) feature is enabled for skill hunt groups, set the ring time-out interval for the RONA feature such that it does not conflict with the coverage ring time-out criteria.

9. If coverage paths are administered for EAS login IDs, using the Feature-Related System Parameters form, set the Coverage - Subsequent Redirection No Answer Interval field to the desired ring time-out interval for calls routed to administered coverage points.

NOTE:
EAS login IDs may be administered as coverage points for a coverage path, and this administered coverage no-answer interval applies to Direct Agent or "personal" calls made to these coverage points as well.

10. Using the COR administration form, set the Direct Agent Calling field to y for any COR to be assigned to a trunk or station user who may initiate a Direct Agent call to an EAS agent, or to be assigned to an EAS login ID that may receive Direct Agent calls.

11. If EAS agent login ID passwords are to be administered, using the Feature-Related System Parameters form, set the Minimum Agent-LoginID Password Length field to the desired number of minimum password digits (0 to 9) which must be specified when agent passwords are administered via the Agent Login ID form. The total number of digits which may be assigned to a password is between the value of the Minimum Agent-Login ID Password Length field and 9 digits. If a password is administered for an agent, this password must be entered in addition to the agent's login ID to log in.

12. Using the Agent Login ID form, add the desired EAS login IDs to be associated with human agents, AUDIX ports, and/or AAS (Auto-Available Split) VRU ports. For human agents, the following fields are administered:

    - Name
    - COR
    - Coverage Path (optional)
    - Security Code (optional for Demand Print feature)
    - LWC Reception (optional)
    - AUDIX Name for Messaging (for MIPS only, if the LWC Reception field is set to audix, or if administered coverage path for the agent has an AUDIX coverage point)
    - Messaging Server Name for Messaging (for MIPS only, if the LWC Reception field is set to msa-spe, or if administered coverage path for the agent has a Messaging Server coverage point)
    - Password (optional)
    - Skills - Skill Level (for at least one skill)

13. For AUDIX and AAS VRU port extensions, when these ports are associated with ACD-type hunt groups, these extensions must be associated with skill hunt groups as part of the cutover to EAS. Additionally, for skill hunt groups used for AAS ports, the AAS? field must be set to y for these hunt groups before any EAS AAS agents can be administered.

    NOTE:
    AUDIX hunt groups do not need to be vector-controlled. This allows for ASAI monitoring of the skill hunt group.

    If AUDIX port extensions (such as for the Embedded AUDIX product) are not associated with an ACD hunt group, no administration is required for these ports as part of the cutover to EAS. For the AUDIX and/or AAS ports that are associated with ACD hunt groups, add EAS agent login IDs for these ports, where only the following fields need to be administered:

    - Name
    - COR
    - Coverage path (optional)
    - AUDIX? (set to y for AUDIX ports)
    - AAS? (set to y for AAS VRU ports)
    - Port Extension (set to the AUDIX or AAS port extension administered in the non-EAS environment)
    - Skills - Skill Level (where a single skill is entered for the skill hunt group associated with the AUDIX or AAS station ports)

14. Using the Station Administration form, administer any stations to be used by EAS agents and the desired work-mode buttons for each station (if not already administered).

    NOTE:
    If stations are already administered with work-mode buttons associated with splits, it is not necessary to readminister these buttons for EAS. If new work-mode buttons are added to a station, it is not possible to enter data in the Grp field after EAS is enabled except for the AUX work-mode button (which may be administered with a hunt group number if the entered hunt group is a non-ACD hunt group).

    NOTE:
    Also, if more than one set of work-mode buttons is administered on a station set, these buttons may be left as is until after the cutover to EAS. After the cutover, it is desirable to remove the extra sets of work-mode buttons since EAS requires only one set of work-mode buttons for agent operations.

Step 2: Pre-EAS Cutover Administration for CMS

If R3V2 or later release CMS is to be used at the EAS customer site, refer to the appropriate CMS Administration document for the procedures used to configure CMS for the EAS feature. This document is also helpful in providing overall planning strategies for implementing call center operations.

Step 3: Pre-EAS Cutover Administration for AUDIX

If EAS agents' login IDs are administered with coverage paths that route to an AUDIX coverage point, the login IDs for these agents must be administered via the AUDIX console so that the caller will hear the appropriate AUDIX voice responses for calls made to EAS login IDs.

NOTE:
On the MIPS, the AUDIX Name for Messaging field on the Agent Login ID form must be set to the correct AUDIX name to provide proper AUDIX coverage of calls made to EAS agents, or to leave LWC messages for EAS agents if LWC reception to AUDIX is set up for the agents' login IDs.

Refer to Step 1: "Pre-EAS Cutover Administration for the System" for information on how to administer EAS login IDs for AUDIX port extensions on the switch.

Step 4: Pre-EAS Cutover Administration for Messaging Server

If EAS agents are administered with coverage paths that route to a Messaging Server coverage point, the Messaging Server adjunct must be administered with extension information that correlates to these EAS login IDs.

NOTE:
The Messaging Server Name for Messaging field on the Agent Login ID form must be administered to provide proper Messaging Server coverage of calls made to EAS agents, or to leave LWC messages for EAS agents if LWC reception to AUDIX is set up for the agents' login IDs.

Step 5: Pre-EAS Cutover Administration for ASAI

With ASAI-based applications for call center operations, the cutover to EAS may necessitate an upgrade of the ASAI-related application software on the adjunct. With OCM (Outgoing Call Management), the upgrade to EAS requires that specialized vectors be administered to handle the launching of calls from VDNs (as opposed to the non-EAS environment where OCM calls are launched from splits). For more information on the procedures to convert an ASAI application for EAS, refer to *DEFINITY Enterprise Communications Server CallVisor ASAI Planning*.

Step 6: EAS Cutover

After all pre-EAS activities have been completed, the EAS feature may be activated. Just prior to the EAS cutover, a tape backup of the current switch translations should be made for possible recovery purposes in case some difficulty is encountered during cutover. In particular, since the transition to EAS results in the removal of all ACD hunt group members, the pre-EAS tape backup could save a considerable amount of time in restoring non-EAS hunt group translations if the cutover to EAS is not completed.

It is recommended that incoming ACD call traffic be blocked to prevent the queuing of new ACD calls to existing splits during the cutover from the non-EAS to EAS environment. Blocking of new incoming calls can be accomplished by:

- Busying out the appropriate trunk groups

  and/or

- Using the Vectoring form and setting the first vector step for actively used incoming call vectors to the “busy” step.

Once this is accomplished, perform these activities:

1. Make sure all EAS agents are logged out of all splits. If CMS or BCMS is operational, the CMS real-time reports for splits or the G3-MA/Manager 1 *mon bcms split* command can be used to identify the terminals where agents may still be logged in.
2. Issue the *busy mis* command at the administration terminal to busy-out the CMS link.
3. Issue the *busy link n* command at the administration terminal to busy-out any AUDIX or Messaging Server switch-to-adjunct links.
4. Issue the *busy station x* command at the administration terminal to busy-out any AAS ports.
5. Using the Hunt Group form, convert any ACD splits to skill hunt groups by setting the Skilled? field to y for these hunt groups.

6. Using the Feature-Related System Parameters form, set the Expert Agent Selection (EAS) Enabled? field to y, and if a R3V2 or later release CMS is installed, set the Adjunct CMS Release field.

   NOTE:
   The EAS feature is not compatible with the R2 and R3V1 CMS releases.

7. Release the link to CMS (if installed) by entering the *release mis* command at the administration terminal.
8. Inform the on-site agents that they can log into their terminals using the EAS login procedure and become available to receive ACD calls using the AUTO-IN or MANUAL-IN work-mode operations.
9. Using the Vectoring form, restore any vector steps temporarily changed to "busy" (to block incoming calls) to their previous vector step format.
10. Using the Trunk Group Administration form, if the routing for incoming trunks is to be changed to EAS-related VDNs, administer the Incoming Destination field for any trunk groups to the appropriate VDN extension number.
11. Issue the *release station x* command at the administration terminal to release any AAS ports (where the EAS login ID associated with each AAS port will be automatically logged in).
12. Issue the *release link n* command at the administration terminal to release any adjunct AUDIX links (where the adjunct will cause the associated ports to be logged in).

At this point, the cutover to EAS is complete. It is recommended that a backup of the switch translations be performed as soon as possible after the cutover to preserve the EAS-related administration changes. Also, if agent stations are administered with multiple sets of work-mode buttons, it is recommended that all but one set of work-mode buttons be removed from these stations. Also, multiple queue lights are required for EAS.

Feature Availability

M

This chapter describes available vectoring enhancements. For a detailed description of any item see the referenced section of this guide.

Vectoring (G3V4 Enhanced) provides the following additional capabilities:

- The ability to specify a priority level with the oldest-call-wait conditional on the *check* and *goto* commands. See these commands in Appendix A.
- The use of enhanced comparators (<>, >=, and <=) with the *goto* and *route-to* commands as well as use of “none” as an entry for digits checking, and “active” or “latest” VDN thresholds for indirect VDN references. See these commands in Appendix A.
- The use of the *interflow-qpos* conditional with the *goto* and *route-to* commands to achieve FIFO or FIFO-like call processing. See Chapter 7, “Look-Ahead Interflow”
- The use of wildcards in digit strings for matching on collected digits and ANI or II-digits. See Appendix A.
- The use of Vector Routing Tables for matching on collected digits and ANI or II-digits. See “Vector Routing Tables” on page 4-8 or “ANI /II-Digits Routing and Caller Information Forwarding (CINFO)” on page 6-1.
- Multiple Audio/Music Sources for use with the *wait-time* command. “Multiple Audio or Music Sources on Delay” on page 3-5.

Vectoring (G3V4 Advanced Routing) provides the following additional capabilities (Vectoring [G3V4 Enhanced] must also be enabled):

- Rolling Average Speed of Answer (ASA) Routing. See “Rolling Average Speed of Answer (ASA)” on page 5-10.
- Expected Wait Time (EWT) Routing. See “Expected Wait Time (EWT)” on page 5-2.

- VDN Calls Routing. See "VDN Calls" on page 5-12.

Vectoring (ANI/II-Digits Routing) provides the following additional capabilities (Vectoring [G3V4 Enhanced] must also be enabled):

- ANI Routing. See "ANI Routing" on page 6-2.
- II-Digits Routing. See "II-Digits Routing" on page 6-5.

Vectoring (CINFO) provides the following additional capabilities (Call Prompting must also be enabled):

- The ability to collect ced and cdpd from the network. See "Caller Information Forwarding" on page 6-8

Vectoring (Best Service Routing) automatically compares splits or skills in ACD environments to find the one that can provide the best service to each caller. BSR can operate at a single site, or it can be used with Look-Ahead Interflow to integrate a network of geographically distributed locations into a virtual call center. See Chapter 8, "Best Service Routing™".

Vectoring (Best Service Routing) without LAI enabled (single-site BSR) provides the following capabilities:

- The use of the *consider split/skill* command.
- The use of the *best* keyword with *queue-to*, *check*, and *goto* commands.
- The *wait-improved* conditional for *check* and *goto* commands. For a call that has already been queued, the *wait-improved* conditional gives you the ability to make any subsequent queuing conditional on the improvement in EWT as compared to the call's EWT in its current queue.

Vectoring (Best Service Routing) with LAI enabled (multi-site BSR) provides the following capabilities:

- The use of the *consider split/skill* and *consider location* commands.
- The use of the *reply-best* command to return data to the sending switch in response to a status poll.
- The use of the *best* keyword with *queue-to*, *check*, and *goto* commands.
- The *wait-improved* conditional for *check* and *goto* commands. For a call that has already been queued, the *wait-improved* conditional gives you the ability to make any subsequent queuing conditional on the improvement in EWT as compared to the call's EWT in its current queue.

Enhanced information forwarding provides the transport of existing call information and new call information such as Universal Call ID and Best Service Routing™. See Chapter 9, "Information Forwarding".

Timed ACW provides the ability to assign a timed ACW interval to a VDN. See "Vector Directory Number" on page 2-8.

Improving Performance

N

Introduction

This appendix provides recommendations on how to write vectors that promote favorable performance practices. Two basic principles to follow are:

1. Minimize the amount of call processing
   - — Minimize the number of vector steps to process a call
   - — Use the lower cost steps when possible (refer to Table N-4 and Table N-5)
2. Avoid vector steps which have a substantial probability of failure
   - — Calls made outside of business hours
   - — Queues to groups with less than desirable resources or characteristics

The most wasteful use of processing resources is frequently caused by inefficient looping. For example, performance could be compromised when a vector loops through steps too often. This is especially true with long queue times.

Some examples with looping are discussed and recommendations are given on how to maximize performance. They are:

- Audible Feedback
- Lookahead Interflow
- Check

Examples other than looping are also discussed. They are:

- After Business Hours
- Lookahead Interflow

All looping examples in this appendix use only loops within a single vector. It is important to also be aware of looping to other vectors through the use of vector chaining. The same principles can be extrapolated from the looping examples. Creating a flow diagram is often helpful for identifying looping errors.

In addition to the example vectors, tables rating the relative performance costs of specific vector commands are also included.

NOTE:
Remember to test vectors for performance in addition to call flow.

Looping Examples

Audible Feedback

Recommendation: Evaluate the length of the wait period between repetitions of an announcement and increase the length, if possible. For optimum performance, add a second announcement after the initial announcement and repeat the second announcement less often.

The example in Screen N-1 repeats the “All representative are busy. Please hold.” announcement every 10 seconds as long as the call is in queue.

```
1. queue-to split 1
2. announcement 2770     (“All representatives are busy. Please hold.”)
3. wait-time 10 seconds hearing music
4. goto step 2 if unconditionally
5. stop
```

Screen N-1. Example Vector

The example in Screen N-2 repeats the announcement only every 60 seconds, thus improving performance.

```
1. queue-to split 1
2. announcement 2770     (“All representatives are busy. Please hold.”)
3. wait-time 60 seconds hearing music
4. goto step 2 if unconditionally
5. stop
```

Screen N-2. Example Vector with Improved Performance

The example in Screen N-3 adds a second announcement, “All representatives are still busy. Please hold.” in addition to the initial announcement and repeats the second announcement less often (every 120 seconds), thus improving performance again.

```
1. queue-to split 1
2. announcement 2770     (“All representatives are busy. Please hold.”)
3. wait-time 120 seconds hearing music
4. announcement 2771     (“All representatives are still busy. Please
                         continue to hold.”)
5. goto step 3 if unconditionally
6. stop
```

Screen N-3. Another Example Vector with Improved Performance

Table N-1 compares the relative processing cost of the three examples by looking at the approximate number of vector steps executed while processing the call. Assumption is that the first announcement is 3 seconds long and the second announcement is 4 seconds long.

Table N-1. Approximate Number of Vector Steps Executed for the Audible Feedback Examples

	Example in Screen N-1	Example in Screen N-2	Example in Screen N-3
when an agent is available in split 1	1	1	1
queueing time of 5 minutes	70	15	9

When a call is queued for 5 minutes, the number of vector steps drops dramatically when the amount of time between announcements is increased (Screen N-2), and drops even more when a second announcement is added, and the amount of time between announcements is increased again (Screen N-3). When an agent in split 1 is immediately available to answer the call, there is no difference in the number of vector steps for the three examples.

Lookahead Interflow

Recommendation 1: Use the *interflow-qpos* conditional to achieve FIFO (first in, first out) or near-FIFO call processing. For more information, see Chapter 7, "Look-Ahead Interflow".

Recommendation 2: If you do not have the *interflow-qpos* conditional, add a wait period between successive lookahead interflow attempts and make the waiting period as long as feasible.

The example in Screen N-4 continuously attempts a lookahead interflow as long as the call is in queue or until a lookahead attempt succeeds.

```
1. queue-to split 1 pri l
2. announcement 3000
3. wait-time 20 seconds hearing music
4. route-to number 93035555555 cov n if unconditionally
5. goto step 4 if unconditionally
```

Screen N-4. Example Vector

The example in Screen N-5 adds a delay so that the lookahead interflow attempt occurs only every 10 seconds.

```
1. queue-to split 1 pri l
2. announcement 3000
3. wait-time 20 seconds hearing music
4. route-to number 93035555555 cov n if unconditionally
5. wait-time 10 seconds hearing music
6. goto step 4 if unconditionally
```

Screen N-5. Example Vector with Improved Performance

The example in Screen N-6 increases performance even more by increasing the delay between lookahead interflow attempts to 30 seconds.

```
1. queue-to split 1 pri l
2. announcement 3000
3. wait-time 20 seconds hearing music
4. route-to number 93035555555 cov n if unconditionally
5. wait-time 30 seconds hearing music
6. goto step 4 if unconditionally
```

Screen N-6. Another Example Vector with Improved Performance

Table N-2 compares the relative processing cost of the three examples by looking at the approximate number of vector steps executed while processing the call. Assumption is that the announcement is 5 seconds long.

Table N-2. Approximate Number of Vector Steps Executed for Lookahead Interflow Examples

	Example in Screen N-4	Example in Screen N-5	Example in Screen N-6
when an agent is available in split 1	1	1	1
queueing time of 5 minutes	up to 1,000	85	30

When a call is queued for 5 minutes, the number of vector steps drops dramatically when a delay is added (Screen N-5), and drops even more when the length of the delay is increased (Screen N-6). When an agent in split 1 is immediately available to answer the call, there is no difference in the number of vector steps for the three examples.

Check

Recommendation: When using check commands to queue a call to backup splits, ensure that an adequate amount of time has elapsed before checking the backup splits again.

NOTE:
With the 'Expected Time Wait Time' feature, the style of programming used in this example is not optimal. The best approach is to use the 'Expected Time Wait' feature to locate the most appropriate split for the call and queue it there.

The example in Screen N-7 checks backup splits continuously as long as the call is in queue.

```
1. queue-to split 1 pri h
2. announcement 3000
3. wait-time 10 seconds hearing music
4. check split 21 pri m if available-agents > 0
5. check split 22 pri m if available-agents > 0
6. check split 23 pri m if available-agents > 0
7. check split 24 pri m if available-agents > 0
8. check split 25 pri m if available-agents > 0
9. goto step 4 if unconditionally
```

Screen N-7. Example Vector

The example in Screen N-8 adds a delay of 10 seconds to ensure that some time has elapsed before checking the backup splits again.

```
1. queue-to split 1 pri h
2. announcement 3000
3. wait-time 30 seconds hearing music
4. check split 21 pri m if available-agents > 0
5. check split 22 pri m if available-agents > 0
6. check split 23 pri m if available-agents > 0
7. check split 24 pri m if available-agents > 0
8. check split 25 pri m if available-agents > 0
9. wait-time 10 seconds hearing music
10. goto step 4 if unconditionally
```

Screen N-8. Example Vector with Improved Performance

Since the agent availability status may not be likely to change every 10 seconds, it may make sense to increase the wait time to 30 seconds, as shown in the example in Screen N-9.

```
1. queue-to split 1 pri h
2. announcement 3000
3. wait-time 30 seconds hearing music
4. check split 21 pri m if available-agents > 0
5. check split 22 pri m if available-agents > 0
6. check split 23 pri m if available-agents > 0
7. check split 24 pri m if available-agents > 0
8. check split 25 pri m if available-agents > 0
9. wait-time 30 seconds hearing music
10. goto step 4 if unconditionally
```

Screen N-9. Another Example Vector with Improved Performance

Table N-3 compares the relative processing cost of the three examples by looking at the approximate number of vector steps executed while processing the call. Assumption is that the announcement is 5 seconds long.

Table N-3. Approximate Number of Vector Steps Executed for Check Examples

	Example in Screen N-7	Example in Screen N-8	Example in Screen N-9
when an agent is available in split 1	1	1	1
queueing time of 5 minutes	up to 1,000	190	65

When a call is queued for 5 minutes, the number of vector steps drops dramatically when a delay is added before checking the backup splits again (Screen N-8), and drops even more when the length of the delay is increased again (Screen N-9). When an agent in split 1 is immediately available to answer the call, there is no difference in the number of vector steps for the three examples.

Other Examples

After Business Hours

Recommendation: Test to see if the destination resources are available (such as during business hours) before queuing.

The example in Screen N-10 queues calls to a hunt group regardless of the time of the call. When the call is made after business hours, the announcement is repeated until the caller hangs up.

```
1. queue-to split 1
2. announcement 5000
                          (“All agents are busy. Please hold.”)
3. wait-time 120 seconds hearing music
4. announcement 5001
                          (“All agents are still busy. Please continue to
                          hold.”)
5. goto step 3 if unconditionally
```

Screen N-10. Example Vector

The example in Screen N-11 tests for business hours before queuing the call. If the call is made after business hours, an announcement informs the caller of the business hours and the call is terminated.

```
1. goto step 7 if time-of-day is all 17:00 to all 8:00
2. queue-to split 1
3. announcement 5000
                              (“All agents are busy. Please hold.”)
4. wait-time 120 seconds hearing music
5. announcement 5001
                              (“All agents are still busy. Please
                              continue to hold.”)
6. goto step 4 if unconditionally
7. disconnect after announcement 5001
                              (“Business hours are 8:00 AM to 5:00 PM,
                              Please call back then.”)
```

Screen N-11. Example Vector with Improved Performance

In the first example, unnecessary processing occurs when a call is queued after business hours and the call is terminated only when the caller hangs up. As shown in the second example, it is more economical to test for business hours before queuing a call.

Lookahead Interflows

Recommendation: When using a lookahead interflow, first test to see if the receiving office is open for business.

The scenario is a sending switch in Los Angeles, with office hours from 8:00 AM to 5:00PM (8:00-17:00) PST and the receiving switch is in New York, with office hours from 8:00 AM to 5:00PM EST (5:00-14:00 PST). There is a three hour difference between the two switches

The example in Screen N-12 routes calls to the New York switch. If there are no agents available at the Los Angeles switch, it is possible for calls to be interflowed during hours that the agents in New York are not available, thus doing unnecessary processing.

```
1. queue-to split 1
2. route-to number 99145555555 cov n if unconditionally
3. announcement 2770              (“All agents are busy. Please hold.”)
4. wait-time 120 seconds hearing music
5. goto step 3 if unconditionally
6. stop
```

Screen N-12. Example Vector

The example in Screen N-13 tests first to see if the New York switch is open before requesting a queue to the New York switch, thus avoiding unnecessary processing.

```
1. queue-to split 1
2. goto step 4 if time-of-day is all 14:00 to all 05:00
3. route-to number 99145555555 cov n if unconditionally
4. announcement 2770              (“All agents are busy. Please hold.”)
5. wait-time 120 seconds hearing music
6. goto step 4 if unconditionally
7. stop
```

Screen N-13. Example Vector with Improved Performance

The example in Screen N-14 can be used if you have Advanced Routing optioned. In this case, the ‘Expected Wait Time’ feature may be used to determine whether it is worthwhile placing a lookahead interflow call attempt.

```
1. queue-to split 1
2. goto step 5 if expected-wait for call < 30
3. goto step 5 if time-of-day is all 14:00 to all 05:00
4. route-to number 99145555555 cov n if unconditionally
5. announcement 2770              (“All agents are busy. Please hold.”)
6. wait-time 120 seconds hearing music
7. goto step 5 if unconditionally
8. stop
```

Screen N-14. Another Example Vector with Improved Performance

There is little reason to attempt an interflow if the call will be answered quickly at the main switch. For the examples in Screen N-13 and Screen N-14, vector steps that do not aid in the call being answered sooner are avoided.

Relative Processing Cost of Vector Commands

Some vector commands use more processing resources than others. Table N-4 and Table N-5 show the relative processing costs of specific vector commands for DEFINITY ECS R5si/vs and DEFINITY ECS R5r respectively. Whenever possible, use the lower cost vector commands. This will minimize your performance costs and upgrade your performance.

Table N-4. Relative Processing Cost of Vector Commands for DEFINITY ECS R6si/vs

relative performance cost	vector command
high	adjunct routing
high	check
high	collect digits
high	consider location
high	queue-to
high	route-to
medium	announcement
medium	collect ced/cdpd digits
medium	consider split/skill
medium	converse
medium	goto step
medium	goto vector
medium	messaging
medium	reply-best
low	busy
low	disconnect
low	stop
low	wait-time

Continued on next page

Table N-5. Relative Processing Cost of Vector Commands for DEFINITY ECS R6r

relative performance cost	vector command
medium	adjunct routing
medium	check
medium	collect digits
medium	consider location
medium	converse
medium	goto vector (table comparison)
medium	messaging
medium	queue-to
medium	route-to
low	announcement
low	busy
low	collect ced/cdpd digits
low	consider split/skill
low	disconnect
low	goto step
low	goto vector
low	reply-best
low	stop
low	wait-time

Call Vectoring System Parameters

O

Table O-1. Maximum System Parameters for Call Vectoring/EAS

ITEM	R6 vs/csi/si	R6 r
Call Vectoring		
Maximum Skills a Call Can Simultaneously Queue to	3	3
Priority Levels	4	4
Recorded Announcements/Analog Sources for Vector Delay	128	256
Steps per Vector	32	32
Vector Directory Numbers	512	20,000
CMS Measured VDNs[1]	512	8,000
Vectors per System	256	512
Number of Collected Digits for Call Prompting or CINFO	16	16
Number of Dial-Ahead Digits for Call Prompting	24	24
Vector Routing Tables	10	100
BSR Application Plans	255	255
Number of Locations per BSR Application	255	255
Number of BSR Application/Location Pairs	1,000	1,000
Expert Agent Selection (EAS)		
Skill Groups	99	600

Continued on next page

Table O-1. Maximum System Parameters for Call Vectoring/EAS

ITEM	R6 vs/csi/si	R6 r
VDN Skill Preferences	3	3
Maximum Skills a Call Can Simultaneously Queue to	3	3
Maximum Agent Login IDs Administered		
1–4 skills	1,500	10,000
10 skills	600	4,000
20 skills	300	2,000
Maximum Agent Login IDs Staffed[2]	500	5,200
Maximum Agent/Skill Pairs Administered[3]	6,000	40,000
Maximum Agent/Skill Pairs Staffed	1,000	10,000
Maximum Skills Per Agent		
No CMS	20	20
R3V2/R3V4 CMS	4	4
R3V5/R3V6 CMS	20	20
Maximum Skill Levels	16	16
Maximum Agents Logged-In[4]		
When Each Has 20 Skills Assigned	50	500
When Each Has 4 Skills Assigned	75	2,500
When Each Has 1 Skill Assigned	500	5,200

1. Measured limits depend on the CMS release and platform used.
2. The total of the administered AAS ports and the ACD agents that log in (regardless of skills per agent) whether measured or not.
3. The total of the administered Login ID skill-pair members (total of agent skills and AAS ports).
4. The number of agents that can be logged in at any given time is dependent on the limits imposed by both the maximums for Agent Login IDs staffed and Agent/Skill Pairs staffed (whichever limit is reached first). Each skill that an agent is assigned counts as an Agent/Skill Pair. Maximum agent limits are reduced by the number of non-ACD members and AAS ports administered.

References

P

This chapter contains a list of user documents for the DEFINITY Enterprise Communications Server (ECS) Release 7. (Most of these documents are backward compatible with, and can be used with, the DEFINITY ECS Release 6 systems.)

To order these or other DEFINITY documents, contact the Lucent Technologies Publications Center at the address and phone number on the back of the title page of this document. A complete catalog of Business Communications Systems (BCS) documents, including previous issues of the documents listed here, is available on the World Wide Web. Ask your account team for the web address.

Basic DEFINITY ECS documents

These documents are issued for all new and upgrade DEFINITY ECS Release 7 systems.

Administration

The primary audience for these documents consists of customer administrators.

DEFINITY ECS Release 7 — Overview, 555-230-024, Issue 7

Provides a brief description of the features available with DEFINITY ECS R7. This book does not provide a general overview of the switch nor of basic telephony. This book is available in the following languages: English, German (DE), Dutch (NL), Brazilian Portuguese (PTB), European French (FR), Latin Spanish (SPL), Italian (IT), Russian (RU), and Japanese (JA). To order, append the language suffix to the document number; for example, 555-230-024DE for German. No suffix is needed for the English version.

DEFINITY ECS Release 7 — Administrator's Guide, 555-233-502, Issue 1

A task-based document that provides step-by-step procedures for administering the switch. This book contains information previously found in DEFINITY ECS Administration and Feature Description, 555-230-522, as well as new information for this release and procedural sections added in Release 7.

DEFINITY ECS Release 7.1 — Change Description, 555-230-405, Issue 1

Provides a high-level overview of what is new in DEFINITY ECS R7.1. Describes the hardware and software enhancements and lists the problem corrections for this release. It also includes any last-minute changes that come in after the remaining books have gone to production.

DEFINITY ECS Release 7 — System Description Pocket Reference, 555-230-211, Issue 4

Provides hardware descriptions, system parameters, listing of hardware required to use features, system configurations, and environmental requirements.

DEFINITY System's Little Instruction Book for basic administration, 555-230-727, Issue 5

Provides step-by-step procedures for performing basic switch administration tasks. Includes managing phones, managing features, routing outgoing calls, and enhancing system security.

DEFINITY System's Little Instruction Book for advanced administration, 555-233-712, Issue 2

Provides step-by-step procedures for managing trunks, managing hunt groups, setting up night service, writing vectors, recording announcements, using reports, and understanding call centers.

DEFINITY System's Little Instruction Book for basic diagnostics, 555-230-713, Issue 2

Provides step-by-step procedures for baselining your system, solving common problems, reading alarms and errors, using features to troubleshoot your system, and contacting Lucent.

DEFINITY ECS Release 7 — Reports, 555-230-511, Issue 5

Provides detailed descriptions of the measurement, status, security, and recent change history reports available in the system and is intended for administrators who validate traffic reports and evaluation system performance. Includes corrective actions for potential problems.

BCS Products Security Handbook, 555-025-600, Issue 6

Provides information about the risks of telecommunications fraud and measures for addressing those risks and preventing unauthorized use of BCS products. This document is intended for telecommunications managers, console operators, and security organizations within companies.

DEFINITY ECS Release 7 — System 75 and System 85 Terminals and Adjuncts Reference, 555-015-201, Issue 10

Provides drawings and full descriptions for all phones, phone adjuncts, and data terminals that can be used with System 75, System 85, DEFINITY Communications System, and DEFINITY ECS. This document is intended for customers and Lucent Technologies account teams for selecting the correct equipment.

Guide Builder™ Software for DEFINITY® Telephones, 555-230-755

Provides the ability to produce laser-printed documentation for specific telephones. A comprehensive user's guide and on-line help support the software. This information applies to Release 7 as well as earlier DEFINITY systems. All customers receive this software.

Installation and maintenance

DEFINITY ECS Release 7 — Administration for Network Connectivity, 555-233-501, Issue 1

Describes the main types of switch-to-switch connections that use OverLAN hardware and software, and the procedures required to administer these connections.

DEFINITY ECS Release 7 — Installation and Test for Single-Carrier Cabinets, 555-230-894, Issue 4

Provides procedures and information for hardware installation and initial testing of single-carrier cabinets. This book is available in the following languages: English, German (DE), Dutch (NL), Brazilian Portuguese (PTB), European French (FR), Latin Spanish (SPL), Italian (IT), Russian (RU), and Japanese (JA). To order, append the language suffix to the document number; for example, 555-230-894DE for German. No suffix is needed for the English version.

DEFINITY ECS Release 7 — Installation and Test for Multi-Carrier Cabinets, 555-230-112, Issue 6

Provides procedures and information for hardware installation and initial testing of multi-carrier cabinets.

DEFINITY ECS Release 7 — Installation for Adjuncts and Peripherals, 555-230-125, Issue 5

Provides procedures and information for hardware installation and initial testing of ECS adjunct and peripheral systems and equipment.

DEFINITY ECS Release 7 — Installation, Upgrades and Additions for Compact Modular Cabinets, 555-230-128, Issue 4

Provides procedures and information for hardware installation and initial testing of compact modular cabinets.

DEFINITY ECS Release 7 — ATM Installation, Upgrades, and Administration, 555-233-106, Issue 1

Provides step-by-step instructions for how to install, upgrade, and administer ATM switches.

DEFINITY ECS Release 7 — Installation and Maintenance for Survivable Remote EPN, 555-233-102, Issue 2

Describes how to install, cable, test, and perform maintenance on a Survivable Remote Expansion Port Network (SREPN). Provides power, ground, and fiber connections.

DEFINITY ECS Release 7 — Upgrades and Additions for R7r, 555-230-121, Issue 5

Provides procedures for an installation technician to upgrade an existing DEFINITY Communications System or DEFINITY ECS to DEFINITY ECS Release 7.

Includes upgrade considerations, lists of required hardware, and step-by-step upgrade procedures. Also includes procedures to add control carriers, switch node carriers, port carriers, circuit packs, auxiliary cabinets, and other equipment.

DEFINITY ECS Release 7 — Upgrades and Additions for R7si, 555-233-104, Issue 1

Provides procedures for an installation technician to upgrade an existing DEFINITY Communications System or DEFINITY ECS to DEFINITY ECS Release 7.

Includes upgrade considerations, lists of required hardware, and step-by-step upgrade procedures. Also includes procedures to add control carriers, switch node carriers, port carriers, circuit packs, auxiliary cabinets, and other equipment.

DEFINITY ECS Release 7 — Maintenance for R7r, 555-230-126, Issue 4

Provides detailed descriptions of the procedures for monitoring, testing, troubleshooting, and maintaining the R7r ECS. Included are maintenance architecture, craft commands, step-by-step trouble-clearing procedures, the procedures for using all tests, and explanations of the system's error codes.

DEFINITY ECS Release 7 — Maintenance for R7si, 555-233-105, Issue 1

Provides detailed descriptions of the procedures for monitoring, testing, troubleshooting, and maintaining the R7si ECS. Included are maintenance architecture, craft commands, step-by-step trouble-clearing procedures, the procedures for using all tests, and explanations of the system's error codes.

DEFINITY ECS Release 7 — Maintenance for R7csi, 555-230-129, Issue 4

Provides detailed descriptions of the procedures for monitoring, testing, troubleshooting, and maintaining the R7csi (Compact Modular Cabinet) ECS. Included are maintenance architecture, craft commands, step-by-step trouble-clearing procedures, the procedures for using all tests, and explanations of the system's error codes.

Call center documents

These documents are issued for DEFINITY ECS Call Center applications. The intended audience is DEFINITY ECS administrators.

DEFINITY

DEFINITY ECS Release 7 — Guide to ACD Call Centers, 555-233-503, Issue 1

This module contains information about the call center-specific features of the DEFINITY ECS. This information was previously contained in DEFINITY ECS Administration and Feature Description.

DEFINITY ECS Release 7 — Call Vectoring/EAS Guide, 555-230-521, Issue 2

Provides information on how to write, use, and troubleshoot vectors, which are command sequences that process telephone calls in an Automatic Call Distribution (ACD) environment.

It is provided in two parts: tutorial and reference. The tutorial provides step-by-step procedures for writing and implementing basic vectors. The reference includes detailed descriptions of the call vectoring features, vector management, vector administration, adjunct routing, troubleshooting, and interactions with management information systems (including the Call Management System).

DEFINITY ECS Release 7 — Basic Call Management System (BCMS) Operations, 555-230-706, Issue 1

Provides detailed instructions on how to generate reports and manage the system. It is intended for telecommunications managers who wish to use Basic Call Management System (BCMS) reports and for system managers responsible for maintaining the system. This documentation applies to Release 7 as well as earlier DEFINITY systems.

CentreVu CMS

The following documents provide information about administration and use of the CentreVu Call Management System.

CentreVu Call Management System Release 3 Version 6— Administration, Issue 1, 585-215-850

CentreVu Supervisor Version 6 — Reports, 585-215-851, Issue 1

CentreVu Call Management System Release 3 Version 5 — Custom Reports, 585-215-822, Issue 1

CentreVu Call Management System Release 3 Version 6 — Upgrades and Migrations, 585-215-856, Issue 1

CentreVu Call Management System Release 3 Version 6 — External Call History Interface, 585-215-854, Issue 1

CentreVu Call Management System Release 3 Version 5 — Forecast, 585-215-825, Issue 1

CentreVu CMS R3V6 — Planning, Configuration, and Implementation, 585-215-879, Issue 1

Application-specific documents

These documents support specific DEFINITY documents.

ASAI

DEFINITY Communications System G3 — CallVisor ASAI Planning Guide, 555-230-222, Issue 4

Provides procedures and directions for the account team and customer personnel for effectively planning and implementing the CallVisor Adjunct/Switch Application Interface (ASAI) PBX-Host environment. The CallVisor ASAI is a communications interface that allows adjunct processors to access switch features and to control switch calls. It is implemented using an Integrated Services Digital Network (ISDN) Basic Rate Interface (BRI). Hardware and software requirements are included.

DEFINITY ECS Release 6 — CallVisor ASAI Protocol Reference, 555-230-221, Issue 7

Provides detailed layer 3 protocol information regarding the CallVisor Adjunct/Switch Application Interface (ASAI) for the systems and is intended for the library or driver programmer of an adjunct processor to create the library of commands used by the applications programmers. Describes the ISDN message, facility information elements, and information elements.

DEFINITY ECS Release 6 — CallVisor ASAI Technical Reference, 555-230-220, Issue 7

Provides detailed information regarding the CallVisor Adjunct/Switch Application Interface (ASAI) for the systems and is intended for the application designer responsible for building and/or programming custom applications and features.

DEFINITY ECS — CallVisor ASAI DEFINITY LAN Gateway over MAP-D Installation, Administration, and Maintenance of, 555-230-113, Issue 2

Provides procedures for installation, administration, and maintenance of the CallVisor Adjunct/Switch Application Interface (ASAI) Ethernet application over the DEFINITY LAN Gateway and is intended for system administrators, telecommunications managers, Management Information System (MIS) managers, LAN managers, and Lucent personnel. The ASAI-Ethernet application provides ASAI functionality using 10Base-T Ethernet rather than BRI as a transport media.

DEFINITY ECS — CallVisor ASAI PC LAN over MAP-D Installation, Administration, and Maintenance of, 555-230-114, Issue 1

Provides procedures for installation, administration, and maintenance of the CallVisor Adjunct/Switch Application Interface (ASAI) Ethernet application over the PC LAN and is intended for system administrators, telecommunications managers, Management Information System (MIS) managers, LAN managers, and Lucent personnel. The ASAI-Ethernet application provides ASAI functionality using 10Base-T Ethernet rather than BRI as a transport media.

DEFINITY ECS Release 6 — CallVisor ASAI Overview, 555-230-225, Issue 2

Provides a general description of Call Visor ASAI.

This document is available in the following languages: English, German (DE), Dutch (NL), Brazilian Portuguese (PTB), European French (FR), Colombian Spanish (SPL), and Japanese (JA). To order, append the language suffix to the document number; for example, 555-230-894DE for German. No suffix is needed for the English version.

DEFINITY ECS Release 6 — CallVisor PC ASAI Installation and Reference, 555-230-227, Issue 3

Provides procedural and reference information for installers, Tier 3 support personnel, and application designers.

ACD

DEFINITY Communications Systems G3 — Automatic Call Distribution (ACD) Agent Instructions, 555-230-722, Issue 5

Provides information for use by agents after they have completed ACD training. Includes descriptions of ACD features and the procedures for using them.

DEFINITY Communications Systems G3 — Automatic Call Distribution (ACD) Supervisor Instructions, 555-230-724, Issue 4

Provides information for use by supervisors after they have completed ACD training. Includes descriptions of ACD features and the procedures for using them.

Console operations

The primary audience for these documents consists of attendant console users.

DEFINITY ECS Console Operations, 555-230-700, Issue 4

Provides operating instructions for the attendant console. Included are descriptions of the console control keys and functions, call-handling procedures, basic system troubleshooting information, and routine maintenance procedures.

DEFINITY ECS Release 7 — Console Operations Quick Reference, 555-230-890, Issue 3

Provides operating instructions for the attendant console.Included are descriptions of the console control keys and functions, call-handling procedures, basic system troubleshooting information, and routine maintenance procedures. This book is available in the following languages: English, German (DE), Dutch (NL), Brazilian Portuguese (PTB), European French (FR), Latin Spanish (SPL), Italian (IT), Russian (RU), and Japanese (JA). To order, append the language suffix to the document number; for example, 555-230-890DE for German. No suffix is needed for the English version.

Hospitality

The primary audience for these documents consists of hotel and health care administrators, and telephone operators at these locations.

DEFINITY ECS Release 7 — Hospitality Operations, 555-230-723, Issue 5

Provides step-by-step procedures for using the features available for the lodging and health industries to improve their property management and to provide assistance to their employees and clients. Includes detailed descriptions of reports.

Non-U.S. audiences

DEFINITY ECS Release 7 — Application Notes for Type Approval

Describes specific hardware and administration required to operate the DEFINITY ECS in countries outside the United States. Available on the DEFINITY ECS Release 7 Library CD only.

Glossary and Abbreviations

Numerics

3B2 Message Server
A software application that combines voice and data messaging services for voice-terminal users whose extensions are connected to a system.

800 service
A service in the United States that allows incoming calls from certain areas to an assigned number for a flat-rate charge based on usage.

A

AA
Archangel. See angel.

AAC
ATM access concentrator

AAR
See Automatic Alternate Routing (AAR).

abandoned call
An incoming call in which the caller hangs up before the call is answered.

Abbreviated Dialing (AD)
A feature that allows callers to place calls by dialing just one or two digits.

AC
1. Alternating current.
2. See Administered Connection (AC).

AAR
Automatic Alternate Routing

ACA
See Automatic Circuit Assurance (ACA).

ACB
See Automatic Callback (ACB).

ACD
See Automatic Call Distribution (ACD).

ACD agent
See agent.

ACU
See Automatic calling unit (ACU)

ACW
See after-call work (ACW) mode.

access code
A 1-, 2-, or 3-digit dial code used to activate or cancel a feature, or access an outgoing trunk.

access endpoint
Either a nonsignaling channel on a DS1 interface or a nonsignaling port on an analog tie-trunk circuit pack that is assigned a unique extension.

access tie trunk
A trunk that connects a main communications system with a tandem communications system in an electronic tandem network (ETN). An access tie trunk can also be used to connect a system or tandem to a serving office or service node. Also called access trunk.

access trunk
See access tie trunk.

ACCUNET
A trademarked name for a family of digital services offered by AT&T in the United States.

ACD
See Automatic Call Distribution (ACD). ACD also refers to a work state in which an agent is on an ACD call.

ACD work mode
See work mode.

active-notification association
A link that is initiated by an adjunct, allowing it to receive event reports for a specific switch entity, such as an outgoing call.

active-notification call
A call for which event reports are sent over an active-notification association (communication channel) to the adjunct. Sometimes referred to as a monitored call.

active notification domain
VDN or ACD split extension for which event notification has been requested.

ACU
See Automatic calling unit (ACU).

AD
See Abbreviated Dialing (AD).

ADAP
AUDIX Data Acquisition Package

ADC
See analog-to-digital converter (ADC).

adjunct
A processor that does one or more tasks for another processor and that is optional in the configuration of the other processor. See also application.

adjunct-control association
A relationship initiated by an application via *Third Party Make Call*, the *Third Party Take Control,* or *Domain (Station) Control* capabilities to set up calls and control calls already in progress.

adjunct-controlled call
Call that can be controlled using an adjunct-control association. Call must have been originated via *Third Party Make Call* or *Domain (Station) Control* capabilities or must have been taken control of via *Third Party Take Control* or *Domain (Station) Control* capabilities.

adjunct-controlled split
An ACD split that is administered to be under adjunct control. Agents logged into such splits must do all telephony work, ACD login/ logout, and changes of work mode through the adjunct (except for auto-available adjunct-controlled splits, whose agents may not log in/out or change work mode).

adjunct-monitored call
An adjunct-controlled call, active-notification call, or call that provides event reporting over a domain-control association.

Adjunct-Switch Application Interface (ASAI)
A recommendation for interfacing adjuncts and communications systems, based on the CCITT Q.932 specification for layer 3.

adjusted EWT
Expected Wait Time (EWT) plus a user adjustment set by a *consider* command.

ADM
Asynchronous data module

administer
To access and change parameters associated with the services or features of a system.

Administered Connection (AC)
A feature that allows the switch to automatically establish and maintain end-to-end connections between access endpoints (trunks) and/or data endpoints (data modules).

administration group
See capability group.

administration terminal
A terminal that is used to administer and maintain a system. See also terminal.

Administration Without Hardware (AWOH)
A feature that allows administration of ports without associated terminals or other hardware.

ADU
See asynchronous data unit (ADU).

AE
See access endpoint.

after-call work (ACW) mode
A mode in which agents are unavailable to receive ACD calls. Agents enter the ACW mode to perform ACD-related activities such as filling out a form after an ACD call.

AG
ASAI Gateway

agent
A person who receives calls directed to a split. A member of an ACD hunt group or ACD split. Also called an ACD agent.

agent report
A report that provides historical traffic information for internally measured agents.

agent selection method
The method DEFINITY ECS uses to select an agent in a hunt group when more than one agent is available to receive the next call: UCD-MIA, UCD-LOA, EAD-MIA, or EAD-LOA

AIM
Asynchronous interface module

AIOD
Automatic Identification of Outward Dialing

ALBO
Automatic Line Build Out

All trunks busy (ATB)
The state in which no trunks are available for call handling.

ALM-ACK
Alarm acknowledge

American Standard Code for Information Interchange
See ASCII (American Standard Code for Information Interchange).

AMW
Automatic Message Waiting

AN
Analog

analog
The representation of information by continuously variable physical quantities such as amplitude, frequency, and phase. See also digital.

analog data
Data that is transmitted over a digital facility in analog (PCM) form. The data must pass through a modem either at both ends or at a modem pool at the distant end.

analog telephone
A telephone that receives acoustic voice signals and sends analog electrical signals along the telephone line. Analog telephones are usually served by a single wire pair (tip and ring). The model-2500 telephone set is a typical example of an analog telephone.

analog-to-digital converter (ADC)
A device that converts an analog signal to digital form. See also digital-to-analog converter (DAC).

angel
A microprocessor located on each port card in a processor port network (PPN). The angel uses the control-channel message set (CCMS) to manage communications between the port card and the archangel on the controlling switch-processing element (SPE). The angel also monitors the status of other microprocessors on a port card and maintains error counters and thresholds.

ANI
See Automatic Number Identification (ANI).

ANSI
American National Standards Institute**.** A United States professional/technical association supporting a variety of standards.

answerback code
A number used to respond to a page from a code-calling or loudspeaker-paging system, or to retrieve a parked call.

AOL
Attendant-offered load

AP
Applications processor

APLT
Advanced Private-Line Termination

appearance
A software process that is associated with an extension and whose purpose is to supervise a call. An extension can have multiple appearances. Also called call appearance, line appearance, and occurrence. See also call appearance.

application
An adjunct that requests and receives ASAI services or capabilities. One or more applications can reside on a single adjunct. However, the switch cannot distinguish among several applications residing on the same adjunct and treats the adjunct, and all resident applications, as a single application. The terms application and adjunct are used interchangeably throughout this document.

In Best Service Routing, an application is any specific implementation of multi-site Best Service Routing.

application plan
Used only in multi-site BSR applications, the application plan identifies the remote switches that may be compared in consider series. The plan also specifies the information used to contact each switch and to interflow calls to it.

applications processor
A micro-computer based, program controlled computer providing application services for the DEFINITY switch. The processor is used with several user-controlled applications such as traffic analysis and electronic documentation.

application service element
See capability group.

architecture
The organizational structure of a system, including hardware and software.

ARS
See Automatic Route Selection (ARS).

ASAI
See Adjunct-Switch Application Interface (ASAI)

ASCII (American Standard Code for Information Interchange)
The standard code for representing characters in digital form. Each character is represented by an 8-bit code (including parity bit).

association
A communication channel between adjunct and switch for messaging purposes. An active association is one that applies to an existing call on the switch or to an extension on the call.

asynchronous data transmission
A method of transmitting data in which each character is preceded by a start bit and followed by a stop bit, thus permitting data characters to be transmitted at irregular intervals. This type transmission is advantageous when transmission is not regular (characters typed at a keyboard). Also called asynchronous transmission. See also synchronous data transmission.

asynchronous data unit (ADU)
A device that allows direct connection between RS-232C equipment and a digital switch.

asynchronous Transfer Mode (ATM)
A packet-like switching technology in which data is transmitted in fixed-size (53-byte) cells. ATM provides high-speed access for data communication in LAN, campus, and WAN environments.

ATB
See All trunks busy (ATB).

ATD
See Attention dial (ATD).

attendant
A person at a console who provides personalized service for incoming callers and voice-services users by performing switching and signaling operations. See also attendant console.

ATM
See asynchronous Transfer Mode (ATM).

attendant console
The workstation used by an attendant. The attendant console allows the attendant to originate a call, answer an incoming call, transfer a call to another extension or trunk, put a call on hold, and remove a call from hold. Attendants using the console can also manage and monitor some system operations. Also called console. See also attendant.

Attention dial (ATD)
A command in the Hayes modem command set for asynchronous modems.

Audio Information Exchange (AUDIX)
A fully integrated voice-mail system. Can be used with a variety of communications systems to provide call-history data, such as subscriber identification and reason for redirection.

AUDIX
See Audio Information Exchange (AUDIX).

auto-in trunk group
Trunk group for which the CO processes all of the digits for an incoming call. When a CO seizes a trunk from an auto-in trunk group, the switch automatically connects the trunk to the destination — typically an ACD split where, if no agents are available, the call goes into a queue in which callers are answered in the order in which they arrive.

Auto-In Work mode
One of four agent work modes: the mode in which an agent is ready to process another call as soon as the current call is completed.

Automatic Alternate Routing (AAR)
A feature that routes calls to other than the first-choice route when facilities are unavailable.***

Automatic Callback (ACB)
A feature that enables internal callers, upon reaching a busy extension, to have the system automatically connect and ring both parties when the called party becomes available.

Automatic Call Distribution (ACD)
A feature that answers calls, and then, depending on administered instructions, delivers messages appropriate for the caller and routes the call to an agent when one becomes available.

Automatic Call Distribution (ACD) split
A method of routing calls of a similar type among agents in a call center. Also, a group of extensions that are staffed by agents trained to handle a certain type of incoming call.

Automatic calling unit (ACU)
A device that places a telephone call.

Automatic Circuit Assurance (ACA)
A feature that tracks calls of unusual duration to facilitate troubleshooting. A high number of very short calls or a low number of very long calls may signify a faulty trunk.

Automatic Number Identification (ANI)
Representation of the calling number, for display or for further use to access information about the caller. Available with Signaling System 7.

automatic restoration
A service that restores disrupted connections between access endpoints (nonsignaling trunks) and data endpoints (devices that connect the switch to data terminal and/or communications equipment). Restoration is done within seconds of a service disruption so that critical data applications can remain operational.

Automatic Route Selection (ARS)
A feature that allows the system to automatically choose the least-cost way to send a toll call.

automatic trunk
A trunk that does not require addressing information because the destination is predetermined. A request for service on the trunk, called a seizure, is sufficient to route the call. The normal destination of an automatic trunk is the communications-system attendant group. Also called automatic incoming trunk and automatic tie trunk.

AUX
Auxiliary

auxiliary equipment
Equipment used for optional system features, such as Loudspeaker Paging and Music-on-Hold.

auxiliary trunk
A trunk used to connect auxiliary equipment, such as radio-paging equipment, to a communications system.

Aux-Work mode
A work mode in which agents are unavailable to receive ACD calls. Agents enter Aux-Work mode when involved in non-ACD activities such as taking a break, going to lunch, or placing an outgoing call.

available agent strategy
Part of the Best Service Routing feature, the available agent strategy determines how BSR commands in a vector identify the best split or skill when several have available agents. The possible available agent strategies are: UCD-MIA, UCD-LOA, EAD-MIA, and EAD-LOA.

AVD
Alternate voice/data

AWOH
See Administration Without Hardware (AWOH).

AWG
American Wire Gauge

AWT
Average work time

B

B8ZS
Bipolar Eight Zero Substitution.

bandwidth
The difference, expressed in hertz, between the defined highest and lowest frequencies in a range.

barrier code
A security code used with the Remote Access feature to prevent unauthorized access to the system.

baud
A unit of transmission rate equal to the number of signal events per second. See also bit rate and bits per second (bps).

BCC
See Bearer capability class (BCC).

BCMS
Basic Call Management System

BCT
See business communications terminal (BCT).

Bearer capability class (BCC)
Code that identifies the type of a call (for example, voice and different types of data). Determination of BCC is based on the caller's characteristics for non-ISDN endpoints and on the Bearer Capability and Low-Layer Compatibility Information Elements of an ISDN endpoint. Current BCCs are 0 (voice-grade data and voice), 1 (DMI mode 1, 56 kbps data transmission), 2 (DMI mode 2, synchronous/asynchronous data transmission up to 19.2 kbps), 3 (DMI mode 3, 64 kbps circuit/packet data transmission), 4 (DMI mode 0, 64 kbps synchronous data), 5 (temporary signaling connection, and 6 (wideband call, 128–1984 kbps synchronous data).

BER
Bit error rate

best
The split, skill, or location that will provide the best service for a caller as determined by Best Service Routing.

Best Service Routing™ (BSR)
A DEFINITY ECS feature, based on call vectoring, that routes ACD calls to the split, skill, or call center best able to service each call. Best Service Routing can be used on a single switch, or it can be used to integrate resources across a network of DEFINITY Enterprise Communication Servers.

BHCC
Busy-hour call completions

bit (binary digit)
One unit of information in binary notation, having two possible values: 0 or 1.

bits per second (bps)
The number of binary units of information that are transmitted or received per second. See also baud and **bit rate**.

bit rate
The speed at which bits are transmitted, usually expressed in bits per second. Also called data rate. See also baud and bits per second (bps).

BLF
Busy Lamp Field

BN
Billing number

BOS
Bit-oriented signaling

BPN
Billed-party number

bps
See bits per second (bps).

bridge (bridging)
The appearance of a voice terminal's extension at one or more other voice terminals.

BRI
The ISDN Basic Rate Interface specification.

bridged appearance
A call appearance on a voice terminal that matches a call appearance on another voice terminal for the duration of a call.

BSR
See Best Service Routing™ (BSR).

BTU
British Thermal Unit

buffer

1. In hardware, a circuit or component that isolates one electrical circuit from another. Typically, a buffer holds data from one circuit or process until another circuit or process is ready to accept the data.
2. In software, an area of memory that is used for temporary storage.

bus
A multiconductor electrical path used to transfer information over a common connection from any of several sources to any of several destinations.

business communications terminal (BCT)
A digital data terminal used for business applications. A BCT can function via a data module as a special-purpose terminal for services provided by a processor or as a terminal for data entry and retrieval.

BX.25
A version of the CCITT X.25 protocol for data communications. BX.25 adds a fourth level to the standard X.25 interface. This uppermost level combines levels 4, 5, and 6 of the ISO reference model.

bypass tie trunks
A 1-way, outgoing tie trunk from a tandem switch to a main switch in an ETN. Bypass tie trunks, provided in limited quantities, are used as a last-choice route when all trunks to another tandem switch are busy. Bypass tie trunks are used only if all applicable intertandem trunks are busy.

byte
A sequence of (usually eight) bits processed together.

C

CACR
Cancellation of Authorization Code Request

cabinet
Housing for racks, shelves, or carriers that hold electronic equipment.

cable
Physical connection between two pieces of equipment (for example, data terminal and modem) or between a piece of equipment and a termination field.

cable connector
A jack (female) or plug (male) on the end of a cable. A cable connector connects wires on a cable to specific leads on telephone or data equipment.

CAG
Coverage answer group

call appearance

1. For the attendant console, six buttons, labeled a–f, used to originate, receive, and hold calls. Two lights next to the button show the status of the call appearance.
2. For the voice terminal, a button labeled with an extension and used to place outgoing calls, receive incoming calls, or hold calls. Two lights next to the button show the status of the call appearance.

call-control capabilities
Capabilities (*Third Party Selective Hold, Third Party Reconnect, Third Party Merge*) that can be used in either of the Third Party Call Control ASE (cluster) subsets (Call Control and Domain Control).

Call Detail Recording (CDR)
A feature that uses software and hardware to record call data (same as CDRU).

Call Detail Recording utility (CDRU)
Software that collects, stores, optionally filters, and outputs call-detail records.

Call Management System (CMS)
An application, running on an adjunct processor, that collects information from an ACD unit. CMS enables customers to monitor and manage telemarketing centers by generating reports on the status of agents, splits, trunks, trunk groups, vectors, and VDNs, and enables customers to partially administer the ACD feature for a communications system.

call-reference value (CRV)
An identifier present in ISDN messages that associates a related sequence of messages. In ASAI, CRVs distinguish between associations.

call vector
A set of up to 15 vector commands to be performed for an incoming or internal call.

callback call
A call that automatically returns to a voice-terminal user who activated the Automatic Callback or Ringback Queuing feature.

call-waiting ringback tone
A low-pitched tone identical to ringback tone except that the tone decreases in the last 0.2 seconds (in the United States). Call-waiting ringback tone notifies the attendant that the Attendant Call Waiting feature is activate and that the called party is aware of the waiting call. Tones in international countries may sound different.

call work code
A number, up to 16 digits, entered by ACD agents to record the occurrence of customer-defined events (such as account codes, social security numbers, or phone numbers) on ACD calls.

CAMA
Centralized Automatic Message Accounting

carrier
An enclosed shelf containing vertical slots that hold circuit packs.

carried load
The amount of traffic served by traffic-sensitive facilities during a given interval.

CARR-POW
Carrier Port and Power Unit for AC Powered Systems

CAS
Centralized Attendant Service or Call Accounting System

CCS or hundred call seconds
A unit of call traffic. Call traffic for a facility is scanned every 100 seconds. If the facility is busy, it is assumed to have been busy for the entire scan interval. There are 3600 seconds per hour. The Roman numeral for 100 is the capital letter C. The abbreviation for call seconds is CS. Therefore, 100 call seconds is abbreviated CCS. If a facility is busy for an entire hour, then it is said to have been busy for 36 CCS. See also **Erlang.**

capability
A request or indication of an operation. For example, *Third Party Make Call* is a request for setting up a call; *event report* is an indication that an event has occurred.

capability group
Set of capabilities, determined by switch administration, that can be requested by an application. Capability groups denote association types. For example, *Call Control* is a type of association that allows certain functions (the ones in the capability group) to be performed over this type of association. Also referred to as administration groups or application service elements (ASEs).

CA-TSC
Call-Associated Temporary Signaling Connection

cause value
A value is returned in response to requests or in event reports when a denial or unexpected condition occurs. ASAI cause values fall into two coding standards: Coding Standard 0 includes any cause values that are part of AT&T and CCITT ISDN specifications; Coding standard 3 includes any other ASAI cause values. This document uses a notation for cause value where the coding standard for the cause is given first, then a slash, then the cause value. Example: CS0/100 is coding standard 0, cause value 100.

CBC
Call-by-call or coupled bonding conductor

CC
Country code

CCIS
Common-Channel Interoffice Signaling

CCITT
CCITT (Comitte Consultatif International Telephonique et Telegraphique), now called *International Telecommunications Union* (ITU). See International Telecommunications Union (ITU).

CCMS
Control-Channel Message Set

CCS
See CCS or hundred call seconds.

CCSA
Common-Control Switching Arrangement

CDM
Channel-division multiplexing

CDOS
Customer-dialed and operator serviced

CDR
See Call Detail Recording (CDR).

CDRP
Call Detail Record Poller

CDRR
Call Detail Recording and Reporting

CDRU
See Call Detail Recording utility (CDRU).

CEM
Channel-expansion multiplexing

center-stage switch (CSS)
The central interface between the processor port network and expansion port networks in a CSS-connected system.

central office (CO)
The location housing telephone switching equipment that provides local telephone service and access to toll facilities for long-distance calling.

central office (CO) codes
The first three digits of a 7-digit public-network telephone number in the United States.

central office (CO) trunk
A telecommunications channel that provides access from the system to the public network through the local CO.

CEPT1
European Conference of Postal and Telecommunications Rate 1

channel

1. A circuit-switched call.

2. A communications path for transmitting voice and data.
3. In wideband, all of the time slots (contiguous or noncontiguous) necessary to support a call. Example: an H0-channel uses six 64-kbps time slots.
4. A DS0 on a T1 or E1 facility not specifically associated with a logical circuit-switched call; analogous to a single trunk.

channel negotiation
The process by which the channel offered in the Channel Identification Information Element (CIIE) in the SETUP message is negotiated to be another channel acceptable to the switch that receives the SETUP message and ultimately to the switch that sent the SETUP. Negotiation is attempted only if the CIIE is encoded as *Preferred*. Channel negotiation is not attempted for wideband calls.

CI
Clock input

circuit
1. An arrangement of electrical elements through which electric current flows.
2. A channel or transmission path between two or more points.

circuit pack
A card on which electrical circuits are printed, and IC chips and electrical components are installed. A circuit pack is installed in a switch carrier.

CISPR
International Special Committee on Radio Interference

Class of Restriction (COR)
A feature that allows up to 64 classes of call-origination and call-termination restrictions for voice terminals, voice-terminal groups, data modules, and trunk groups. See also Class of Service (COS).

Class of Service (COS)
A feature that uses a number to specify if voice-terminal users can activate the Automatic Callback, Call Forwarding All Calls, Data Privacy, or Priority Calling features. See also Class of Restriction (COR).

cm
Centimeter

CM
Connection Manager

CMDR
Centralized Message Detail Recording

CMS
Call Management System

CO
See central office (CO).

common-control switching arrangement (CCSA)
A private telecommunications network using dedicated trunks and a shared switching center for interconnecting company locations.

communications system
The software-controlled processor complex that interprets dialing pulses, tones, and keyboard characters and makes the proper connections both within the system and external to the system.

The communications system itself consists of a digital computer, software, storage device, and carriers with special hardware to perform the connections. A communications system provides voice and data communications services, including access to public and private networks, for telephones and data terminals on a customer's premises. See also switch.

confirmation tone
A tone confirming that feature activation, deactivation, or cancellation has been accepted.

connectivity
The connection of disparate devices within a single system.

consider series
Consider commands are typically written in a set of two or more. This set of consider commands is called a consider series.

consider sequence
A consider series plus a *queue-to best*, *check-best*, or *reply-best* step is called a consider sequence.

console
See attendant console.

contiguous
Adjacent DS0s within one T1 or E1 facility or adjacent TDM or fiber time slots. The first and last TDM bus, DS0, or fiber time slots are not considered contiguous (no wraparound). For an E1 facility with a D-channel, DS0s 15 and 17 are considered contiguous.

control cabinet
See control carrier.

control carrier
A carrier in a multicarrier cabinet that contains the SPE circuit packs and, unlike an R5r control carrier, port circuit packs. Also called control cabinet in a single-carrier cabinet. See also switch-processing element (SPE).

controlled station
A station that is monitored and controlled via a domain-control association.

COR
See Class of Restriction (COR).

COS
See Class of Service (COS).

coverage answer group
A group of up to eight voice terminals that ring simultaneously when a call is redirected to it by Call Coverage. Any one of the group can answer the call.

coverage call
A call that is automatically redirected from the called party's extension to an alternate answering position when certain coverage criteria are met.

coverage path
The order in which calls are redirected to alternate answering positions.

coverage point
An extension or attendant group, VDN, or ACD split designated as an alternate answering position in a coverage path.

covering user
A person at a coverage point who answers a redirected call.

CP
Circuit pack

CPE
Customer-premises equipment

CPN
Called-party number

CPN/BN
Calling-party number/billing number

CPTR
Call-progress-tone receiver

CRC
Cyclical Redundancy Checking

critical-reliability system
A system that has the following duplicated items: control carriers, tone clocks, EI circuit packs, and cabling between port networks and center-stage switch in a CSS-connected system. See also duplicated common control, and duplication.

CSA
Canadian Safety Association

CSCC
Compact single-carrier cabinet

CSCN
Center-stage control network

CSD
Customer-service document

CSM
Centralized System Management

CSS
See center-stage switch (CSS).

CSSO
Customer Services Support Organization

CSU
Channel service unit

CTS
Clear to Send

CWC
See call work code.

D

DAC
1. Dial access code or Direct Agent Calling
2. See digital-to-analog converter (DAC).

data channel
A communications path between two points used to transmit digital signals.

data-communications equipment (DCE)
The equipment (usually a modem, data module, or packet assembler/disassembler) on the network side of a communications link that makes the binary serial data from the source or transmitter compatible with the communications channel.

data link
The configuration of physical facilities enabling end terminals to communicate directly with each other.

data module
An interconnection device between a BRI or DCP interface of the switch and data terminal equipment or data communications equipment.

data path
The end-to-end connection used for a data communications link. A data path is the combination of all elements of an interprocessor communication in a DCS.

data port
A point of access to a computer that uses trunks or lines for transmitting or receiving data.

data rate
See bit rate.

data service unit (DSU)
A device that transmits digital data on transmission facilities.

data terminal
An input/output (I/O) device that has either switched or direct access to a host computer or to a processor interface.

data terminal equipment (DTE)
Equipment consisting of the endpoints in a connection over a data circuit. In a connection between a data terminal and host, the terminal, the host, and their associated modems or data modules make up the DTE.

dB
Decibel

dBA
Decibels in reference to amperes.

dBrnC
Decibels above reference noise with C filter.

DC
Direct current

DCE
Data-communications equipment

D-channel backup
Type of backup used with Non-Facility Associated Signaling (NFAS). A primary D-channel provides signaling for an NFAS D-channel group (two or more PRI facilities). A second D-channel, on a separate PRI facility of the NFAS D-channel group, is designated as backup for the D-channel. Failure of the primary D-channel causes automatic transfer of call-control signaling to the backup D-channel. The backup becomes the primary D-channel. When the failed channel returns to service, it becomes the backup D-channel.

DCO
Digital central office

DCP
Digital Communications Protocol

DCS
Distributed Communications System

DDC
Direct Department Calling

DDD
Direct Distance Dialing

delay-dial trunk
A trunk that allows dialing directly into a communications system (digits are received as they are dialed).

denying a request
Sending a negative acknowledgement (NAK), done by sending an FIE with a *return error* component (and a cause value). It should not be confused with the denial event report that applies to calls.

designated voice terminal
The specific voice terminal to which calls, originally directed to a certain extension, are redirected. Commonly used to mean the forwarded-to terminal when Call Forwarding All Calls is active.

dial-repeating trunks
A PBX tie trunk that is capable of handling PBX station-signaling information without attendant assistance.

dial-repeating tie trunk
A tie trunk that transmits called-party addressing information between two communications systems.

DID
Direct Inward Dialing

digit conversion
A process used to convert specific dialed numbers into other dialed numbers.

digital
The representation of information by discrete steps. See also analog.

digital communications protocol (DCP)
A proprietary protocol used to transmit both digitized voice and digitized data over the same communications link. A DCP link is made up of two 64-kbps information (I-) channels and one 8-kbps signaling (S-) channel.

digital data endpoints
In DEFINITY ECS, devices such as the 510D terminal or the 515-type business communications terminal (BCT).

digital multiplexed interface (DMI)
An interface that provides connectivity between a communications system and a host computer or between two communications systems using DS1 24th-channel signaling. DMI provides 23 64-kbps data channels and 1 common-signaling channel over a twisted-pair connection. DMI is offered through two capabilities: bit-oriented signaling (DMI-BOS) and message-oriented signaling (DMI-MOS).

digital signal level 0 (DS0)
A single 64-kbps voice channel. A DS0 is a single 64-kbps channel in a T1 or E1 facility and consists of eight bits in a T1 or E1 frame every 125 microseconds.

digital signal level 1 (DS1)
A single 1.544-Mbps (United States) or 2.048-Mbps (outside the United States) digital signal carried on a T1 transmission facility. A DS1 converter complex consists of a pair, one at each end, of DS1 converter circuit packs and the associated T1/E1 facilities.

digital terminal data module (DTDM)
An integrated or adjunct data module that shares with a digital telephone the same physical port for connection to a communications system. The function of a DTDM is similar to that of a PDM and MPDM in that it converts RS-232C signals to DCP signals.

digital-to-analog converter (DAC)
A device that converts data in digital form to the corresponding analog signals. See also analog-to-digital converter (ADC).

digital transmission
A mode of transmission in which information to be transmitted is first converted to digital form and then transmitted as a serial stream of pulses.

digital trunk
A circuit that carries digital voice and/or digital data in a telecommunications channel.

DIOD
Direct Inward and Outward Dialing

direct agent
A feature, accessed only via ASAI, that allows a call to be placed in a split queue but routed only to a specific agent in that split. The call receives normal ACD call treatment (for example, announcements) and is measured as an ACD call while ensuring that a particular agent answers.

Direct Extension Selection (DXS)
A feature on an attendant console that allows an attendant direct access to voice terminals by pressing a group-select button and a DXS button.

Direct Inward Dialing (DID)
A feature that allows an incoming call from the public network (not FX or WATS) to reach a specific telephone without attendant assistance.

Direct Inward Dialing (DID) trunk
An incoming trunk used for dialing directly from the public network into a communications system without help from the attendant.

disk drive
An electromechanical device that stores data on and retrieves data from one or more disks.

distributed communications system (DCS)
A network configuration linking two or more communications systems in such a way that selected features appear to operate as if the network were one system.

DIVA
Data In/Voice Answer

DLC
Data line circuit

DLDM
Data-line data module

DMI
Digital-multiplexed interface

DND
Do not disturb

DNIS
Dialed-Number Identification Service

DOD
Direct Outward Dialing

domain
VDNs, ACD splits, and stations. The VDN domain is used for active-notification associations. The ACD-split domain is for active-notification associations and domain-control associations. The station domain is used for the domain-control associations.

domain-control association
A *Third Party Domain Control Request* capability initiates a unique CRV/link number combination, which is referred to as a domain-control association.

domain-controlled split
A split for which *Third Party Domain Control* request has been accepted. A domain-controlled split provides an event report for logout.

domain-controlled station
A station for which a *Third_Party_Domain_Control* request has been accepted. A domain-controlled station provides event reports for calls that are alerting, connected, or held at the station.

domain-controlled station on a call
A station that is active on a call, and which provides event reports over one or two domain-control associations.

DOSS
Delivery Operations Support System

DOT
Duplication Option Terminal

DPM
Dial Plan Manager

DPR
Dual-port RAM

DS1
Digital Signal Level 1

DS1C
Digital Signal Level-1 protocol C

DS1 CONV
Digital Signal Level-1 converter

DSI
Digital signal interface

DSU
Data service unit

DTDM
Digital-terminal data module

DTE
Data-terminal equipment

DTGS
Direct Trunk Group Select

DTMF
Dual-tone multifrequency

DTS
Disk-tape system

duplicated common control
Two processors ensuring continuous operation of a communications system. While one processor is online, the other functions as a backup. The backup processor goes online periodically or when a problem occurs.

duplication
The use of redundant components to improve availability. When a duplicated subsystem fails, its backup redundant system automatically takes over.

duplication option
A system option that duplicates the following: control carrier containing the SPE, EI circuit packs in carriers, fiber-optic cabling between port networks, and center-stage switch in a CSS-connected system.

DWBS
DEFINITY Wireless Business System

DXS
Direct extension selection

E

E1
A digital transmission standard that carries traffic at 2.048 Mbps. The E1 facility is divided into 32 channels (DS0s) of 64 kbps information. Channel 0 is reserved for framing and synchronization information. A D-channel occupies channel 16.

E & M
Ear and mouth (receive and transmit)

EA
Expansion archangel

EAL
Expansion archangel link

ear and mouth (E & M) signaling
Trunk supervisory signaling, used between two communications systems, whereby signaling information is transferred through 2-state voltage conditions (on the E and M leads) for analog applications and through a single bit for digital applications.

EBCDIC
Extended Binary-Coded Decimal Interexchange Code

ECC
Error Correct Code

ECMA
European Computer Manufacturers Association

EFP
Electronic power feed

EI
Expansion interface

EIA
Electronic Industries Association

EIA-232
A physical interface specified by the EIA. EIA-232 transmits and receives asynchronous data at speeds of up to 19.2 kbps over cable distances of up to 50 feet. EIA-232 replaces RS-232 protocol in some DEFINITY applications.

electronic tandem network (ETN)
A tandem tie-trunk network that has automatic call-routing capabilities based on the number dialed and the most preferred route available. Each switch in the network is assigned a unique private network office code (RNX), and each voice terminal is assigned a unique extension.

Electronics Industries Association (EIA)
A trade association of the electronics industry that establishes electrical and functional standards.

eligible queue
A portion of queued calls that are candidates to be interflowed in enhanced Look-Ahead Interflow.

emergency transfer
If a major system failure occurs, automatic transfer is initiated to a group of telephones capable of making outgoing calls. The system operates in this mode until the failure is repaired and the system automatically returns to normal operation. Also called power-failure transfer.

EMI
Electromagnetic interference

end-to-end signaling
The transmission of touch-tone signals generated by dialing from a voice terminal to remote computer equipment. These digits are sent over the trunk as DTMF digits whether the trunk signaling type is marked as tone or rotary and whether the originating station is tone or rotary.

Example: a call to a voice-mail machine or automated-attendant service. A connection is first established over an outgoing trunk. Then additional digits are dialed to transmit information to be processed by the computer equipment.

enhanced private-switched communications service (EPSCS)
An analog private telecommunications network based on the No. 5 crossbar and 1A ESS that provides advanced voice and data telecommunications services to companies with many locations.

EPN
Expansion-port network

EPROM
Erasable programmable read-only memory

EPSCS
Enhanced Private Switched Communications Services

ERL
Echo return loss

Erlang
A unit of traffic intensity, or load, used to express the amount of traffic needed to keep one facility busy for one hour. One Erlang is equal to 36 CCS. See also CCS or hundred call seconds.

ESF
Extended superframe format

ESPA
European Standard Paging Access

ETA
Extended Trunk Access; also Enhanced Terminal Administration

ETN
Electronic tandem network

ETSI
European Telecommunications Standards Institute

EWT
See expected wait time (EWT).

expansion archangel (EAA)
A network-control microprocessor located on an expansion interface (EI) port circuit pack in an expansion port network. The EA provides an interface between the EPN and its controlling switch-processing element.

expansion-archangel link (EAL)
A link-access function on the D-channel (LAPD) logical link that exists between a switch-processing element and an expansion archangel (EA). The EAL carries control messages from the SPE to the EA and to port circuit packs in an expansion port network.

expansion control cabinet
See expansion control carrier.

expansion control carrier
A carrier in a multicarrier cabinet that contains extra port circuit packs and a maintenance interface. Also called expansion control cabinet in a single-carrier cabinet.

expansion interface (EI)
A port circuit pack in a port network that provides the interface between a PN's TDM bus/ packet bus and a fiber-optic link. The EI carries circuit-switched data, packet-switched data, network control, timing control, and DS1 control. In addition, an EI in an expansion port network communicates with the master maintenance circuit pack to provide the EPN's environmental and alarm status to the switch-processing element.

expansion port network (EPN)
A port network (PN) that is connected to the TDM bus and packet bus of a processor port network (PPN). Control is achieved by indirect connection of the EPN to the PPN via a port-network link (PNL). See also port network (PN).

expected wait time (EWT)
Prediction of how long a call will wait in queue before it is answered.

extension-in
Extension-In (ExtIn) is the work state agents go into when they answer (receive) a non-ACD call. If the agent is in Manual-In or Auto-In and receives an extension-in call, it is recorded by CMS as an AUX-In call.

extension-out
The work state that agents go into when they place (originate) a non-ACD call.

external measurements
Those ACD measurements that are made by the External CMS adjunct.

extension
A 1- to 5-digit number by which calls are routed through a communications system or, with a Uniform Dial Plan (UDP) or main-satellite dialing plan, through a private network.

external call
A connection between a communications system user and a party on the public network or on another communications system in a private network.

F

FAC
Feature Access Code

facility
A telecommunications transmission pathway and associated equipment.

facility-associated signaling (FAS)
Signaling for which a D-channel carries signaling only for those channels on the same physical interface.

FAS
Facility-associated signaling

FAT
Facility access trunk

FAX
Facsimile

FCC
Federal Communications Commission

FEAC
Forced Entry of Account Codes

feature
A specifically defined function or service provided by the system.

feature button
A labeled button on a telephone or attendant console used to access a specific feature.

FEP
Front-end processor

FIC
Facility interface codes

fiber optics
A technology using materials that transmit ultrawideband electromagnetic light-frequency ranges for high-capacity carrier systems.

fixed
A trunk allocation term. In the fixed allocation scheme, the time slots necessary to support a wideband call are contiguous, and the first time slot is constrained to certain starting points.

flexible
A trunk allocation term. In the flexible allocation scheme, the time slots of a wideband call can occupy noncontiguous positions within a single T1 or E1 facility.

floating
A trunk allocation term. In the floating allocation scheme, the time slots of a wideband call are contiguous, but the position of the first time slot is not fixed.

FNPA
Foreign Numbering-Plan Area

foreign-exchange (FX)
A CO other than the one providing local access to the public telephone network.

foreign-exchange trunk
A telecommunications channel that directly connects the system to a CO other than its local CO.

foreign numbering-plan area code (FNPAC)
An area code other than the local area code, that must be dialed to call outside the local geographical area.

FRL
Facilities Restriction Level

FX
Foreign exchange

G

G3-MA
Generic 3 Management Applications

G3-MT
Generic 3 Management Terminal

G3r
Generic 3, RISC (Reduced Instruction Set Computer)

generalized route selection (GRS)
An enhancement to Automatic Alternate Routing/Automatic Route Selection (AAR/ARS) that performs routing based on call attributes, such as Bearer Capability Classes (BCCs), in addition to the address and facilities restriction level (FRL), thus facilitating a Uniform Dial Plan (UDP) that is independent of the type of call being placed.

glare
The simultaneous seizure of a 2-way trunk by two communications systems, resulting in a standoff.

GM
Group manager

GPTR
General-purpose tone receiver

grade of service
The number of call attempts that fail to receive service immediately. Grade of service is also expressed as the quantity of all calls that are blocked or delayed.

ground-start trunk
A trunk on which, for outgoing calls, the system transmits a request for services to a distant switching system by grounding the trunk ring lead. To receive the digits of the called number, that system grounds the trunk tip lead. When the system detects this ground, the digits are sent.

GRS
Generalized Route Selection

H

H0
An ISDN information transfer rate for 384-kbps data defined by CCITT and ANSI standards.

H11
An ISDN information transfer rate for 1536-kbps data defined by CCITT and ANSI standards.

H12
An ISDN information transfer rate for 1920-kbps data defined by CCITT and ANSI standards.

handshaking logic
A format used to initiate a data connection between two data module devices.

hertz (Hz)
A unit of frequency equal to one cycle per second.

high-reliability system
A system having the following: two control carriers, duplicate expansion interface (EI) circuit packs in the PPN (in R5r with CSS), and duplicate switch node clock circuit packs in the switch node (SN) carriers. See also duplicated common control, duplication, duplication option, and critical-reliability system.

HNPA
See home numbering-plan area code (HNPA).

holding time
The total length of time in minutes and seconds that a facility is used during a call.

home numbering-plan area code (HNPA)
The local area code. The area code does not have to be dialed to call numbers within the local geographical area.

hop
Nondirect communication between two switch communications interfaces (SCI) where the SCI message passes automatically without intermediate processing through one or more intermediate SCIs.

host computer
A computer, connected to a network, that processes data from data-entry devices.

hunt group
A group of extensions that are assigned the Station Hunting feature so that a call to a busy extension reroutes to an idle extension in the group. See also ACD work mode.

Hz
See hertz (Hz).

I

I1
The first information channel of DCP.

I2
The second information channel of DCP.

I2 Interface
A proprietary interface used for the DEFINITY Wireless Business System for the radio-controller circuit packs. Each interface provides communication between the radio-controller circuit pack and up to two wireless fixed bases.

I3 Interface
A proprietary interface used for the DEFINITY Wireless Business System for the cell antenna units. Each wireless fixed base can communicate to up to four cell antenna units.

IAS
Inter-PBX Attendant Service

ICC
Intercabinet cable or intercarrier cable

ICD
Inbound Call Director

ICDOS
International Customer-Dialed Operator Service

ICHT
Incoming call-handling table

ICI
Incoming call identifier

ICM
Inbound Call Management

IDDD
International Direct Distance Dialing

IDF
Intermediate distribution frame

IE
Information element

immediate-start tie trunk
A trunk on which, after making a connection with a distant switching system for an outgoing call, the system waits a nominal 65 ms before sending the digits of the called number. This allows time for the distant system to prepare to receive digits. On an incoming call, the system has less than 65 ms to prepare to receive the digits.

IMT
Intermachine trunk

in
Inch

INADS
Initialization and Administration System

information forwarding
A process for transporting call data along with the call.

incoming gateway
A PBX that routes an incoming call on a trunk *not* administered for Supplementary Services Protocol B to a trunk *not* administered for Supplementary Services Protocol B.

information exchange
The exchange of data between users of two different systems, such as the switch and a host computer, over a LAN.

Information Systems Network (ISN)
A WAN and LAN with an open architecture combining host computers, minicomputers, word processors, storage devices, PCs, high-speed printers, and nonintelligent terminals into a single packet-switching system.

INS
ISDN Network Service

inside call
A call placed from one telephone to another within the local communications system.

Integrated Services Digital Network (ISDN)
A public or private network that provides end-to-end digital communications for all services to which users have access by a limited set of standard multipurpose user-network interfaces defined by the CCITT. Through internationally accepted standard interfaces, ISDN provides digital circuit-switched or packet-switched communications within the network and links to other ISDNs to provide national and international digital communications. See also Integrated Services Digital Network Basic Rate Interface (ISDN-BRI) and Integrated Services Digital Network Primary Rate Interface (ISDN-PRI).

Integrated Services Digital Network Basic Rate Interface (ISDN-BRI)
The interface between a communications system and terminal that includes two 64-kbps B-channels for transmitting voice or data and one 16-kbps D-channel for transmitting associated B-channel call control and out-of-band signaling information. ISDN-BRI also includes 48 kbps for transmitting framing and D-channel contention information, for a total interface speed of 192 kbps. ISDN-BRI serves ISDN terminals and digital terminals fitted with ISDN terminal adapters. See also Integrated Services Digital Network (ISDN) and Integrated Services Digital Network Primary Rate Interface (ISDN-PRI).

Integrated Services Digital Network Primary Rate Interface (ISDN-PRI)
The interface between multiple communications systems that in North America includes 24 64-kbps channels, corresponding to the North American digital signal level-1 (DS1) standard rate of 1.544 Mbps. The most common arrangement of channels in ISDN-PRI is 23 64-kbps B-channels for transmitting voice and data and 1 64-kbps D-channel for transmitting associated B-channel call control and out-of-band signaling information. With nonfacility-associated signaling (NFAS), ISDN-PRI can include 24 B-channels and no D-channel. See also Integrated Services Digital Network (ISDN) and Integrated Services Digital Network Basic Rate Interface (ISDN-BRI).

intelligent polling
See polling, intelligent.

intercept tone
A tone that indicates a dialing error or denial of the service requested.

interface
A common boundary between two systems or pieces of equipment.

internal call
A connection between two users within a system.

International Telecommunications Union (ITU)
Formerly known as International Telegraph and Telephone Consultative Committee (CCITT), ITU is an international organization that sets universal standards for data communications, including ISDN. ITU members are from telecommunications companies and organizations around the world. See also BX.25.

International Telegraph and Telephone Consultative Committee
See International Telecommunications Union (ITU).

interflow
To route an incoming call to an external switch without answering it at the origin switch.

intraflow
The ability for calls to redirect to other splits on the same PBX on a conditional or unconditional basis using call coverage busy, don't answer, or all criteria.

internal measurements
BCMS measurements that are made by the system. ACD measurements that are made external to the system (via External CMS) are referred to as external measurements.

in-use lamp
A red light on a multiappearance voice terminal that lights to show which call appearance will be selected when the handset is lifted or which call appearance is active when a user is off-hook.

INWATS
Inward Wide Area Telephone Service

IO
Information outlet

ISDN
See Integrated Services Digital Network (ISDN).

ISDN Gateway (IG)
A feature allowing integration of the switch and a host-based telemarketing application via a link to a gateway adjunct. The gateway adjunct is a 3B-based product that notifies the host-based telemarketing application of call events.

ISDN trunk
A trunk administered for use with ISDN-PRI. Also called ISDN facility.

ISDN-PRI terminal adapter
An interface between endpoint applications and an ISDN PRI facility. ISDN-PRI terminal adapters are currently available from other vendors and are primarily designed for video conferencing applications. Accordingly, currently available terminal adapters adapt the two pairs of video codec data (V.35) and dialing (RS-366) ports to an ISDN PRI facility.

IS/DTT
Integrated Services/digital tie trunk

ISN
Information Systems Network

ISO
International Standards Organization

ISV
Independent software vendor

ITP
Installation test procedure

ITU
International Telecommunications Union

IXC
Interexchange carrier code

K

kHz
Kilohertz

kbps
Kilobits per second

kbyte
Kilobyte

kg
Kilogram

L

LAN
Local area network

LAP-D
Link Access Procedure on the D-channel

LAPD
Link Access Procedure data

LATA
Local access and transport area

lb
Pound

LBO
Line buildout

LDN
Listed directory number

LDS
Long-distance service

LEC
Local exchange carrier

LED
See light-emitting diode (LED).

light-emitting diode (LED)
A semiconductor device that produces light when voltage is applied. LEDs provide a visual indication of the operational status of hardware components, the results of maintenance tests, the alarm status of circuit packs, and the activation of telephone features.

lightwave transceiver
Hardware that provides an interface to fiber-optic cable from port circuit packs and DS1 converter circuit packs. Lightwave transceivers convert electrical signals to light signals and vice versa.

line
A transmission path between a communications system or CO switching system and a voice terminal or other terminal.

line appearance
See appearance.

line buildout
A selectable output attenuation is generally required of DTE equipment because T1 circuits require the last span to lose 15–22.5 dB.

line port
Hardware that provides the access point to a communications system for each circuit associated with a telephone or data terminal.

link
A transmitter-receiver channel that connects two systems.

link-access procedure on the D-channel (LAPD)
A link-layer protocol on the ISDN-BRI and ISDN-PRI data-link layer (level 2). LAPD provides data transfer between two devices, and error and flow control on multiple logical links. LAPD is used for signaling and low-speed packet data (X.25 and mode 3) on the signaling (D-) channel and for mode-3 data communications on a bearer (B-) channel.

LINL
Local indirect neighbor link

local area network (LAN)
A networking arrangement designed for a limited geographical area. Generally, a LAN is limited in range to a maximum of 6.2 miles and provides high-speed carrier service with low error rates. Common configurations include daisy chain, star (including circuit-switched), ring, and bus.

logical link
The communications path between a processor and a BRI terminal.

loop-start trunk
A trunk on which, after establishing a connection with a distant switching system for an outgoing call, the system waits for a signal on the loop formed by the trunk leads before sending the digits of the called number.

LSU
Local storage unit

LWC
Leave Word Calling

M

MAC
Medium access

MADU
Modular asynchronous data unit

main distribution frame (MDF)
A device that mounts to the wall inside the system equipment room. The MDF provides a connection point from outside telephone lines to the PBX switch and to the inside telephone stations.

main-satellite-tributary
A private network configuration that can either stand alone or access an ETN. A main switch provides interconnection, via tie trunks, with one or more subtending switches, called satellites; all attendant positions for the main/satellite configuration; and access to and from the public network. To a user outside the complex, a main/satellite configuration appears as one switch, with one listed directory number (LDN). A tributary switch is connected to the main switch via tie trunks, but has its own attendant positions and LDN.

maintenance
Activities involved in keeping a telecommunications system in proper working condition: the detection and isolation of software and hardware faults, and automatic and manual recovery from these faults.

management terminal
The terminal that is used by the system administrator to administer the switch. The terminal may also be used to access the BCMS feature.

major alarm
An indication of a failure that has caused critical degradation of service and requires immediate attention. Major alarms are automatically displayed on LEDs on the attendant console and maintenance or alarming circuit pack, logged to the alarm log, and reported to a remote maintenance facility, if applicable.

Manual-In work mode
One of four agent work modes: the mode in which an agent is ready to process another call manually. *See* Auto-In Work mode for a contrast.

MAP
Maintenance action process

MAPD
Multiapplication platform for DEFINITY

MA-UUI
Message-Associated User-to-User Signaling

Mbps
Megabits per second

M-Bus
Memory bus

Mbyte
Megabyte

MCC
Multicarrier cabinet

MCS
Message Center Service

MCT
Malicious Call Trace

MCU
Multipoint control unit

MDF
Main distribution frame

MDM
Modular data module

MDR
Message detail record

MEM
Memory

memory
A device into which information can be copied and held, and from which information can later be obtained.

memory shadowing link
An operating-system condition that provides a method for memory-resident programs to be more quickly accessed, allowing a system to reboot faster.

message center
An answering service that supplies agents to and stores messages for later retrieval.

message center agent
A member of a message-center hunt group who takes and retrieves messages for voice-terminal users.

MET
Multibutton electronic telephone

MF
Multifrequency

MFB
Multifunction board

MFC signaling
Multifrequency-compelled signaling

MHz
Megahertz

MIM
Management information message

minor alarm
An indication of a failure that could affect customer service. Minor alarms are automatically displayed on LEDs on the attendant console and maintenance or alarming circuit pack, sent to the alarm log, and reported to a remote maintenance facility, if applicable.

MIPS
Million instructions per second

MIS
Management information system

MISCID
Miscellaneous identification

MMCS
Multimedia Call Server

MMCH
Multimedia call handling

MMI
Multimedia interface

MMS
Material Management Services

MO
Maintenance object

modem
A device that converts digital data signals to analog signals for transmission over telephone circuits. The analog signals are converted back to the original digital data signals by another modem at the other end of the circuit.

modem pooling
A capability that provides shared conversion resources (modems and data modules) for cost-effective access to analog facilities by data terminals. When needed, modem pooling inserts a conversion resource into the path of a data call. Modem pooling serves both outgoing and incoming calls.

modular processor data module (MPDM)
A processor data module (PDM) that can be configured to provide several kinds of interfaces (RS-232C, RS-449, and V.35) to customer-provided data terminal equipment (DTE). See also processor data module (PDM).

modular trunk data module (MTDM)
A trunk data module that can be configured to provide several kinds of interfaces (RS-232, RS-449, and V.35) to customer-provided data terminal equipment.

modulator-demodulator
See modem.

monitored call
See active-notification call.

MOS
Message-oriented signaling

MPDM
Modular processor data module

MS
Message server

ms
Millisecond

MS/T
Main satellite/tributary

MSA
Message servicing adjunct

MSG
Message service

MSL
Material stocking location

MSM
Modular System Management

MSS
Mass storage system

MSSNET
Mass storage/network control

MT
Management terminal

MTDM
Modular trunk data module

MTP
Maintenance tape processor

MTT
Multitasking terminal

multiappearance voice terminal
A terminal equipped with several call-appearance buttons for the same extension, allowing the user to handle more than one call on that same extension at the same time.

Multicarrier cabinet
A structure that holds one to five carriers. See also single-carrier cabinet.

Multifrequency Compelled (MFC) Release 2 (R2) signaling
A signal consisting of two frequency components, such that when a signal is transmitted from a switch, another signal acknowledging the transmitted signal is received by the switch. R2 designates signaling used in the United States and in countries outside the United States.

multiplexer
A device used to combine a number of individual channels into a single common bit stream for transmission.

multiplexing
A process whereby a transmission facility is divided into two or more channels, either by splitting the frequency band into a number of narrower bands or by dividing the transmission channel into successive time slots. See also time-division multiplexing (TDM).

multirate
The new N x DS0 service (see N x DS0).

MWL
Message-waiting lamp

N

N+1
Method of determining redundant backup requirements. Example: if four rectifier modules are required for a DC-powered single-carrier cabinet, a fifth rectifier module is installed for backup.

N x DS0
N x DS0, equivalently referred to as N x 64 kbps, is an emerging standard for wideband calls separate from H0, H11, and H12 ISDN channels. The emerging N x DS0 ISDN multirate circuit mode bearer service will provide circuit-switched calls with data-rate multiples of 64 kbps up to 1536 kbps on a T1 facility or up to 1920 kbps on an E1 facility. In the switch, N x DS0 channels will range up to 1984 kbps using NFAS E1 interfaces.

NANP
North American Numbering Plan

narrowband
A circuit-switched call at a data rate up to and including 64 kbps. All nonwideband switch calls are considered narrowband.

native terminal support
A predefined terminal type exists in switch software, eliminating the need to alias the terminal (that is, manually map call appearances and feature buttons onto some other natively supported terminal type).

NAU
Network access unit

NCA/TSC
Noncall-associated/temporary-signaling connection

NCOSS
Network Control Operations Support Center

NCSO
National Customer Support Organization

NEC
National Engineering Center

NEMA
National Electrical Manufacturer's Association

NETCON
Network-control circuit pack

network
A series of points, nodes, or stations connected by communications channels.

network-specific facility (NSF)
An information element in an ISDN-PRI message that specifies which public-network service is used. NSF applies only when Call-by-Call Service Selection is used to access a public-network service.

network interface
A common boundary between two systems in an interconnected group of systems.

NFAS
See Nonfacility-associated signaling (NFAS).

NI
Network interface

NID
Network Inward Dialing

NM
Network management

NN
National number

node
A switching or control point for a network. Nodes are either tandem (they receive signals and pass them on) or terminal (they originate or terminate a transmission path).

Nonfacility-associated signaling (NFAS)
A method that allows multiple T1 and/or E1 facilities to share a single D-channel to form an ISDN-PRI. If D-channel backup is not used, one facility is configured with a D-channel, and the other facilities that share the D-channel are configured without D-channels. If D-channel backup is used, two facilities are configured to have D-channels (one D-channel on each facility), and the other facilities that share the D-channels are configured without D-channels.

NPA
Numbering-plan area

NPE
Network processing element

NQC
Number of queued calls

NSE
Night-service extension

NSU
Network sharing unit

null modem cable
Special wiring of an RS-232-C cable such that a computer can talk to another computer (or to a printer) without a modem.

NXX
Public-network office code

O

OA
Operator assisted

occurrence
See appearance.

OCM
Outbound Call Management

offered load
The traffic that would be generated by all the requests for service occurring within a monitored interval, usually one hour.

ONS
On-premises station

OPS
Off-premises station

OPX
Off-premises extension

OQT
Oldest queued time

OSHA
Occupational Safety and Health Act

OSI
Open Systems Interconnect

OSS
Operations Support System

OSSI
Operational Support System Interface

OTDR
Optical time-domain reflectometer

othersplit
The work state that indicates that an agent is currently active on another split's call, or in ACW for another split.

OTQ
Outgoing trunk queuing

outgoing gateway
A PBX that routes an incoming call on a trunk administered for Supplementary Services Protocol B to a trunk *not* administered for Supplementary Services Protocol B.

P

PACCON
Packet control

packet
A group of bits (including a message element, which is the data, and a control information element (IE), which is the header) used in packet switching and transmitted as a discrete unit. In each packet, the message element and control IE are arranged in a specified format. See also **packet bus** and **packet switching**.

packet bus
A wide-bandwidth bus that transmits packets.

packet switching
A data-transmission technique whereby user information is segmented and routed in discrete data envelopes called packets, each with its own appended control information, for routing, sequencing, and error checking. Packet switching allows a channel to be occupied only during the transmission of a packet. On completion of the transmission, the channel is made available for the transfer of other packets. See also BX.25and **packet**.

PAD
Packet assembly/disassembly

paging trunk
A telecommunications channel used to access an amplifier for loudspeaker paging.

party/extension active on call
A party is on the call if he or she is actually connected to the call (in active talk or in held state). An originator of a call is always a party on the call. Alerting parties, busy parties, and tones are not parties on the call.

PBX
Private branch exchange

PC
See personal computer (PC).

PCM
See pulse-code modulation (PCM).

PCOL
Personal central-office line

PCOLG
Personal central-office line group

PCS
Permanent switched calls

PDM
See processor data module (PDM).

PDS
Premises Distribution System

PE
Processing element

PEC
Price element code

PEI
Processor element interchange

personal computer (PC)
A personally controllable microcomputer.

PGATE
Packet gateway

PGN
Partitioned group number

PI
Processor interface

PIB
Processor interface board

pickup group
A group of individuals authorized to answer any call directed to an extension within the group.

PIDB
Product image database

PKTINT
Packet interface

PL
Private line

PLS
Premises Lightwave System

PMS
Property Management System

PN
Port network

PNA
Private network access

POE
Processor occupancy evaluation

poll
See status poll.

poll suppression
A component of BSR intelligent polling that eliminates wasteful polling of remote locations which have returned poor adjusted EWTs.

polling, intelligent
An automatic feature of Best Service Routing that significantly reduces the number of status polls executed. When a remote location cannot be the best resource at a given moment in time, the intelligent polling feature temporarily suppresses polls to that location.

POP
Point of presence

port
A data- or voice-transmission access point on a device that is used for communicating with other devices.

port carrier
A carrier in a multicarrier cabinet or a single-carrier cabinet containing port circuit packs, power units, and service circuits. Also called a port cabinet in a single-carrier cabinet.

port network (PN)
A cabinet containing a TDM bus and packet bus to which the following components are connected: port circuit packs, one or two tone-clock circuit packs, a maintenance circuit pack, service circuit packs, and (optionally) up to four expansion interface (EI) circuit packs in DEFINITY ECS. Each PN is controlled either locally or remotely by a switch processing element (SPE). See also expansion port network (EPN) and processor port network (PPN).

port-network connectivity
The interconnection of port networks (PNs), regardless of whether the configuration uses direct or switched connectivity.

PPM
1. Parts per million
2. Periodic pulse metering

PPN
See processor port network (PPN).

PRI
See Primary Rate Interface (PRI).

primary extension
The main extension associated with the physical voice or data terminal.

Primary Rate Interface (PRI)
A standard ISDN frame format that specifies the protocol used between two or more communications systems. PRI runs at 1.544 Mbps and, as used in North America, provides 23 64-kbps B-channels (voice or data) and one 64-kbps D-channel (signaling). The D-channel is the 24th channel of the interface and contains multiplexed signaling information for the other 23 channels.

PRI endpoint (PE)
The wideband switching capability introduces PRI endpoints on switch line-side interfaces. A PRI endpoint consists of one or more contiguous B-channels on a line-side T1 or E1 ISDN PRI facility and has an extension. Endpoint applications have call-control capabilities over PRI endpoints.

principal
A terminal that has its primary extension bridged on one or more other terminals.

principal (user)
A person to whom a telephone is assigned and who has message-center coverage.

private network
A network used exclusively for the telecommunications needs of a particular customer.

private network office code (RNX)
The first three digits of a 7-digit private network number.

PROCR
Processor

processor carrier
See control carrier.

processor data module (PDM)
A device that provides an RS-232C DCE interface for connecting to data terminals, applications processors (APs), and host computers, and provides a DCP interface for connection to a communications system. See also modular processor data module (MPDM).

processor port network (PPN)
A port network controlled by a switch-processing element that is directly connected to that PN's TDM bus and LAN bus. See also port network (PN).

processor port network (PPN) control carrier
A carrier containing the maintenance circuit pack, tone/clock circuit pack, and SPE circuit packs for a processor port network (PPN) and, optionally, port circuit packs.

Property Management System (PMS)
A stand-alone computer used by lodging and health-services organizations for services such as reservations, housekeeping, and billing.

protocol
A set of conventions or rules governing the format and timing of message exchanges to control data movement and correction of errors.

PSC
Premises service consultant

PSDN
Packet-switch public data network

PT
Personal terminal

PTC
Positive temperature coefficient

PTT
Postal Telephone and Telegraph

public network
The network that can be openly accessed by all customers for local and long-distance calling.

pulse-code modulation (PCM)
An extension of pulse-amplitude modulation (PAM) in which carrier-signal pulses modulated by an analog signal, such as speech, are quantized and encoded to a digital, usually binary, format.

Q

QPPCN
Quality Protection Plan Change Notice

quadrant
A group of six contiguous DS0s in fixed locations on an ISDN-PRI facility. Note that this term comes from T1 terminology (one-fourth of a T1), but there are five quadrants on an E1 ISDN-PRI facility (30B + D).

queue
An ordered sequence of calls waiting to be processed.

queuing
The process of holding calls in order of their arrival to await connection to an attendant, to an answering group, or to an idle trunk. Calls are automatically connected in first-in, first-out sequence.

R

RAM
See random-access memory (RAM).

random-access memory (RAM)
A storage arrangement whereby information can be retrieved at a speed independent of the location of the stored information.

RBS
Robbed-bit signaling

RC
Radio controller

RCL
Restricted call list

read-only memory (ROM)
A storage arrangement primarily for information-retrieval applications.

recall dial tone
Tones signalling that the system has completed a function (such as holding a call) and is ready to accept dialing.

redirection criteria
Information administered for each voice terminal's coverage path that determines when an incoming call is redirected to coverage.

Redirection on No Answer
An optional feature that redirects an unanswered ringing ACD call after an administered number of rings. The call is then redirected back to the agent.

remote home numbering-plan area code (RHNPA)
A foreign numbering-plan area code that is treated as a home area code by the Automatic Route Selection (ARS) feature. Calls can be allowed or denied based on the area code and the dialed CO code rather than just the area code. If the call is allowed, the ARS pattern used for the call is determined by these six digits.

Remote Operations Service Element (ROSE)
A CCITT and ISO standard that defines a notation and services that support interactions between the various entities that make up a distributed application.

REN
Ringer equivalency number

reorder tone
A tone to signal that at least one of the facilities, such as a trunk or a digit transmitter, needed for the call was not available.

report scheduler
Software that is used in conjunction with the system printer to schedule the days of the week and time of day that the desired reports are to be printed.

resource
A general term for an agent, split, skill, or location.

RFP
Request for proposal

RHNPA
See remote home numbering-plan area code (RHNPA).

RINL
Remote indirect neighbor link

RISC
Reduced-instruction-set computer

RLT
Release-link trunk

RMATS
Remote Maintenance, Administration, and Traffic System

RNX
Route-number index (private network office code)

ROM
See read-only memory (ROM).

RPN
Routing-plan number

RS-232C
A physical interface specified by the Electronic Industries Association (EIA). RS-232C transmits and receives asynchronous data at speeds of up to 19.2 kbps over cable distances of up to 50 feet.

RS-449
Recommended Standard 449

RSC
Regional Support Center

ROSE
See Remote Operations Service Element (ROSE).

S

S1
The first logical signalling channel of DCP. The channel is used to provide signaling information for DCP's I1 channel.

S2
The second logical signaling channel of DCP. The channel is used to provide signaling information for DCP's I2 channel.

SABM
Set Asynchronous Balance Mode

SAC
Send All Calls

SAKI
See sanity and control interface (SAKI).

sanity and control interface (SAKI)
A custom VLSI microchip located on each port circuit pack. The SAKI provides address recognition, buffering, and synchronization between the angel and the five control time slots that make up the control channel. The SAKI also scans and collects status information for the angel on its port circuit pack and, when polled, transmits this information to the archangel.

SAT
System access terminal

SCC
1. See single-carrier cabinet.
2. Serial communications controller

SCD
Switch-control driver

SCI
Switch communications interface

SCO
System control office

SCOTCH
Switch Conferencing for TDM Bus in Concentration Highway

SCSI
See small computer system interface (SCSI).

SDDN
Software-Defined Data Network

SDI
Switched Digital International

SDLC
Synchronous data-link control

SDN
Software-defined network

SFRL
Single-frequency return loss

SID
Station-identification number

simplex system
A system that has no redundant hardware.

simulated bridged appearance
The same as a temporary bridged appearance; allows the terminal user (usually the principal) to bridge onto a call that had been answered by another party on his or her behalf.

single-carrier cabinet
A combined cabinet and carrier unit that contains one carrier. See also Multicarrier cabinet.

single-line voice terminal
A voice terminal served by a single-line tip and ring circuit (models 500, 2500, 7101A, 7103A).

SIT
Special-information tones

small computer system interface (SCSI)
An ANSI bus standard that provides a high-level command interface between host computers and peripheral devices.

SMDR
Station Message Detail Recording, known as Call Detail Recording in DEFINITY ECS.

SN
Switch Node

SNA
Systems Network Architecture

SNC
Switch Node Clock

SNI
Switch Node Interface

SNMP
Simple Network Management Protocol

software
A set of computer programs that perform one or more tasks.

SPE
Switch Processing Element

SPID
Service Profile Identifier

split
See ACD work mode.

split condition
A condition whereby a caller is temporarily separated from a connection with an attendant. A split condition automatically occurs when the attendant, active on a call, presses the start button.

split number
The split's identity to the switch and BCMS.

split report
A report that provides historical traffic information for internally measured splits.

split (agent) status report
A report that provides real-time status and measurement data for internally measured agents and the split to which they are assigned.

SSI
Standard serial interface

SSM
Single-site management

SSV
Station service

ST3
Stratum 3 clock board

staffed
Indicates that an agent position is logged in. A staffed agent functions in one of four work modes: Auto-In, Manual-In, ACW, or AUX-Work.

STARLAN
Star-Based Local Area Network

Station Message Detail Recording (SMDR)
An obsolete term now called CDR — a switch feature that uses software and hardware to record call data. See Call Detail Recording (CDR).

standard serial interface (SSI)
A communications protocol developed for use with 500-type business communications terminals (BCTs) and 400-series printers.

status lamp
A green light that shows the status of a call appearance or a feature button by the state of the light (lit, flashing, fluttering, broken flutter, or unlit).

status poll
A call placed by a *consider location* vector command to obtain status data from a remote location in a multi-site BSR application.

stroke counts
A method used by ACD agents to record up to nine customer-defined events per call when CMS is active.

SVN
Security-violation notification

switch
Any kind of telephone switching system. See also communications system.

switchhook
The buttons located under the receiver on a voice terminal.

switch-node (SN) carrier
A carrier containing a single switch node, power units, and, optionally, one or two DS1 converter circuit packs. An SN carrier is located in a center-stage switch.

switch-node (SN) clock
The circuit pack in an SN carrier that provides clock and maintenance alarm functions and environmental monitors.

switch-node interface (SNI)
The basic building block of a switch node. An SNI circuit pack controls the routing of circuit, packet, and control messages.

switch-node link (SNL)
The hardware that provides a bridge between two or more switch nodes. The SNL consists of the two SNI circuit packs residing on the switch nodes and the hardware connecting the SNIs. This hardware can include lightwave transceivers that convert the SNI's electrical signals to light signals, the copper wire that connects the SNIs to the lightwave transceivers, a full-duplex fiber-optic cable, DS1 converter circuit cards and DS1 facilities if a company does not have rights to lay cable, and appropriate connectors.

switch-processing element (SPE)
A complex of circuit packs (processor, memory, disk controller, and bus-interface cards) mounted in a PPN control carrier. The SPE serves as the control element for that PPN and, optionally, for one or more EPNs.

SXS
Step-by-step

synchronous data transmission
A method of sending data in which discrete signal elements are sent at a fixed and continuous rate and specified times. See also association.

SYSAM
System Access and Administration

system administrator
The person who maintains overall customer responsibility for system administration. Generally, all administration functions are performed from the Management Terminal. The switch requires a

special login, referred to as the system administrator login, to gain access to system-administration capabilities.

system printer
An optional printer that may be used to print scheduled reports via the report scheduler.

system report
A report that provides historical traffic information for internally measured splits.

system-status report
A report that provides real-time status information for internally measured splits.

system manager
A person responsible for specifying and administering features and services for a system.

system reload
A process that allows stored data to be written from a tape into the system memory (normally after a power outage).

T

T1
A digital transmission standard that in North America carries traffic at the DS1 rate of 1.544 Mbps. A T1 facility is divided into 24 channels (DS0s) of 64 kbps. These 24 channels, with an overall digital rate of 1.536 Mbps, and an 8-kbps framing and synchronization channel make up the 1.544-Mbps transmission. When a D-channel is present, it occupies channel 24. T1 facilities are also used in Japan and some Middle-Eastern countries.

TAAS
Trunk Answer from Any Station

TABS
Telemetry asynchronous block serial

TAC
Trunk-access code

tandem switch
A switch within an electronic tandem network (ETN) that provides the logic to determine the best route for a network call, possibly modifies the digits outpulsed, and allows or denies certain calls to certain users.

tandem through
The switched connection of an incoming trunk to an outgoing trunk without human intervention.

tandem tie-trunk network (TTTN)
A private network that interconnects several customer switching systems.

TC
Technical consultant

TCM
Traveling class mark

TDM
See time-division multiplexing (TDM).

TDR
Time-of-day routing

TEG
Terminating extension group

terminal
A device that sends and receives data within a system. See also administration terminal.

tie trunk
A telecommunications channel that directly connects two private switching systems.

time-division multiplex (TDM) bus
A bus that is time-shared regularly by preallocating short time slots to each transmitter. In a PBX, all port circuits are connected to the TDM bus, permitting any port to send a signal to any other port.

time-division multiplexing (TDM)
Multiplexing that divides a transmission channel into successive time slots. See also multiplexing.

time interval
The period of time, either one hour or one-half hour, that BCMS measurements are collected for a reports.

time slice
See **time interval**.

time slot
64 kbps of digital information structured as eight bits every 125 microseconds. In the switch, a time slot refers to either a DS0 on a T1 or E1 facility or a 64-kbps unit on the TDM bus or fiber connection between port networks.

time slot sequence integrity
The situation whereby the N octets of a wideband call that are transmitted in one T1 or E1 frame arrive at the output in the same order that they were introduced.

to control
An application can invoke *Third Party Call Control* capabilities using either an adjunct-control or domain-control association.

to monitor
An application can receive *event reports* on an active-notification, adjunct-control, or domain-control association.

TOD
Time of day

tone ringer
A device with a speaker, used in electronic voice terminals to alert the user.

TOP
Task-oriented protocol

trunk
A dedicated telecommunications channel between two communications systems or COs.

trunk allocation
The manner in which trunks are selected to form wideband channels.

trunk-data module
A device that connects off-premises private-line trunk facilities and DEFINITY ECS. The trunk-data module converts between the RS-232C and the DCP, and can connect to DDD modems as the DCP member of a modem pool.

trunk group
Telecommunications channels assigned as a group for certain functions that can be used interchangeably between two communications systems or COs.

TSC
Technical Service Center

TTI
Terminal translation initialization

TTR
Touch-tone receiver

TTT
Terminating trunk transmission

TTTN
See tandem tie-trunk network (TTTN).

TTY
Teletypewriter

U

UAP
Usage-allocation plan

UART
Universal asynchronous transmitter

UCD
Uniform call distribution

UCL
Unrestricted call list

UDP
See Uniform Dial Plan (UDP).

UL
Underwriter Laboratories

UM
User manager

Uniform Dial Plan (UDP)
A feature that allows a unique 4- or 5-digit number assignment for each terminal in a multiswitch configuration such as a DCS or main-satellite-tributary system.

UNMA
Unified Network Management Architecture

UNP
Uniform numbering plan

UPS
Uninterruptible power supply

User-toUser Information
Call data that DEFINITY forwards with the call.

USOP
User service-order profile

UUCP
UNIX-to-UNIX Communications Protocol

UUI
User-to-user information

V

VAR
Value-added reseller

VDN
See vector directory number (VDN).

vector directory number (VDN)
An extension that provides access to the Vectoring feature on the switch. Vectoring allows a customer to specify the treatment of incoming calls based on the dialed number.

vector-controlled split
A hunt group or ACD split administered with the vector field enabled. Access to such a split is possible only by dialing a VDN extension.

VIS
Voice Information System

VLSI
Very-large-scale integration

VM
Voltmeter

VNI
Virtual nodepoint identifier

voice terminal
A single-line or multiappearance telephone.

W

WATS
See Wide Area Telecommunications Service (WATS).

WCC
World-Class Core

WCR
World-Class Routing

WCTD
World-Class Tone Detection

WFB
Wireless fixed base

Wide Area Telecommunications Service (WATS)
A service in the United States that allows calls to certain areas for a flat-rate charge based on expected usage.

wideband
A circuit-switched call at a data rate greater than 64 kbps. A circuit-switched call on a single T1 or E1 facility with a bandwidth between 128 and 1536 (T1) or 1984 (E1) kbps in multiples of 64 kbps. H0, H11, H12, and N x DS0 calls are wideband.

wideband access endpoint
Access endpoints, extended with wideband switching to include wideband access endpoints. A wideband access endpoint consists of one or more contiguous DS0s on a line-side T1 or E1 facility and has an extension. The Administered Connections feature provides call control for calls originating from wideband access endpoints.

wink-start tie trunk
A trunk with which, after making a connection with a distant switching system for an outgoing call, the system waits for a momentary signal (wink) before sending the digits of the called number. Similarly, on an incoming call, the system sends the wink signal when ready to receive digits.

work mode
One of four states (Auto-In, Manual-In, ACW, AUX-Work) that an ACD agent can be in. Upon logging in, an agent enters AUX-Work mode. To become available to receive ACD calls, the agent enters Auto-In or Manual-In mode. To do work associated with a completed ACD call, an agent enters ACW mode.

work state
An ACD agent may be a member of up to three different splits. Each ACD agent continuously exhibits a work state for every split of which it is a member. Valid work states are Avail, Unstaffed, AUX-Work, ACW, ACD (answering an ACD call), ExtIn, ExtOut, and OtherSpl. An agent's work state for a particular split may change for a variety of reasons (example: when a call is answered or abandoned, or the agent changes work modes). The BCMS feature monitors work states and uses this information to provide BCMS reports.

write operation
The process of putting information onto a storage medium, such as a hard disk.

WSA
Waiting session accept

WSS
Wireless Subscriber System

Z

ZCS
Zero Code Suppression

Index

Symbols

sign, 4-17, A-49
  dialed ahead digits, 4-17
  with digits, A-37
* symbol
  dial-ahead digits, A-36
  dialed ahead digits, 4-17
  with digits, A-37

A

abbreviated dialing special characters
  route-to, A-85, A-88
ACD agent login ID
  form, 11-21
active VDN, 2-11
adapting
  to a long wait, 1-15
  to changing call traffic, 1-15
adjunct routing
  considerations, C-3
  function, 10-1
  hardware and software requirements, B-7
  relationship table for treatment and goto steps, 10-5
  with call prompting, 10-1
  with look-ahead interflow, 10-1
adjunct routing command, xxiv, 2-16, A-17
  cancelling ASAI route request, A-20
  neutral vector command, 7-5, A-21
  success/failure criteria, A-100
  syntax, A-17
  troubleshooting, E-4
administering
  VDN skills, 11-8
advanced vector routing, xxiii, 5-1
  expected wait time, 5-2
  hardware and software requirements, B-4
  rolling average speed of answer, 5-10
  VDN calls, 5-12
agent login ID
  associated capabilities, 11-23
  form, 11-21
agents
  available
    definition, 2-5
  direct, xxii
  logical, xxii, 11-2
  matching talents with needs of caller, 11-1
  optimal utilization, 7-1
  staffed
    definition, 2-5

agents (continued)
when available, 1-7
when not available, 1-7
ANI routing, 6-2
calling party number, 6-2
function, 6-2
use in North America, 6-2
ANI/ii-digits
hardware and software requirements, B-5, B-6
ANI/ii-digits routing, xxiv
ANI routing, 6-2
requirements, 1-4
announcement command, 2-16
classifications of, 3-3
differences between G2 and R5, F-6
example, 3-3, 3-4, 3-9
neutral vector command, 7-5, A-25
success/failure criteria, A-100
syntax, A-24
troubleshooting, E-5
announcements, A-24
example, 3-4, 3-5
answer supervision considerations
adjunct routing, A-21
announcement, A-25
busy, A-26
check-backup, A-31
collect digits, A-38
converse-on, A-50
disconnect, A-58
goto step, A-45, A-66
goto vector, A-73
messaging, A-75
queue-to, A-80
route-to, A-84, A-88
stop, A-94
wait-time, A-98
answering agent's display, 7-17
application
example
adjunct routing, 12-10, 12-19
ANI routing, 12-11
automated attendant, 12-4
basic call vectoring, 12-2, 12-5, 12-8, 12-10, 12-11, 12-15, 12-19
call prompting, 12-4, 12-5, 12-10, 12-11, 12-19
customer service center, 12-2
data in/voice answer, 12-5
data/message collection, 12-5
distributed call centers, 12-8
DIVA and data/message collection, 12-5
expected wait-time, 12-11
expert agent selection, 12-15, 12-19
help desk, 12-10
insurance agency/service agency, 12-11
look-ahead interflow, 12-8
resort reservation service, 12-19
rolling ASA, 12-11
VDN calls, 12-11

application
example (continued)
warranty service, 12-15
warranty service call center, 12-17, 12-18
ASA, 5-10
definition, G-13
ASAI
link failure, A-18
ASAI message
contents of, 10-2
assigning call answering tasks to splits, 2-7
asterisk (*)
*, use of, A-35
automatic number identification, 6-2
calling party number, 6-2
use in North America, 6-2
automating tasks, 1-16
availability of agents, 2-5
average speed of answer, 5-10
definition, G-13
awaiting the response to the call route request, 10-6

B

basic call vectoring, xxiii
considerations, C-1
hardware and software requirements, B-2
basic components of call vectoring, xx
BCMS, A-27
description of, G-1
function, G-1
interactions with
adjunct routing, A-23
busy, A-27
check-backup, A-33
converse-on, A-57
disconnect, A-59
messaging, A-77
queue-to, A-82
route-to, A-93
reports, G-13
BCMS Split Report, G-14
for security use, J-2
VDN Real-Time Report, G-14
VDN Summary Report, G-14
standards, G-3
for interpreting split flows, G-5
for interpreting VDN flows, G-3
benefits of call vectoring, 1-15
Best Service Routing (BSR)
benefits, 8-2
BSR Available Agent Strategies, 8-13
commands
check, A-28
consider, A-39
goto step, A-60

Best Service Routing (BSR)
commands (continued)
goto vector, A-67
queue-to, A-78
reply-best, A-83
determining the best resource, 8-12
hardware and software requirements, B-5
multi-site
administration procedures, 8-49 to 8-51
Application Plan form, 8-28
application plans, 8-28
applications, 8-26
commands used, 8-25
examples
with 2 switches, 8-29
with 4 switches, 8-36
4 switches, limited trunks, 8-36
forms required, 8-23
planning, 8-47 to 8-49
overview, 8-1
requirements
for networks, 8-6
for switches, 8-5
single-site, 8-10
administration procedures, 8-46 to 8-47
commands used, 8-11
examples
basic, 8-14
user adjustments, 8-17
forms required, 8-11
planning, 8-46
vectors
tips for writing, 8-52
better utilization of agents, 2-4
blocking new incoming calls, L-8
branching, 1-10, 2-16
branching and programming, 2-15, 3-2
busy, 2-16, A-26
difference between G2 and R5, F-7
busy command
success/failure criteria, A-100
syntax, A-26
troubleshooting, E-5

C

call center setup
EAS, K-11
agent skills worksheet, K-22
assigning names to skills, K-12
current split operation worksheet, K-19
customer needs worksheet, K-20
dial plan administration, K-15

call center setup
EAS (continued)
individual agent skill worksheet, K-21
objectives, K-11
objectives worksheet, K-18
skill status assignments, K-15
steps, K-11
VDN skill preferences worksheet, K-23
vector design worksheet, K-24
key factors, K-1
non-EAS
current split operation worksheet, K-8
customer needs worksheet, K-9
guidelines, K-3
objectives worksheet, K-7
steps, K-2
vector design worksheet, K-10
call flow method, 2-2
adjunct routing, 2-3
interflow, 2-2
intraflow, 2-2
look-ahead interflow, 2-3
multiple split queuing, 2-2
call flows
answered and abandoned calls, G-2
busies and disconnects, G-3
classes of, G-2
converse-VRI calls, I-1
defining and interpreting, G-2
split inflows, outflows, and dequeues, G-4
types that are tracked, G-2
VDN inflows and outflows, G-3
vector inflows and outflows, G-4
call handling
optimal, 7-1
call not queued at stop step, E-19
call prompting
call set, 4-2
capabilities, xxiii
command categories, 4-2, 6-2
considerations, C-2
digit entry, 4-3
entering variable length digit strings, 4-4
functions, 4-5
creating service observing vectors, 4-12
passing digits to an adjunct, 4-12
treating digits as a destination, 4-6
using digits on the agent's set, 4-10
using digits to collect branching information, 4-7
using digits to select options, 4-10
hardware and software requirements, B-3
purpose, xxiii, 4-1
removing incorrect digits, 4-3
variable length digit string, 4-3
with VRI, 4-1
call route request, 10-2, 10-6, 10-7

call treatment
customizing, 1-16
personalization, 1-16
call vectoring
benefits, 1-15
definition, xix
difference between G2 and R5, F-8
features, xxiii
adjunct routing, xxiv
advanced vector routing, xxiii
ANI/ii-digits, xxiv
basic call vectoring, xxiii
call prompting, xxiii
look-ahead interflow, xxiv
principles, xxv
removing incorrect digits, 4-4, 4-16
upgrading to, B-8
call vectoring command
neutral vector command, A-76
call-back provisions
diagram of, 12-22
caller entered digits, 6-8, A-34
Caller Information Forwarding (CINFO)
answer supervision, A-38
example, 6-10
interactions, 6-10
with collect digits command, A-34, A-37
caller needs
example table matching skills and needs, 11-3
calling
a direct agent, 2-5
during non-business hours, 1-13
CALLR-INFO button
format of display, 4-11
cancelling ASAI route request, A-20
cdpd, see customer database provided digits
ced, see caller entered digits
chaining of vector steps, 2-2
changing vectors, 1-3, B-9
check-backup, 2-16
check-backup command, 1-11, A-28
example, 3-14
neutral vector command, 7-5, A-31
success/failure criteria, A-101
syntax, A-28
troubleshooting, E-5, E-6
checking
availability of split, 1-12
queue capacity, 1-12
CINFO, see Caller Information Forwarding
CMS
description of, G-1
function, G-1
interactions
with adjunct routing, A-22
with busy, A-27
with check digits, A-38
with check-backup, A-31
with converse-on, A-57

CMS
interactions (continued)
with disconnect, A-59
with goto step, A-66
with goto vector, A-45, A-73
with messaging, A-76
with queue-to, A-80
with route-to, A-91
with stop, A-95
with wait-time, A-99
reports, G-13
for security use, J-2
Split Summary Report, G-13
VDN Report, G-13
Vector Report, G-13
standards, G-3
for interpreting split flows, G-5
for interpreting VDN flows, G-3
using in expert agent selection environment, G-14
collect digits, 2-3, A-34
collect digits command, 2-16, 4-2
entering an extension, xxiii
success/failure criteria, A-101
syntax, A-34
troubleshooting, E-7
collecting and acting on information, 2-15
collecting caller information, 1-16
command category
for advanced vector routing, 5-1
for ANI/ii-digits, 6-1
for basic call vectoring, 3-1
for call prompting, 4-2, 6-2
command table
for advanced vector routing, 5-1
for ANI/ii-digits, 6-1
for basic call vectoring, 3-1
for call prompting, 4-2
comparison operators, 2-18
connecting to voice mail, 1-16
consider command, A-39
multi-site examples, 8-36, 8-41
single-site examples, 8-14, 8-17
considerations
adjunct routing, C-3
basic call vectoring, C-1
call prompting, C-2
look-ahead interflow, 7-20
VDN return destination, C-4
control flow
type
conditional branching, 2-14
sequential flow, 2-14
unconditional branching, 2-14
controlling call processing, xxiii

converse VRI calls
call flow phase
data passing, I-3
data return, I-7
script completion, I-10
script execution, I-7
converse-on command, 2-16, A-46
function, A-47
neutral vector command, 7-5
success/failure criteria, A-103
syntax, A-39, A-46
troubleshooting, E-9
converse-VRI calls
call flow phase
VRU data collection, I-6
creating
a new vector, 1-3
service observing vectors, 4-6
customer database provided digits, 6-8, A-34
customizing call treatment, 1-16, 2-5

D

defining desired service, 2-8
deleting
vector step, 1-5, 1-6
delivery of queued calls, 1-7
denying access, 2-9
dequeued average queue time
definition, G-13
dial-ahead digits
ASAI provided, 4-19
digits, 4-3
ASAI provided dial-ahead digits, 4-19
collect digits
maximum number, A-34
collect digits command
maximum number, A-17
collected prior to timeout, A-35
dial-ahead digits with *, A-36
entering, 4-3
dial-ahead digits, 4-3, 4-5
variable-length digit strings, 4-4
including # sign, A-36
maximum number, A-36
removing
incorrect digit strings, 4-3, 4-4
returned by VRU, A-34
Touch-Tone, A-35
with # sign, A-37
with * symbol, A-37
direct agent, xxii
direct agent call
definition, 2-5
directing calls to a specific agent, 11-2

disconnect command, 2-16, A-58
example, 3-10
success/failure criteria, A-103
syntax, A-58
troubleshooting, E-10
displaying digits on the agent's set, 4-6
during peak
calling periods, 2-3
heavy traffic, 1-11

E

EAS
definition, xxii
function, 11-1
editing, 1-5
enabling the vector disconnect timer, B-8
encouraging caller to remain on-line, 1-9
entering
a command
in abbreviated form, 1-5
a vector, 1-1
dial-ahead digits, 4-5
digits, 4-3
use of #, 4-4
variable-length digit strings, 4-3, 4-4
vector steps, 1-3
evaluating
calls prior to processing, xxiv
effectiveness of vector programming, G-2
performance, G-2
split performance, G-11
event type
adjunct route failed, E-22
events, E-16, E-19
example application
remote access with host provided security, C-5
saving in trunk facilities between call centers, C-7
split flow tracking, G-6
VDN override, 2-11
warranty service call center, 12-17, 12-18
example vector
accessing voice response scripts, 3-12
accommodate a super agent pool, 11-17
adjunct routing vector, 10-2
with redundancy, 10-11
ANI routing example, 6-3
automated attendant application, 12-4
call interflow, 3-18
claims application, 12-13
conditional branching, 3-21
customer service application, 12-14
customer service center application, 12-3
delay announcement, 3-3
delay with audible feedback, 3-5
delay with multiple audio/music source feedback, 3-6
dial-ahead digits, 4-15, 4-16

example vector (continued)
disconnecting a call, 3-10
distributed call centers application, 12-9
DIVA and data/message collection application, 12-6
emergency and routine service application, 12-23, 12-24
expected wait time
for a call, 5-3
for a split, 5-2
routing and passing VRU wait, 5-6
expected wait time routing
routing to the best split, 5-8
field agent vector application, 12-12
forced announcement, 3-4
help desk application, 12-10
ii-digits routing example, 6-8
information announcement, 3-4
late caller application, 12-26
leaving recorded messages, 3-15, 3-16
messaging options application, 12-28
multiple split queuing, 3-14
notifying callers of wait-time without a VRU, 5-7
passing digits to an adjunct, 4-12
providing busy tone, 3-9
receiving switch inflow vector, 7-8
remote access service observing vector, 4-13
return destination vector
with announcement, C-8
with remote access, C-6
rolling ASA routing, 5-12
sending switch outflow vector, 7-7
service agency clients application, 12-14
service observing vector, 4-13, 4-14
skipping/non skipping of treatment command with
ASAI link down, 10-4
stopping vector processing, 3-22
supplementary delay announcement, 3-4
tandem switch vector, 7-16
testing
for ANI in vector routing table, 6-5
for digit, 4-9
for digits in vector routing table, 4-9
for digits not in vector routing table, 4-9
treating digits as a destination, 4-6
treatment step used as a delay for adjunct routing, 10-7
unconditional branching, 3-20
using digits to collect branching information, 4-7
using digits to select options, 4-10
VDN calls routing, 5-13
vector for service observing, 3-19
example vector routing table
for ANI routing, 6-4
for call prompting, 4-8
example vector step
adjunct routing, A-17
announcement, A-24
check-backup, A-29
collect digits, A-34
converse-on, A-47
disconnect, A-58

example vector step (continued)
goto step, A-64
goto vector, A-71
messaging, A-74
queue-to, A-78
route-to, A-86
wait-time, A-97
executing VRU scripts, 2-15
expected wait time, 5-2
algorithm, 5-4
factors causing for split priority level to decrease, 5-9
factors effecting the value, 5-9
for a call, 5-3
passing to a VRU, 5-3
when infinite, 5-3
expert agent selection
adjunct and feature interactions, 11-24
adjunct interactions, 11-30
conversion
administration for, L-2
blocking of new incoming calls, L-8
considerations prior to, L-1
steps, L-1
steps for cutover, L-8
definition, xxii
feature interactions, 11-24
function, 11-1
requirements, 11-1
requires ACD, 11-1
requires call vectoring, 11-1
splits, 11-1
tracking
agents and their skills, G-15
direct agent calls, G-15
for VDN skill preferences, G-15
non-ACD calls, G-15
upgrading to, L-1
upgrading to R5, 11-38
using CMS, G-14
extended wait time
tracing, E-33

F

feature interactions
with adjunct routing, A-21
with announcement, A-25
with busy, A-26
with check digits, A-38
with check-backup, A-31
with converse-on, A-50
with disconnect, A-59
with goto step, A-45, A-66
with goto vector, A-73
with messaging, A-76
with queue-to, A-80
with route-to, A-88

feature interactions (continued)
with stop, A-94
with wait-time, A-98
features of call vectoring, xxiii
adjunct routing command, xxiv
advanced vector routing, xxiii
ANI/ii-digits, xxiv
basic call vectoring, xxiii
call prompting, xxiii
look-ahead interflow, xxiv
functions
of basic call vectoring, 3-2
of call prompting, 4-5

G

goto command
differences between G2 and R5, F-4
example, 3-20, 3-21
neutral vector command, 7-5
success/failure criteria, A-103
troubleshooting, E-10
goto step command, 2-16, A-60
neutral vector command, A-66
goto vector command, 2-16, A-67
neutral vector command, A-73

H

handling multiple calls, 2-6

I

identifying caller needs
call prompting/VRU digits, 11-4
direct agent calling, 11-5
DNIS/ISDN called party, 11-4
example prompts, 11-5
host database lookup, 11-5
methods of, 11-3
table of services and DNIS digits, 11-4
ii-digits
table of those currently available, 6-6
values associated with them, 6-6
ii-digits routing, 6-5
uses for, 6-5
improving
performance, 2-1, N-1
expert agent selection, 11-1
service, 2-4, 2-7
the average speed of answer, 1-16

information forwarding
determining user information needs, 9-7
enhanced information forwarding, 9-4
backward compatibility, 9-6
benefits of, 9-2
call-related information, 9-5
collected digits, 9-5
function, 9-1
global support, 9-6
in-VDN time, 9-5
function, 9-1
network requirements, 9-3
troubleshooting, 9-10
inserting vector steps, 1-5
international
equivalent circuit packs, B-2, B-3

L

LAI
function, 7-1
latest VDN, 2-11
leaving a message, xxiii, 1-14, 3-15
listing existing vectors, 1-3
load balancing, xxiv
optimal, 7-1
providing, 1-16
logical agent, xxii
look-ahead interflow, xxiv
achieving FIFO, 7-9
ADR, 7-18
alternate destination redirection, 7-18
diagram of example, 7-19
considerations, 7-20
diagram of tandem switch configuration, 7-15
diagram of two switch configuration, 7-3
DNIS and VDN override, 7-17
DNIS information, 7-17
enhanced, 7-9
function, 7-1
hardware and software requirements, B-6
interflow eligibility, 7-11
multisite applications, 7-20
route-to command, 7-6
setting the minimum EWT, 7-12
tandem switch configuration
far end switch operation, 7-16
sending switch operation, 7-15
tandem switch operation, 7-15
troubleshooting, 7-21, E-3
two switch configuration
receiving switch operation, 7-7

M

maximizing performance, N-1, N-3, N-4, N-5, N-8, N-9
  example vector, N-3, N-5, N-6, N-8, N-9
messaging, 2-17, A-74
  ASAI
    contents of, 10-2
  example, 3-16
  leaving a message, 1-14
messaging command
  example, 3-15
  neutral vector command, 7-5
  success/failure criteria, A-104
  syntax, A-74
  troubleshooting, E-11
multiple call handling, 2-6

N

naming
  a vector, 1-3
neutral vector command, 7-5
non-business hours
  call during, 1-13
numbering
  of vector steps, 1-6

O

observing VDNs, 2-13
off-loading calls, xxiv
option
  VDN override, 2-11
originator's display, 7-18

P

passing digits
  to an adjunct, 4-6
  to PBX, 2-10
performance
  basic principles for improving, N-1
  effects of ASAI link failure, 10-3
  evaluating, G-2
    effectiveness of vector programming, G-2
    for split, G-11
  improving, N-3, N-4, N-5, N-8, N-9
    example vector, N-3, N-5, N-6, N-8, N-9
  looping, N-1

performance (continued)
maximizing, N-1, N-4, N-5, N-8, N-9, N-3
processing cost
comparisons, N-3, N-5, N-7
of vector steps, N-10, N-11
testing vectors, N-2
personalizing call treatment, 1-16
placing a call in queue, xxiii
preventing unauthorized users access, J-1
prioritizing calls, 1-7, 1-11, 1-16, 2-4
process
involving general number dialing
diagram of, 12-21
involving specific number dialing
diagram of, 12-20
processing calls
faster, 1-15
functions, 3-2
intelligently, 1-15
programming call processing, xxiii
prompting a caller, 2-11
properties, 2-8
providing
an announcement, xxiii
call treatments, 2-15, 3-2
caller feedback, 1-15
choices to callers, 1-15
faster service, 1-16
feedback, 1-8, 1-9, 1-10
initial feedback to caller, 2-6
load balancing, 1-16
night service, 1-15
security, 2-9

Q

queue-to command, 2-17, A-78
queue-to main
differences between G2 and R5, F-3
neutral vector command, A-80
queue-to main command
neutral vector command, 7-5, A-80
success/failure criteria, A-104
syntax, A-78
troubleshooting, E-6
queuing calls
methods for, 2-2
to split, 2-4
maximum number of, 2-4
without call vectoring, 2-4

R

receiving and implementing the call route, 10-7
receiving feedback about a call, 1-8
redirecting calls
  methods for, 2-2
reducing
  caller hold time, 1-16
  number of needed agents, 2-7
  staffing requirements, 1-16
  transferred calls, 1-16, 2-3
removing incorrect digits strings, 4-4
reply-best command, A-83
reporting
  agent handling, 2-8
  call handling, 2-8
  via Basic Call Management System, 2-8
  via BCMS, 2-8
  via CentreVu Call Management System, 2-8
  via CMS, 2-8
reports
  BCMS
    BCMS Split Report, G-14
    VDN Real-Time Report, G-14
    VDN Summary Report, G-14
  CMS
    Split Summary Report, G-13
    VDN Report, G-13
    Vector Report, G-13
requeuing calls, 2-5
requirements
  software and hardware
    for adjunct routing, B-7
    for advanced vector routing, B-4
    for ANI/ii-digits routing, B-5, B-6
    for basic call vectoring, B-2
    for Best Service Routing, B-5
    for call prompting, B-3
    for look-ahead interflow, B-6
rolling ASA
  considerations, 5-11
  split calculation, 5-10
  VDN calculation, 5-11
rolling average speed of answer, 5-10
route validation, 10-7, 10-8, A-20
route validation failure, 10-8, A-20
route-to
  look-ahead interflow, 7-6
route-to command, A-85
  differences between G2 and R5, F-5
  neutral vector command, 7-5, A-90
  summary of conditions for destination types, H-1
  syntax, A-85
  troubleshooting, E-12
route-to digits, 2-17
route-to number, 2-17

route-to requests
multiple outstanding, 10-11
routing, 6-2
ii-digits, 6-5
uses for, 6-5
routing calls, xxiii, xxiv, 1-16, 2-3, 2-15, 3-2
based on DNIS, 2-7
example table of call distribution via UCD/EAD, 11-20
example table of UCD/EAD call scenario, 11-20
intelligently, 7-1
overriding specifications, 2-11
to an agent, 11-18
delivery from a skill hunt group, 11-18
to skill queue
using call prompting, 11-14
using expert agent selection, 11-18
using super agent pool, 11-15
routing tables, 6-4

S

security
main type of problem, J-1
method
front-ending remote access, J-2
advantages, J-2
replacing remote access, J-2
methods for preventing remote access abuse, J-1
preventing unauthorized users access, J-1
providing, 2-9
replacing remote access, J-2
with EAS, J-3
with expert agent selection, J-3
with remote access, J-1
with service observing, J-4
with vector initiated service observing, J-4
sending the route call request, 10-2
service observing, 2-13, 4-13
silence, 2-15
when occurs, 2-6, 2-7, 2-17
skill
definition, 11-5
example table for an auto club, 11-6
table for auto club application, 11-16
table of preferences assignments for VDN 1616, 11-16
skill call
example table of distribution for a single agent, 11-19
skill call queue sequence
example table, 11-19
split
backup
definition, 2-4
main
definition, 2-4
split flows
differences among G1/G2/G3, F-11

staffed agent
for ACD split, 2-5
staffed agents
basis of call management decisions, 2-5
check backup command, 2-5
conditional branching, 2-14
definition of, 2-5
for non-ACD hunt groups, 2-5
goto command, 2-5
number of, 2-16
status lamp, 4-11
CALLR-INFO button, 4-11
NORMAL button, 4-11
steps
maximum number of, 2-14
stop command, 2-17
example, 3-22
neutral vector command, 7-5, A-94
success/failure criteria, A-105
syntax, A-94
troubleshooting, E-12

T

tandem switch
far end operation, 7-16
far end switch operation, 7-15
sending switch operation, 7-15
testing call treatment, 1-15
testing vectors, B-9
tracing
call flow, E-32
extended wait time, E-33
vdn, E-32
vector, E-32
tracking
agents and their skills, G-15
calls, G-2
direct agent calls, G-15
example
split flow, G-6
for abandoned calls, G-8
for call answered
after route to split, G-11
by a primary split, G-6
by non-primary split, G-7
after route to VDN, G-10
by primary split
after route to VDN, G-9
for non-ACD calls, G-15
VDN skill preferences, G-15
transfer call management control
caller-selected routing, 2-3
messaging, 2-3
treating digits as a destination, 4-5, 4-6

troubleshooting
1,000 step executed, E-19
AAS split cannot queue, E-30
adjunct
link error, E-23
route cancelled, E-22
route failed, E-22
administration change, E-19
agent
drops converse, E-24
not logged in, E-23
not member of split, E-23
receiving phantom call, E-3
all look-ahead interflow attempts accepted, E-3
all trunks busy on a quiet system, E-4, E-12
alternate audio/music source not heard, E-13
ANI digits not passed, E-9
ANI not avail - digits, E-25
ANI not avail - table, E-25
announcement not heard, E-5, E-10
while waiting for digits, E-7
ASA - invalid VDN, E-25
ASA - no staffed agents, E-31
ASAI transfer converse, E-24
audible feedback
lasts longer than the delay interval, E-2
longer than delay interval, E-13
shorter than delay interval, E-13
AUDIX link down, E-30
branch is not made
to the specified step, E-10
to the specified vector, E-10
busy step for CO trunk, E-23
busy tone, E-11
call apparently answered in wrong order, E-6
call cannot be queued, E-20
call does not enter queue or terminate to agent, E-5, E-6
call dropped, E-12, E-19
call dropped by vector disconnect timer, E-19
call stuck in converse, E-14
caller information button denied, E-8
Can’t connect idle agent, E-30
collect
announcement
not heard, E-16
not heard and first collected digit incorrect, E-8
collect step and announcement skipped, E-7
converse
drop during data, E-24
no ANI digits, E-24
no prompt digits, E-24
no qpos digits, E-24
step skipped, E-14
transfer denied, E-24
coverage conference denied, E-25

troubleshooting (continued)
data return
no digits, E-24
timeout, E-24
delay before AUDIX answers, E-11
delay before hearing announcement, E-8
dial-ahead digits not recognized, E-8
dial-ahead discarded, E-20
digits incomplete, E-15
double coverage attempt, E-22
expected wait-time
call no working agents, E-31
call not queued, E-26
no split queue, E-31
not sent to VRU, E-26
split locked, E-31
split no working agents, E-31
split queue full, E-31
expected wait-time no history for split, E-31
extra delay, E-10
before hearing announcement, E-5
first set of digits not collected, E-14
ii-digits not avail - digits, E-26
ii-digits not avail - table, E-26
incomplete announcement, E-5, E-8
insufficient digits collected
call routed to intercept, E-8
invalid
destination, E-22
direct agent, E-23
EAS hunt group used in the vector step, E-25
look-ahead
DNIS name not displayed, E-3
interflow retry, E-22
messages not found, E-11
messaging step failed, E-21
music not heard, E-13
network reorder, E-12
no announcement available, E-20
no available trunks, E-22
no data returned from VRU, E-9
no digits
collected, E-23
to route-to, E-22
no entries in routing table, E-25
no look-ahead interflow attempts accepted, E-3
no Touch-Tone Receiver available, E-20
no vector steps, ANI sent, E-25
not a messaging split, E-30
not all digits returned to the DEFINITY switch, E-16
not vector-controlled, E-30
prompting buffer overflow, E-21
qpos digits not passed, E-9
queue before route, E-22
queued to three splits, E-20
redirect
of call failed, E-24
unanswered call, E-23

troubleshooting (continued)
retrying announcement, E-19
ringback heard instead of busy tone, E-5
route -to step failed, E-21
route-to step failed, E-22
routing table not assigned, E-25
second set of digits
is the same as the first digits passed, E-15
not collected, E-14
skill indirection used improperly, E-25
split queue is full, E-30
step skipped, E-4, E-11
no message left, E-11
that is, default treatment, E-12
steps
display event report, E-18
display events form, E-17
system clock change, E-26
time not set, E-23
unexpected
busy tone, E-4
intercept or reorder tone heard, E-4
network reorder or intercept, E-4
silence after announcement, E-5
step skipped (that is, default treatment), E-4
unexpected intercept or reorder tone heard, E-12
vector processing halted at collect step, announcement heard again upon return, E-8
vector processing stops, E-5
vector stuck, E-2, E-8, E-10
with busy, E-11
with ringback, E-11
vector with no steps, E-19
VRU script
not executed, E-9
terminated prematurely, E-9
wait digits not passed, E-9
wait step
music failed, E-23
ringback failed, E-23

U

upgrading
a call center to expert agent selection, L-1
to a call vectoring environment, B-8
using digits
to collect branching information, 4-5
to select options, 4-5

V

valid entries
for check-backup, A-28
for collect digits, A-34
for converse-on, A-39, A-46
for disconnect, A-58
for goto step, A-62
for goto vector, A-69
for messaging, A-74
for queue-to, A-78
for route-to, A-85
for wait-time, A-96
VDN, 2-8
active, 2-11
calls, 5-12
calls counts
which calls included, 5-13
definition, xxi, 2-1, 2-8
in coverage path
application uses, 2-12
latest, 2-11
multiple, xxi
observing, 2-13
override
example application, 2-11
properties
1st/2nd/3rd skill, 2-10
acceptable service level, 2-10
allow VDN override, 2-9
AUDIX name, 2-10
class of restriction (COR), 2-9
extension, 2-9
measured, 2-10
messaging server name, 2-10
name, 2-9
return destination, 2-10
tenant partition number (TN), 2-9
VDN of origin announcement extension, 2-10
vector number, 2-9
return destination
considerations, C-4
skills
administering, 11-8
vdn
tracing, E-32

vector
changing existing, 1-3, B-9
creating a new, 1-3
definition, xxi, 2-8
disconnect timer, B-8
editing, 1-5
entering, 1-1
events, E-16, E-19
example, 3-15
accessing voice response scripts, 3-12
accommodate a super agent pool, 11-17
adjunct routing vector, 10-2
adjunct routing vector with redundancy, 10-11
automated attendant application, 12-4
call interflow, 3-18
claims application, 12-13
conditional branching, 3-21
customer service application, 12-14
customer service center application, 12-3
delay announcement, 3-3
delay with audible feedback, 3-5
delay with multiple audio/music source feedback, 3-6
dial-ahead digits, 4-15, 4-16
disconnecting a call, 3-10
distributed call centers application, 12-9
DIVA and data/message collection application, 12-6
emergency and routine service application, 12-23, 12-24
expected wait time
for a call, 5-3
for a split, 5-2
routing - routing to the best split, 5-8
routing and passing VRU wait, 5-6
field agent vector application, 12-12
forced announcement, 3-4
help desk application, 12-10
ii-digits routing example, 6-8
information announcement, 3-4
late caller application, 12-26
leaving recorded message, 3-16
leaving recorded messages, 3-16
messaging options application, 12-28
multiple split queueing, 3-14
notifying callers of wait-time without a VRU, 5-7
passing digits to an adjunct, 4-12
providing busy tone, 3-9
receiving switch inflow vector, 7-8
remote access service observing vector, 4-13
return destination vector
with announcement, C-8
with remote access, C-6
rolling ASA routing, 5-12
sending switch outflow vector, 7-7
service agency clients application, 12-14

vector
example (continued)
service observing vector, 4-13, 4-14
skipping/non-skipping of treatment commands
with ASAI link down, 10-4
stopping vector processing, 3-22
supplementary delay announcement, 3-4
tandem switch vector, 7-16
testing
for ANI in vector routing table, 6-5
for digits in vector routing table, 4-9
treating digits as a destination, 4-6
unconditional branching, 3-20
using digits
to collect branching information, 4-7
to select options, 4-10
VDN calls routing, 5-13
vector for service observing, 3-19
listing existing, 1-3
naming, 1-3
testing, B-9
tracing, E-32
vector applications
table of examples, 12-1
vector chaining
goto command, 3-22
multiple, xxii
multiple vectors, 3-22
purpose, 3-22
route-to, 3-22
vector command
adjunct routing command, 2-16, A-17
advanced vector routing, 5-1
command table, 5-1
ANI/ii-digits, 6-1
command table, 6-1
announcement command, A-24
announcements, 2-16
available with
call prompting, A-3
call vectoring, A-3
basic call vectoring, 3-1
command table, 3-1
busy, 2-16, A-26
call acceptance
qualification of commands, 7-3
call denial
qualification of commands, 7-4
call prompting, 4-2, 6-2
command table, 4-2
check-backup, 2-16, A-28
collect digits, 2-16, A-34
comparison operators, 2-18
condition testing, 2-17
consider, A-39
converse-on, A-46

vector command (continued)
converse-on command, 2-16
deleting, 1-5
disconnect, 2-16
disconnect command, A-58
function of each, A-1
goto step, 2-16
goto step command, A-60
goto vector, 2-16, A-67
maximum number, 1-5
messaging, 2-17, A-74
neutral
qualification of commands, 7-5
OCM predictive calls, 11-29
parameters, A-6
queue-to, 2-17
queue-to command, A-78
reply-best, A-83
route-to, A-85
route-to digits, 2-17
route-to number, 2-17
stop, 2-17
success/failure criteria, A-100
syntax, A-6
wait-time, 2-17, A-96
vector directory number
definition, 2-1, 2-8
properties, 2-8
vector routing tables, 6-4
vector event
advantages of tracking unexpected, E-16
displaying, E-16, E-17
logging of, E-17, E-18
range of type, E-18
report, E-18
tracking, E-16
unexpected, E-16
unique number, E-18
with debugging, E-14
vector processing
ASAI link failure, A-18
BDMS Report
description, G-14
branching, 2-14, 2-15, 2-16
collecting from caller, 2-17
control flow, 2-1, 2-8
types of, 2-14
factors, 2-1
failure
converse-on step, A-53
resulting in these destinations, A-89
maximum number of steps, 2-14
programming
collecting and acting on information, 2-15
collecting from caller, 2-15
providing treatments, 2-15
routing calls, 2-15
programming capabilities
branching, 2-14

vector processing (continued)
Split Summary Report
description, G-13
stopping, 2-1, 2-13, 2-14, 2-15, 3-1, 3-2, 3-5, 3-21, 3-22
terminating, 3-16, 3-17, 3-18, 3-20
termination, 2-16
termination vs stopping, 2-14
troubleshooting, E-2
VDN Real-Time Report
description, G-14
VDN Report
description, G-13
VDN Summary Report
description, G-14
Vector Report
description, G-13
with coverage, 2-12, 4-7
vector routing table, 4-8, 6-4
vector step
chaining, 2-2
conditional branching, 2-14
deleting, 1-6
entering, 1-3
example
adjunct routing, A-17
announcement, A-24
check-backup, A-29
collect digits, A-34
converse-on, A-47
disconnect, A-58
goto step, A-64
goto vector, A-71
messaging, A-74
queue-to, A-78
route-to, A-86
treatment step
used as a delay for adjunct routing, 10-7
wait-time, A-97
inserting, 1-5
maximum number, 2-2
numbering, 1-6
sequential flow, 2-14
stopping, 2-14
terminating, 2-14
termination vs stopping, 2-14
unconditional branching, 2-14
vector-controlled split, 3-15, 3-16
voice response script, 3-1, 3-10
accessing, 3-11
checking amount of time for execution, 3-13
execution of, 3-11
interruption of, 3-12
VRI
advantage of, 3-11
capabilities, 3-10
description, 3-10

VRU, 3-10
activating a voice response script, A-46
advantages of, 3-11
executing a script, 3-1
execution of VRU script, 3-10
normal override rules, A-56
offloading recorded announcements to, A-54
outpulsing data, A-48, A-50, A-55
outpulsing to extension, 3-12
passing data between VRU and DEFINITY switch, 3-10
passing EWT to, 3-11
returning data to the switch, A-47
service observing pending mode, A-55
storing received data, A-47
tandemed to ASAI host, 3-10
used as an external announcement, 3-10
using digits returned from, A-34
VRU digits
conditional branching, A-48
displayed via CALLR-INFO button, A-48
extension in a route-to command, A-48
tandemed to an ASAI host, A-48

W

wait command
adjunct routing command, 10-6
wait-time, 2-17, A-96
predictions
circumstances that will limit, 5-5
when to use predictions, 5-4
wait-time command
differences between G2 and R5, F-7
example, 3-6
neutral vector command, 7-5, A-98
success/failure criteria, A-105
syntax, A-96
troubleshooting, E-13
wildcards, 4-7, 4-8
work mode
after-call-work mode, 2-5
auto-in work mode, 2-5
auxiliary-work mode, 2-5
manual-in work mode, 2-6