

Critical Release Notice

Publication number: 297-2621-372
Publication release: Standard 04.03

The content of this customer NTP supports the
SN06 (DMS) software release.

Bookmarks used in this NTP highlight the changes between the baseline NTP and the current release. The bookmarks provided are color-coded to identify release-specific content changes. NTP volumes that do not contain bookmarks indicate that the baseline NTP remains unchanged and is valid for the current release.

Bookmark Color Legend

Black: Applies to new or modified content for the baseline NTP that is valid through the current release.

Red: Applies to new or modified content for NA017 that is valid through the current release.

Blue: Applies to new or modified content for NA018 (SN05 DMS) that is valid through the current release.

Green: Applies to new or modified content for SN06 (DMS) that is valid through the current release.

Attention!

Adobe® Acrobat® Reader™ 5.0 is required to view bookmarks in color.

Publication History

March 2004

Standard release 04.03 for software release SN06 (DMS).

Change of phone number from 1-800-684-2273 to 1-877-662-5669, Option 4 + 1.

297-2621-372

Digital Switching Systems

UCS DMS-250

CAIN/FlexDial Interactions

UCS11 Standard 04.02 May 1999

NORTEL
NORTHERN TELECOM

Digital Switching Systems

UCS DMS-250

CAIN/FlexDial Interactions

Publication number: 297-2621-372

Product release: UCS11

Document release: Standard 04.02

Date: May 1999

© 1997-1999 Northern Telecom

All rights reserved

Printed in the United States of America

NORTHERN TELECOM CONFIDENTIAL: The information contained in this document is the property of Northern Telecom. Except as specifically authorized in writing by Northern Telecom, the holder of this document shall keep the information contained herein confidential and shall protect same in whole or in part from disclosure and dissemination to third parties and use same for evaluation, operation, and maintenance purposes only.

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

DMS, MAP, NORTEL, NORTEL NETWORKS, NORTHERN TELECOM, NT, and SUPERNODE are trademarks of Northern Telecom.

Publication history

May, 1999

Standard release 04.02 for software release UCS11.

March, 1999

Preliminary release 04.01 for software release UCS11.

Changes to document include:

- added *O_Abandon* EDP
- added *Switch_Hook_Flash* event at *O_Mid_Call* EDP
- added software optionality control (SOC) CAIN0802, “Takeback & Transfer”
- added the *Tollfree_Service* IN/1 trigger
- modified *CallingPartyID* parameter population precedence rules

November, 1998

Standard release 03.02 for software release UCS09.

October, 1998

Preliminary release 03.01 for software release UCS09.

August, 1998

Standard release 02.02 for software release UCS08.

May, 1998

Preliminary release 02.01 for software release UCS08.

September, 1997

Standard release 01.02 for software release UCS07.

August, 1997

Preliminary release 01.01 for software release UCS07.

Contents

About this document	ix
When to use this document	ix
Intended audience	ix
How this document is organized	ix
How to check the version and issue of this document	x
References in this document	xi
What precautionary messages mean	xi
Document conventions	xii
PICs, TDPs, EDPs, triggers, and events	xii
Messaging	xii
Input prompt (>)	xiii
Commands and fixed parameters	xiii
Variables	xiii
Responses	xiii
Introduction to CAIN/FlexDial	1-1
What is NetworkBuilder?	1-2
IN and AIN	1-2
Call model	1-3
Subscription	1-4
Supported PICs	1-5
SCP interaction	1-7
What is the FlexDial Framework?	1-8
Provisioning areas	1-8
Table control	1-8
Why CAIN/FlexDial?	1-11
Digit collection and digit validation	1-12
Software optionality control	1-13
Provisioning NetworkBuilder for AXCESS originations	2-1
Provisioning	2-2

Step 1: Familiarize yourself with the call model	2-5
Step 1: Familiarize yourself with the call model PIC 1: O_Null	2-6
Step 1: Familiarize yourself with the call model PIC 3: Collect_Information	2-8
Step 1: Familiarize yourself with the call model PIC 4: Analyze_Information	2-19
Step 1: Familiarize yourself with the call model PIC 5: Select_Route	2-28
Step 1: Familiarize yourself with the call model PIC 7: Send_Call	2-30
Step 1: Familiarize yourself with the call model PIC 8: O_Alerting	2-32
Step 2: Familiarize yourself with related OA&M	2-35
Step 2: Familiarize yourself with related OA&M Treatments	2-36
Step 2: Familiarize yourself with related OA&M Billing	2-38
Step 3: Be familiar with agent setup messaging	2-41
Step 6: Datafill the AXXESS agent as CAIN-capable	2-44
Step 6: Datafill the required agents as CAIN-capable originating AXXESS agents	2-46
Step 8: Choose the type of subscription	2-47
Step 8: Choose the type of subscription Precedence Level 2 (Address)	2-65
Step 8: Choose the type of subscription Precedence Level 2 (Address)	2-72
Step 8: Choose the type of subscription Precedence Level 3 (SUBR and SUBRPARAM)	2-74
Step 8: Choose the type of subscription Precedence Level 3 (CALLTYPE)	2-76
Step 8: Choose the type of subscription Precedence Level 4 (SUBR and SUBRPARAM)	2-78
Step 8: Choose the type of subscription Precedence Level 5 (Agent)	2-80
Step 9: Define the trigger criteria	2-83
Step 10: Be familiar with digit collection	2-86
Step 10: Be familiar with digit collection Mapping OFTRREQ to FlexDial	2-93

Call processing **3-1**

Exit scenarios	3-1
Re-entry scenarios	3-3
Recollection	3-4
Dialplan index	3-5
Examples	3-6

TCAP messaging **4-1**

Outgoing message parameters **5-1**

CAIN nature of address determination	5-1
CalledPartyID	5-4
CallingPartyID	5-7
ChargeNumber	5-10

Incoming messages and parameters **6-1**

SCP responses	6-1
---------------	-----

CAIN and FlexDial tools **7-1**

CAINTEST	7-1
SCP simulator	7-1
FLEXSIM	7-1
VPTRACE	7-2

FLEXSIM 7-3

Appendix A Engineering guidelines **A-1**

General engineering rules for CAIN A-1

General engineering rules for FlexDial A-2

General engineering rules for CAIN/FlexDial Interactions A-2

Limitations and restrictions A-2

Appendix B List of terms **B-1**

Ordering information **10-1**

About this document

When to use this document

This document is meant to be used as a supplement to the *UCS DMS-250 NetworkBuilder Application Guide* and the *UCS DMS-250 FlexDial Framework Application Guide*. Only the interactions between the two applications are discussed. These interactions only affect CAIN call processing for AX-CESS originations; FlexDial call processing without CAIN interactions is unaffected.

Intended audience

This publication assists telecommunications engineers, technicians, switching system developers, operating company personnel, and anyone else who requires technical information on Carrier AIN functionality.

This document assumes the user's switch is installed, commissioned, and active.

Personnel implementing this feature require the following:

- Thorough understanding of CAIN provisioning and use
- Thorough understanding of the FlexDial Framework, including provisioning and use
- Table Editor training
- Nortel (Northern Telecom) approved datafill, translations, and maintenance training

How this document is organized

The chapters in this document provide the following:

Chapter 1, Introduction to CAIN and FlexDial

Chapter 1 provides an overview of CAIN and FlexDial.

Chapter 2, Provisioning CAIN for the AXCESS agent

Chapter 2 provides instructions for provisioning the switch for CAIN/FlexDial interaction.

Chapter 3, Call processing

Chapter 3 provides a look at how FlexDial exit and re-entry scenarios interact with CAIN call processing.

Chapter 4, TCAP messaging

Chapter 4 provides an overview of TCAP messaging.

Chapter 5, CAIN outgoing message parameters

Chapter 5 provides information on the population of query parameters.

Chapter 6, Incoming messages and parameters

Chapter 6 provides information on the response parameters.

Chapter 7, CAIN tools

Chapter 7 describes the CAINTEST tool, FLEXSIM, SCP simulator, and VPTRACE.

Appendix A, Engineering guidelines

Appendix A provides engineering guidelines you should consider when provisioning your switch.

How to check the version and issue of this document

The version and issue of the document are indicated by numbers, for example, 01.01.

The first two digits indicate the version. The version number increases each time the document is updated to support a new software release. For example, the first release of a document is 01.01. In the *next* software release cycle, the first release of the same document is 02.01.

The second two digits indicate the issue. The issue number increases each time the document is revised but rereleased in the *same* software release cycle. For example, the second release of a document in the same software release cycle is 01.02.

To determine which version of this document applies to the software in your office and how documentation for your product is organized, check the release information in *Product Documentation Directory*, 297-8991-001.

This document is written for all DMS-100 Family offices. More than one version of this document may exist. To determine whether you have the latest version of this document and how documentation for your product is organized, check the release information in *Product Documentation Directory*, 297-8991-001.

References in this document

The following documents are referred to in this document:

- *Bellcore TR-NWT-000533*
- *UCS DMS-250 Billing Records Application Guide, 297-2631-395*
- *UCS DMS-250 Commands Reference Manual, 297-2621-819*
- *UCS DMS-250 Data Schema Reference Manual, 297-2631-851*
- *UCS DMS-250 FlexDial Framework Application Guide, 297-2621-390*
- *UCS DMS-250 Local Number Portability Application Guide, 297-2621-371*
- *UCS DMS-250 Logs Reference Manual, 297-2631-840*
- *UCS DMS-250 Master Index, 297-2621-001*
- *UCS DMS-250 NetworkBuilder AIN 0.2 TCAP Protocol Definition, PLN-2621-010*
- *UCS DMS-250 NetworkBuilder Application Guide, 297-2621-370*
- *UCS DMS-250 Operational Measurements Reference Manual, 297-2631-814*
- *UCS DMS-250 Programmable Service Node Application Guide, 297-2621-380*
- *UCS DMS-250 Software Optionality Control User's Manual, 297-2621-301*

What precautionary messages mean

The types of precautionary messages used in NT documents include attention boxes and danger, warning, and caution messages.

An attention box identifies information that is necessary for the proper performance of a procedure or task or the correct interpretation of information or data. Danger, warning, and caution messages indicate possible risks.

Examples of the precautionary messages follow.

ATTENTION Information needed to perform a task

ATTENTION

If the unused DS-3 ports are not deprovisioned before a DS-1/VT Mapper is installed, the DS-1 traffic will not be carried through the DS-1/VT Mapper, even though the DS-1/VT Mapper is properly provisioned.

CAUTION Possibility of service interruption or degradation



CAUTION

Possible loss of service

Before continuing, confirm that you are removing the card from the inactive unit of the peripheral module. Subscriber service will be lost if you remove a card from the active unit.

Document conventions

This document conforms to the following conventions.

PICs, TDPs, EDPs, triggers, and events

PICs are represented in Helvetica Bold font, for example:

O_Null

TDPs and EDPs are represented in Helvetica Bold-Italic font, for example:

Origination_Attempt

Triggers and events are represented in Helvetica Italic font, for example:

Off_Hook_Immediate

Messaging

Messages are represented in Courier Bold font, for example:

Origination_Attempt

Parameters are represented in Courier Bold-Italic font, for example:

CalledPartyID

ExtensionParameters and any sub-parameters are represented in Courier font, for example:

universalAccess

Input prompt (>)

An input prompt (>) indicates that the information that follows is a command:

>CAINTEST

Commands and fixed parameters

Commands and fixed parameters that are entered at a MAP terminal are shown in uppercase letters:

>CAINTEST

Variables

Variables are shown in lowercase letters:

>SHOWFLDS fields

The letters or numbers that the variable represents must be entered. Each variable is explained in a list that follows the command string.

Responses

Responses correspond to the MAP display and are shown in a different type:

```
APPLICATION      : cain02
TIMEOUT          : 3      sec
MESSAGE          : info_analyzed
                  Cldno   : N00    2145323773
                  Clgno   : N00    2143562784
                  Chgno   : N00    2143562784
```

The following excerpt from a procedure shows the command syntax used in this document:

- 1 Show the defined fields by typing the following:

>SHOWFLDS fields

and pressing the Enter key.

where

fields is the type of field to be shown.

Sample entry: **>SHOWFLDS all**

Example of a MAP response:

```
APPLICATION      : cain02
TRANSPORT       : tcap_sccp cain_clid_gt 214
TIMEOUT         : 3      sec
MESSAGE         : info_analyzed
                  Cldno   : N00    2145323773
```

xiv About this document

Clgno : N00 2143562784
Chgno : N00 2143562784

Introduction to CAIN/FlexDial

Welcome to the CAIN/FlexDial Interaction Application Guide. Beginning with the UCS07 software release, NetworkBuilder is paired with the FlexDial Framework to provide a more provisionable call processing environment on the UCS DMS-250 switch.

NetworkBuilder call processing changes the way the switch handles a call. Instead of controlling all aspects of a call, the switch off-loads some of the call processing functions to an intelligent service control point (SCP). This off-loading can relieve switch responsibilities and allow you greater control over the call services you offer to your subscribers.

The FlexDial Framework also changes the way the switch handles a call by creating an open software system provisioned entirely through datafilled options and features. You provision the interface between your network, the originating switch, and the end subscriber.

Together, NetworkBuilder and FlexDial offer a more flexible approach to both in-switch and SCP processing of a UCS DMS-250 call.

This chapter provides an introduction to NetworkBuilder and the FlexDial Framework. The remainder of this document addresses the interaction between the two software products. Before proceeding, you should understand and be proficient in normal NetworkBuilder (non-FlexDial) call processing, as well as understand the provisioning of the FlexDial Framework.

Note 1: NetworkBuilder supports Bellcore TR-NWT-000533 to perform IN/1 functionality. This document provides information on NetworkBuilder and FlexDial interactions for AXXESS agents. For information on IN/1 functionality, refer to the *UCS DMS-250 NetworkBuilder Application Guide*.

Note 2: For more information on NetworkBuilder, refer to the *UCS DMS-250 NetworkBuilder Application Guide*; for more information on the FlexDial Framework, refer to the *UCS DMS-250 FlexDial Framework Application Guide*.

What is NetworkBuilder?

NetworkBuilder provides optional intelligent networking (IN/1) and optional advanced intelligent networking (AIN) software to the UCS DMS-250 switching platform. The IN/1 system NetworkBuilder supports is based on Bellcore's *TR-NWT-000533* specifications. The AIN system is called Carrier AIN (CAIN). CAIN is based on Bellcore's AIN 0.2 specifications.

IN and AIN

Before the concept of intelligent networking was developed, all network functions were performed in-switch. The development of an intelligent network (IN) off-loaded card validation, subscriber screening, and N00 services to a database located on an IN SCP.

In many cases, this database was a shared resource across a carrier's network of switches. The switches communicated with the SCP through CCS7/TCAP.

In most IN models, there was only one query/response transaction per call. The SCP was used as a remote database without intelligence and never took control of a call.

The next generation of intelligent networking, AIN 0.2, allows call processing to be off-loaded from the SSP to a customer-defined SCP. AIN 0.2 allows call processing to be off-loaded from the SSP to a customer-defined SCP. The AIN SCP is able to take control of a call and direct call processing. This intelligent SCP may also contain feature logic and necessary databases.

AIN network model

You develop the software on your AIN peripherals to interact with AIN software on the SSP. The AIN network may consist of the following AIN peripherals:

- service switching point (SSP)
- service control point (SCP)
- signal transfer point (STP)
- service management system (SMS)
- service creation environment (SCE)
- intelligent peripheral (IP)

An SSP and an SCP are required for an AIN network. Peripheral software development is independent of SSP software development.

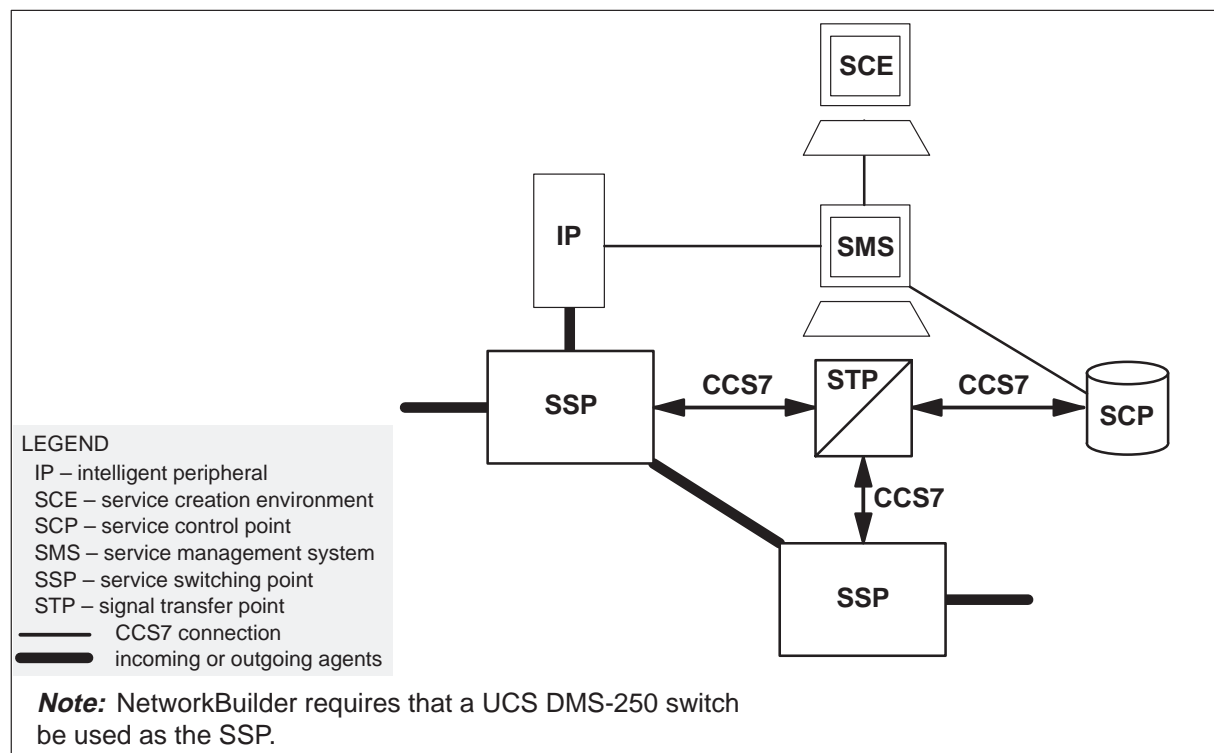
Note 1: NetworkBuilder requires the use of a UCS DMS-250 switch as the SSP.

Note 2: Nortel also offers an SCP product called ServiceBuilder.

Figure 1-1 shows the hardware components available for designing an AIN system. Descriptions of each component follow Figure 1-1.

For more information on the hardware components, refer to the *UCS DMS-250 NetworkBuilder Application Guide*.

Figure 1-1
Example of an AIN network model



Call model

The call model divides the call processing logic into key states (known as a points in call [PIC]). The switch performs a defined set of functions within the PIC.

Once certain events/actions have occurred within the PIC, the call encounters trigger detection points (TDP) and evaluates triggers.

Once call processing encounters a TDP, in-switch logic and datafill is consulted to determine if the trigger criteria is met. When met, the switch can off-load call processing to the SCP as directed by datafill. Datafill can

direct the switch to continue in-switch processing or to suspend call processing and request direction from the SCP.

Event Detection Points (EDPs) are similar to TDPs. When a call queries the SCP, the SCP can return a conversational package identifying one or more EDPs that occur later in the call model. At this point, the EDPs are said to be armed. An EDP is armed to send an EDP-Request or EDP-Notification message when encountered. By arming EDPs, the SCP deactivates triggering until the armed EDPs are deactivated.

The UCS08 and beyond NetworkBuilder software release allows EDPs to be activated in calls originating on AXXESS agents and integrates EDPs into the FlexDial call model. For more information on EDPs, refer to the *UCS DMS-250 NetworkBuilder Application Guide* for more information.

In FlexDial calls, if EDPs are active when the processing of collectables is resumed (due to the “RECOL” or “DPIDX” actions), the switch deactivates EDPs and sends a `close` message to the SCP with `closeCause` value `eDPsCompleted`. This allows the UCS DMS-250 SSP to encounter TDPs after the collectable processing is completed. If route advancing (the “RTEADV” action) occurs, EDPs remain active. Refer to the *UCS DMS-250 FlexDial Framework Application Guide* for more information on the RECOL, DPIDX, and RTEADV actions.

Note: Because the FLEXSIM tool does not simulate NetworkBuilder call processing after the *Analyze_Information* PIC, there is no interaction between FLEXSIM and EDPs.

Subscription

Any call originating on a supported agency (PTS DAL, PTS FGD, SS7 FGD, PRI, Inter-IMT, Global-IMT, or AXXESS) can subscribe to NetworkBuilder services.

Addresses, authcodes, ANIs, SUBR collectables, SUBRPARM collectables, CALLTYPE collectables, originating agents, or the office can subscribe to a CAIN group provisioned in table CAINGRP. The CAIN group, in turn, enables one or more triggers. Enabling a trigger provides an index into the appropriate trigger table where trigger criteria (required to query the SCP) is defined.

Any call terminating to a supported agency (DAL, FGB, FGC, FGD, IMT, AXXESS, or PRI) can subscribe to NetworkBuilder terminating services.

Only terminating agents can be provisioned to subscribe to a terminating CAIN group that enables a trigger in the terminating call model.

Note: In addition, the SCP can return a CAIN group, thereby controlling subscription for the call.

Within the trigger tables, you define the call conditions required (trigger criteria) and the actions the switch takes when the conditions are met. If a call meets the trigger criteria, the switch performs one of the following:

- query the SCP for instructions
- route advance at *Network_Busy*, *O_Called_Party_Busy*, or *O_No_Answer*
- ignore the criteria and continue processing
- continue processing as directed by datafill
- block the call and apply AINF treatment

NetworkBuilder call processing stores up to six subscription groups for use throughout the call. The six groups are determined through:

- SCP-returned CAIN group
- Address subscription (through table STDPRTCT)
- Authorization code subscription (for non-AXXESS originations: table AUTHCODU, AUTHCDU2, AUTHCDU3, AUTHCDU4, AUTHCDU5; for AXXESS originations: through the SUBR collectable, SUBRPARM collectable, or CALLTYPE collectable with the CAINFLG option)
- ANI subscription (for non-AXXESS originations: table ANISCUSP, or tables ANIVAL and UNIPROF; for AXXESS originations: through the SUBR collectable or SUBRPARM collectable with the CALLING option)
- Agent subscription (for non-AXXESS originations: table TRKGRP; for PRI originations: table CALLATTR; for AXXESS originations: table TRKFEAT)
- Office subscription (table CAINPARM)

Supported PICs

NetworkBuilder software supports the following PICs, TDPs, and triggers:

Table 1-1
Supported PICs, TDPs, EDPs, triggers, and events

PIC	TDP/EDP	Trigger/Event
PIC 1: O_Null	Origination_Attempt TDP	<i>Off_Hook_Immediate</i> trigger
PIC 3: Collect_Information	O_Feature_Requested TDP	<i>O_Feature_Requested</i> trigger
	Info_Collected TDP	<i>Tollfree_Service</i> (Note 1)
		<i>Offhook_Delay</i> trigger
		<i>Shared_Interoffice_Trunk</i> trigger
PIC 4: Analyze_Information	O_Abandon EDP	<i>O_Abandon</i> event
	Info_Analyzed TDP	<i>Specific_Feature_Code</i> trigger
		<i>Customized_Dialing_Plan</i> trigger
		<i>Specific_Digit_String</i> trigger
PIC 5: Select_Route	O_Abandon EDP	<i>Office_Code</i> trigger
	Network_Busy TDP	<i>O_Abandon</i> event
	Network_Busy EDP	<i>Network_Busy</i> trigger
	O_Abandon EDP	<i>Network_Busy</i> event
PIC 7: Send_Call	O_Term_Seized EDP	<i>O_Abandon</i> event
<p>Note 1: NetworkBuilder supports Bellcore's <i>TR-NWT-000533</i> toll-free service specifications. The ability to subscribe to multiple CAIN groups allows for a service integration of the IN/1 trigger offerings (<i>Tollfree_Service</i>) defined in <i>TR-NWT-000533</i> and CAIN triggers.</p> <p>Note 2: Since the FlexDial Framework does not support PRI, CAIN/FlexDial interaction does not support <i>PRI_B-Channel</i>.</p>		
—continued—		

Table 1-1
Supported PICs, TDPs, EDPs, triggers, and events (continued)

PIC	TDP/EDP	Trigger/Event
PIC 8: O_Alerting	<i>O_Called_Party_Busy</i> TDP	<i>O_Called_Party_Busy</i> trigger
	<i>O_Called_Party_Busy</i> EDP	<i>O_Called_Party_Busy</i> event
	<i>O_Mid_Call</i> TDP	<i>O_IEC_Reorigination</i> trigger
	<i>O_Mid_Call</i> EDP	<i>Switch_Hook_Flash</i> event
	<i>O_Abandon</i> EDP	<i>O_Abandon</i> event
	<i>O_Answer</i> EDP	<i>O_Answer</i> event
	<i>O_No_Answer</i> TDP	<i>O_No_Answer</i> trigger
	<i>O_No_Answer</i> EDP	<i>O_No_Answer</i> event
PIC 9: O_Active	<i>O_Mid_Call</i> TDP	<i>O_IEC_Reorigination</i> trigger
	<i>O_Mid_Call</i> EDP	<i>Switch_Hook_Flash</i> event
	<i>O_Abandon</i> EDP	<i>O_Abandon</i> event
	<i>O_Disconnect</i> EDP	<i>O_Disconnect</i> event
PIC 10: O_Suspended	<i>O_Mid_Call</i> TDP	<i>O_IEC_Reorigination</i> trigger
	<i>O_Mid_Call</i> EDP	<i>Timeout</i> event
	<i>O_Mid_Call</i> EDP	<i>Switch_Hook_Flash</i> event
	<i>O_Disconnect</i> EDP	<i>O_Disconnect</i> event
	<i>O_Mid_Call</i> TDP	<i>O_IEC_Reorigination</i> trigger
<p>Note 1: NetworkBuilder supports Bellcore's <i>TR-NWT-000533</i> toll-free service specifications. The ability to subscribe to multiple CAIN groups allows for a service integration of the IN/1 trigger offerings (<i>Tollfree_Service</i>) defined in <i>TR-NWT-000533</i> and CAIN triggers.</p> <p>Note 2: Since the FlexDial Framework does not support PRI, CAIN/FlexDial interaction does not support <i>PRI_B-Channel</i>.</p>		
—continued—		

Table 1-1
Supported PICs, TDPs, EDPs, triggers, and events (continued)

PIC	TDP/EDP	Trigger/Event
	<i>O_Mid_Call</i> EDP	<i>Timeout</i> event
PIC 11: <i>T_Null</i>	<i>Termination_Attempt</i> TDP	<i>Termination_Attempt</i> trigger
<p>Note 1: NetworkBuilder supports Bellcore's <i>TR-NWT-000533</i> toll-free service specifications. The ability to subscribe to multiple CAIN groups allows for a service integration of the IN/1 trigger offerings (<i>Tollfree_Service</i>) defined in <i>TR-NWT-000533</i> and CAIN triggers.</p> <p>Note 2: Since the FlexDial Framework does not support PRI, CAIN/FlexDial interaction does not support <i>PRI_B-Channel</i>.</p>		
—end—		

SCP interaction

NetworkBuilder call processing interacts with the SCP to determine how the call is handled. The SCP may direct the switch to collect digits, apply treatment and disconnect, play announcements, connect to an IP, or route the call according to SCP instructions.

To throttle the switch during periods of congestion, manual and automatic code gapping (ACG) can be implemented. Manual and automatic code gapping methods reduce the number of outgoing queries from the switch to a given SCP. The SCP provides the switch with information on which queries to block by sending a message containing an ACG Control. The switch stores the controls in a control list. The control list is searched before sending each query message to determine if that query should be blocked.

What is the FlexDial Framework?

The FlexDial Framework is an optional software package that allows you to define the interface between your switch and the originating switch.

Provisioning areas

There are three provisioning areas in the FlexDial framework:

- agent provisioning
- dialplan interaction provisioning
- subscriber and call type feature provisioning

Agent provisioning

The AXXESS agent is datafilled, as any other agent, in table TRKGRP. However, the signaling requirements are provisioned in table TRKSIG. The AXXESS agent can be datafilled as a unique agent, or can mimic an existing

agent when combined with an appropriately datafilled dialplan. AXXESS can mimic the following agents: DAL, EDAL, DAL TIE, 2-wire ONALs, 4-wire ONALs, ONAT, PTS FGD, and SS7 FGD.

Dialplan interaction provisioning

In the past, dialing plans have been hard-coded and linked to specific originating agents. However, with the FlexDial Framework, you can create dialing plans to meet your own unique needs.

Subscriber and call type feature provisioning

Subscriber provisioning allows you to identify features, characteristics, and billing information for subscriber numbers. In addition, you can provision call type data independently of the subscriber numbers.

Table control

The FlexDial Framework consists of a complex system of table control. Figure 1-2 shows the three provisioning areas interact with each other. These interactions or associations are listed on a per-table basis, where one table contains an index or reference to another table. Figure 1-3 identifies the complete matrix of interactions.

Figure 1-2
FlexDial framework provisioning model

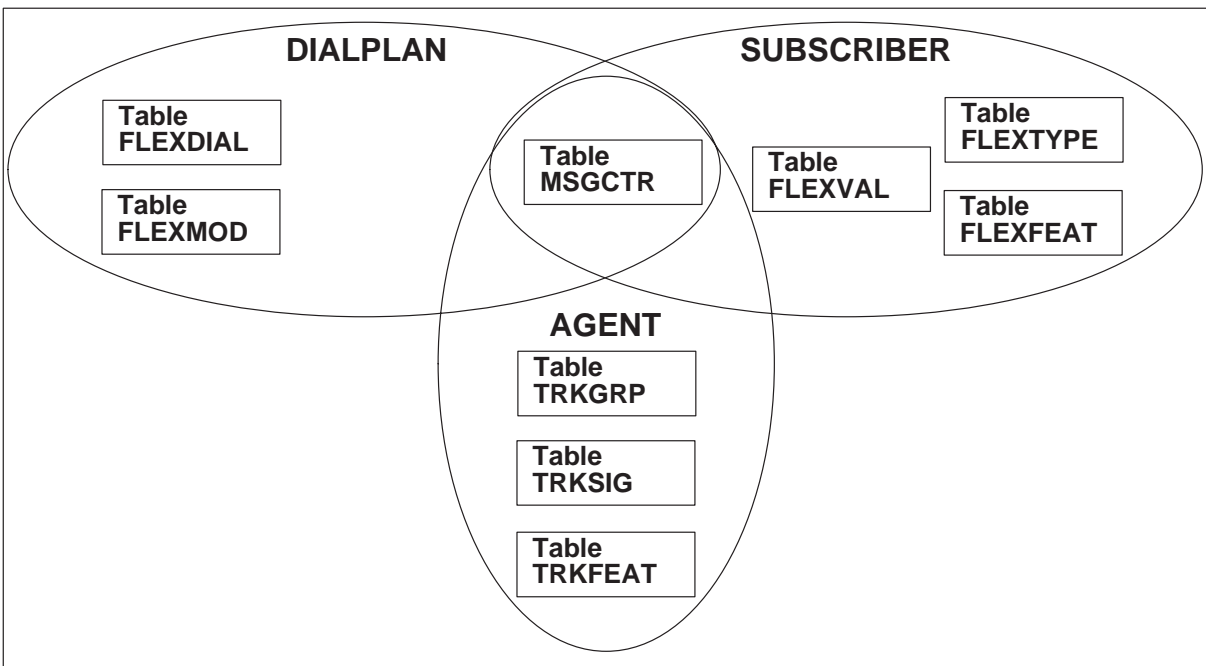


Figure 1-3
FlexDial provisioning framework table associations

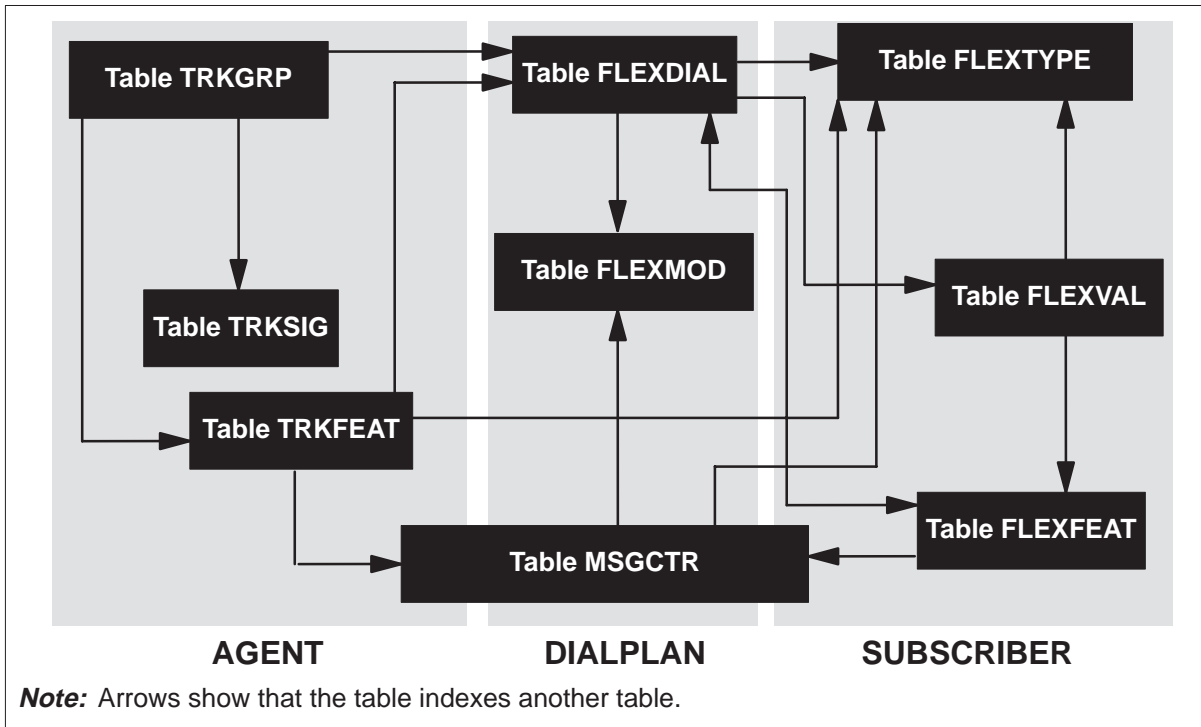


Table 1-2 provides FlexDial framework provisioning system table associations.

Table 1-2
Provisioning system table associations

Table Association	Rationale
TRKGRP → TRKSIG	The TRKSIG table identifies the provisioned signaling information for the trunk group. This table replaces the use of TRKSGRP for trunk group signaling datafill for AXXESS trunk group types. The SIGIDX field identifies the index in table TRKSIG that must be previously provisioned. Many trunk groups can share the same TRKSIG index.
TRKGRP → TRKFEAT	The TRKFEAT table identifies the provisioned features and characteristics of the trunk group. Features and characteristics are no longer provisioned in the TRKGRP and TRKGRP1 tables. The FEATIDX field identifies the index in table TRKFEAT that must be previously provisioned. Many trunk groups can share the same TRKFEAT index.

Table 1-2
Provisioning system table associations (continued)

Table Association	Rationale
TRKGRP → FLEXDIAL	The trunk group tuple DPIDX field defines the initial dialing plan defined interaction with the originating agent.
TRKFEAT → FLEXDIAL	The COSOVE option identifies a FLEXDIAL table index to provide a new thread of execution for FLEXDIAL call processing.
TRKFEAT → MSGCTR	Through the MSGCTR option, the TRKFEAT table contains indexes into the MSGCTR table.
TRKFEAT → FLEXTYPE	Through the REORIGAL option, table TRKFEAT identifies subscriber number types provisioned in the FLEXTYPE table that require revalidation on reoriginated calls.
FLEXDIAL → FLEXMOD	The MODDIGS sequence collectable contains an index into table FLEXMOD to provide a table lookup scheme for modifying digits.
FLEXDIAL → FLEXTYPE	The SUBR and CALLTYPE collectables in the FLEXDIAL table contain a FLEXTYPE table index that identifies the type of subscriber number being collected or call type features and characteristics being set.
FLEXDIAL → FLEXVAL	The SUBR digit collectable in the FLEXDIAL table provides a numeric index for validating the subscriber number digits in table FLEXVAL.
FLEXDIAL → FLEXFEAT	The CALLTYPE collectable identifies a FLEXFEAT table index outlining features and characteristics that apply to the call. The CLRFTRS collectable identifies a list of FLEXFEAT defined features and characteristics that are to be reset (to their default value).
MSGCTR → FLEXTYPE	Tuples in the MSGCTR table can reference indexes in the FLEXTYPE table.
MSGCTR → FLEXMOD	The MSGCTR MODDIGS message type contains an index into the FLEXMOD table.
FLEXVAL → FLEXFEAT	The FLEXVAL table FEATIDX field identifies an index into the FLEXFEAT table. This index provides the feature and characteristic provisioning information for the subscriber number that is used to index the FLEXVAL table.
—continued—	

Table 1-2
Provisioning system table associations (continued)

Table Association	Rationale
FLEXVAL → FLEXTYPE	A part of the FLEXVAL table key consists of a valid FLEXTYPE table index.
FLEXFEAT → FLEXDIAL	The DPIDX and REORIGACT FLEXFEAT table options identify an index into the FLEXDIAL table. The FLEXDIAL index provides a new thread of execution for FlexDial call processing.
FLEXFEAT → MSGCTR	Through the MSGCTR option, the FLEXFEAT table contains indexes into the MSGCTR table. Additionally, the REORGACT option contains a MSGCTR table index that is applied on reoriginated calls.
—end—	

Why CAIN/FlexDial?

CAIN/FlexDial interaction enhances the existing NetworkBuilder call processing functions by

- supporting calls originating and/or terminating on an AXXESS agent
- allowing subscription on any subscriber digits (such as authorization codes, personal identification numbers, account codes, or calling card numbers)
- allowing subscription based on the dialplan (using subscription by call type)

Interactions with the FlexDial Framework changes NetworkBuilder call processing in the following ways:

- allows originations from AXXESS agents
- provides new subscription methods for FlexDial collectables
- recognition of digits collected by the FlexDial Framework

The following changes were made to the FlexDial Framework in order to allow interaction with NetworkBuilder call processing:

- support for CAIN-capable AXXESS originations in table TRKFEAT
- allows subscription to CAIN groups in tables FLEXFEAT and TRKFEAT through the CAINGRP option

- adds the CAINFLG option to table FLEXTYPE. The CAIN flag maps the subscriber number to a digit type that NetworkBuilder can understand.

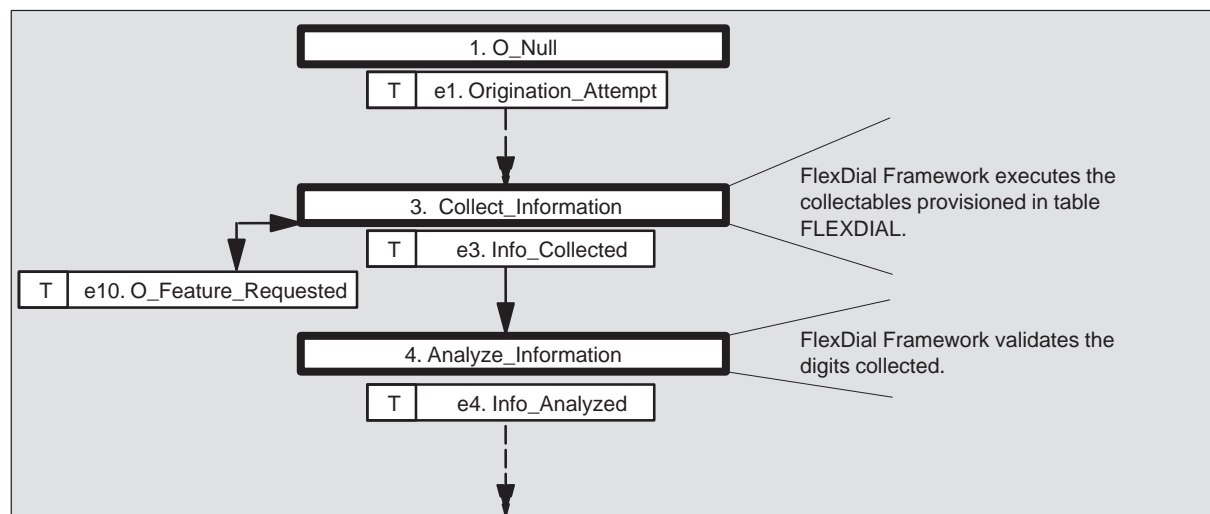
NetworkBuilder interacts with the FlexDial Framework in two ways:

- supporting originating AXXESS agents (which requires use of the datafillable dialplans in the FlexDial Framework)
- digit collection provided through the datafillable dialplans

Digit collection and digit validation

The FlexDial Framework provides the digit collection and digit validation for originating AXXESS agents. The majority of the interaction between NetworkBuilder and FlexDial occurs during **Collect_Information** and **Analyze_Information** (Figure 1-4).

Figure 1-4
NetworkBuilder call model



For more information on the NetworkBuilder call model, refer to the *UCS DMS-250 NetworkBuilder Application Guide*.

Software optionality control

Software optionality control (SOC), part of the DMS Evolution product delivery process, controls the delivery of product computing module loads (PCL). All features in a PCL are categorized as either base or optional. Base applications are available for immediate use; optional applications are

grouped into commercial units called SOC options. SOC options can be purchased by operating companies.

ATTENTION

CAIN/FlexDial interaction does not require additional SOCs.

The following NetworkBuilder and FlexDial SOC options are available in the UCS11 software release:

- NetworkBuilder SOC options
 - CAIN0100 (Messages)
 - CAIN0200 (Extension ParmS)
 - CAIN0300 (SCP Simulator)
 - CAIN0400 (Test Query Tool)
 - CAIN0500 (CUSTDP Trigger)
 - CAIN0501 (SPECDIG Trigger)
 - CAIN0502 (OFFHKIM Trigger)
 - CAIN0503 (SIOTRK Trigger)
 - CAIN0504 (PRIBCHNL Trigger)

Note: CAIN/FlexDial interaction does not support the *PRI_B-Channel* trigger.

- CAIN0505 (ONOANSWER Trigger)
- CAIN0506 (NETBUSY Trigger)
- CAIN0507 (OCLDBUSY Trigger)
- CAIN0508 (OFTRREQ Trigger)
- CAIN0509 (OIECREO Trigger)
- CAIN0510 (TERMATT Trigger)
- CAIN0511 (SPECFEAT Trigger)
- CAIN0512 (OFFHKDEL Trigger)
- CAIN0513 (TOLLFREE Trigger)
- CAIN0600 (Con Digit Collect)
- CAIN0601 (SCP Trigger Sub)
- CAIN0602 (EDPs)
- CAIN0603 (STR Connection)

- CAIN0604 (Inter IMT Support)
- CAIN0605 (Global IMT Support)
- CAIN0606 (1129-Style IP)
- CAIN0607 (Virtual IP)
- CAIN0609 (Term Notification)
- CAIN0610 (CAINPRT Digit Coll)
- CAIN0700 (LNP QOO)
- CAIN0800 (Mid-Call Services 1)
- CAIN0801 (Mid-Call Services 2)
- CAIN0802 (Takeback & Transfer)
- CAIN0900 (Auto Code Gapping)
- CAIN0901 (Manual Code Gapping)
- FlexDial SOC option
 - UTRS0002 (Flexible Dial Plans)
- FlexCDR SOC option
 - UBFR0001 (Flexible CDRs)

Note: Although the UBFR0001 SOC is not required, Nortel recommends its purchase when combining FlexDial and NetworkBuilder.

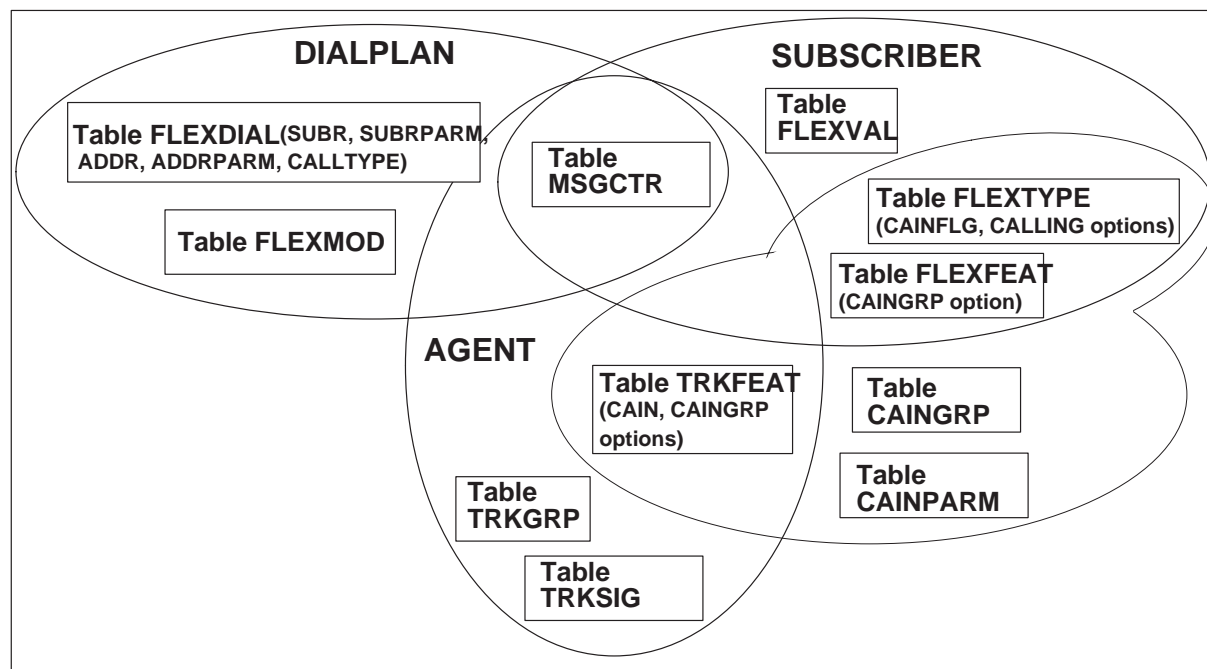
Refer to the *UCS DMS-250 Software Optionality Control User's Manual* for more information about SOCs.

Provisioning NetworkBuilder for AXCESS originations

Implementing the integrated NetworkBuilder and FlexDial services requires a base knowledge of table control, the originating and terminating call models, CCS7 network connections, subscription, triggering, messaging, resource allocation requirements, announcement and tone requirements, and AXCESS agent provisioning.

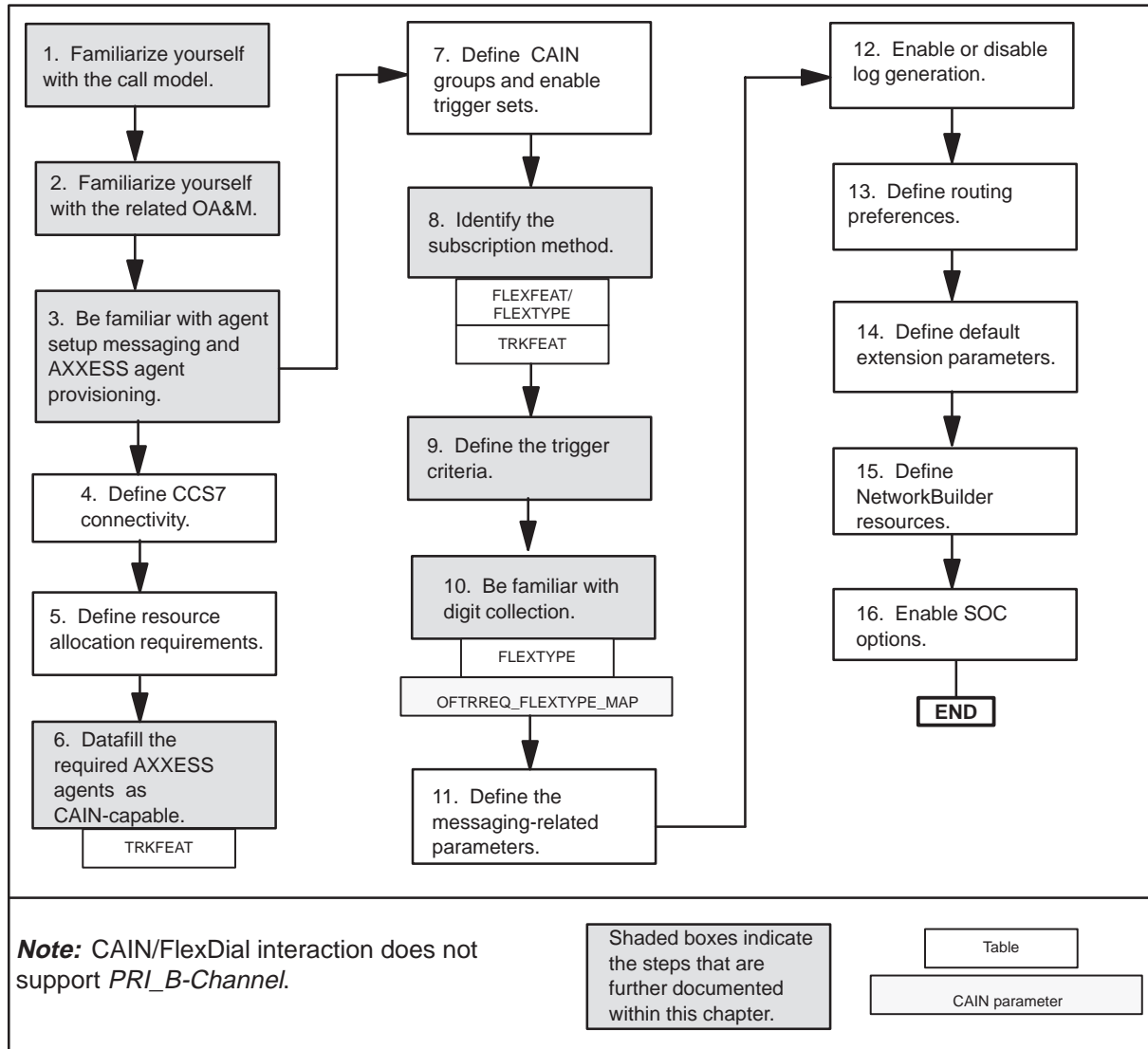
Figure 2-1 shows the FlexDial framework with NetworkBuilder interactions.

Figure 2-1
FlexDial framework provisioning model



This chapter provides a base level knowledge required to activate NetworkBuilder/FlexDial services into your network. Figure 2-2 shows a basic flow of the steps you should take to provision your switch.

Figure 2-2
Provisioning NetworkBuilder for AXXESS interactions



For information on activating NetworkBuilder, refer to the *UCS DMS-250 NetworkBuilder Application Guide*. For information on activating FlexDial, refer to the *UCS DMS-250 FlexDial Framework Application Guide*.

Provisioning

The remainder of this chapter shows you how to provision your switch for NetworkBuilder services originating from an AXXESS agent. Figure 2-3 shows how sections are titled and divided.

Figure 2-3
Example of a provisioning section

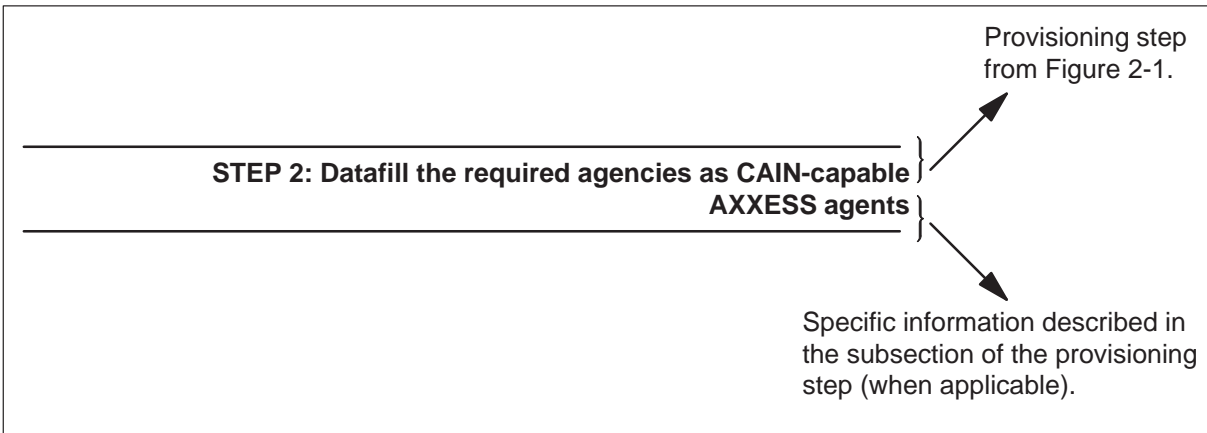


Table 2-1 shows the layout of the rest of this chapter and the information available.

Table 2-1
Chapter layout

Step	Title	Page
1	Familiarize yourself with the call model	see Note
2	Familiarize yourself with related OA&M	2-35
	Treatments	2-36
	Billing	2-38
	Logs	see Note
	OMs	see Note
	DS	see Note
3	Be familiar with agent setup messaging	2-41
Note: These steps remain the same for CAIN/FlexDial interactions. Refer to the <i>UCS DMS-250 NetworkBuilder Application Guide</i> for more information on these steps.		
—continued—		

2-4 Provisioning NetworkBuilder for AXXESS originations

Table 2-1
Chapter layout (continued)

Step	Title	Page
4	Define CCS7 connectivity	see Note
5	Define resource allocation requirements	see Note
6	Datafill the AXXESS agents as CAIN-capable originating AXXESS agents	2-44 2-46
7	Define CAIN groups and enable trigger sets	see Note
8	Choose the type of subscription	2-47
9	Define the trigger criteria	2-83
10	Be familiar with digit collection OFTRREQ_FLEXTYPE_MAP	2-86 2-93
11	Define the messaging-related parameters	see Note
12	Enable or disable log generation	see Note
13	Define routing preferences	see Note
14	Define default extension parameters	see Note
15	Define NetworkBuilder resources	see Note
16	Enable SOC options	see Note
Note: These steps remain the same for CAIN/FlexDial interactions. Refer to the <i>UCS DMS-250 NetworkBuilder Application Guide</i> for more information on these steps.		
—end—		

Step 1: Familiarize yourself with the call model

The following point in calls (PICs), detection points (DPs), triggers, and trigger detection points (TDP)/ event detection points (EDPs) relate to NetworkBuilder/FlexDial interactions:

- PIC 1: **O_Null**
 - **Origination_Attempt** TDP
 - *Off_Hook_Immediate* trigger
- PIC 3: **Collect_Information**
 - **O_Feature_Requested** TDP
 - *O_Feature_Requested* trigger
 - **Info_Collected** TDP
 - *Offhook_Delay* trigger
 - *Shared_Interoffice_Trunk* trigger

Note: CAIN/FlexDial Interaction does not support the *PRI_B-Channel* trigger.
- PIC 4: **Analyze_Information**
 - **Info_Analyzed** TDP
 - *Specific_Feature_Code* trigger
 - *Customized_Dialing_Plan* trigger
 - *Specific_Digit_String* trigger
 - *Office_Code* trigger
- PIC 5: **Select_Route**
 - **Network_Busy** DP
 - *Network_Busy* trigger/event
- PIC 7: **Send_Call**
 - **O_Called_Party_Busy** DP
 - *O_Called_Party_Busy* trigger/event
- PIC 8: **O_Alerting**
 - **O_No_Answer** DP
 - *O_No_Answer* trigger/event

This section describes the above PICs. For detailed information on the NetworkBuilder call model and comprehensive information on PICS, DPs, and TDPs/EDPs, see the *UCS DMS-250 NetworkBuilder Application Guide*.

Step 1: Familiarize yourself with the call model

PIC 1: O_Null

Call processing enters **O_Null** when the switch detects off-hook.

Origination_Attempt TDP

Origination_Attempt is encountered when a call is initiated on the originating agent.

When **Origination_Attempt** is encountered, the NetworkBuilder directs the switch to check the following:

- call subscription to a CAIN group through table TRKFEAT, TRKGRP, or CAINPARAM

Note: Table TRKGRP is not used for agent subscription on CAIN/FlexDial Interactions; use table TRKFEAT to provision agent subscription.

- enabled triggers (Triggers are enabled through table CAINGRP.)

Note 1: NetworkBuilder software supports the *Off_Hook_Immediate* trigger at **Origination_Attempt**.

Note 2: NetworkBuilder software supports the *Off_Hook_Immediate* for AXCESS originations with FXS or FXO signaling.

Off_Hook_Immediate trigger

ATTENTION

The *Off_Hook_Immediate* trigger requires the CAIN0502 SOC option. Refer to the *UCS DMS-250 Software Optionality Control User's Manual* for more information about SOCs.

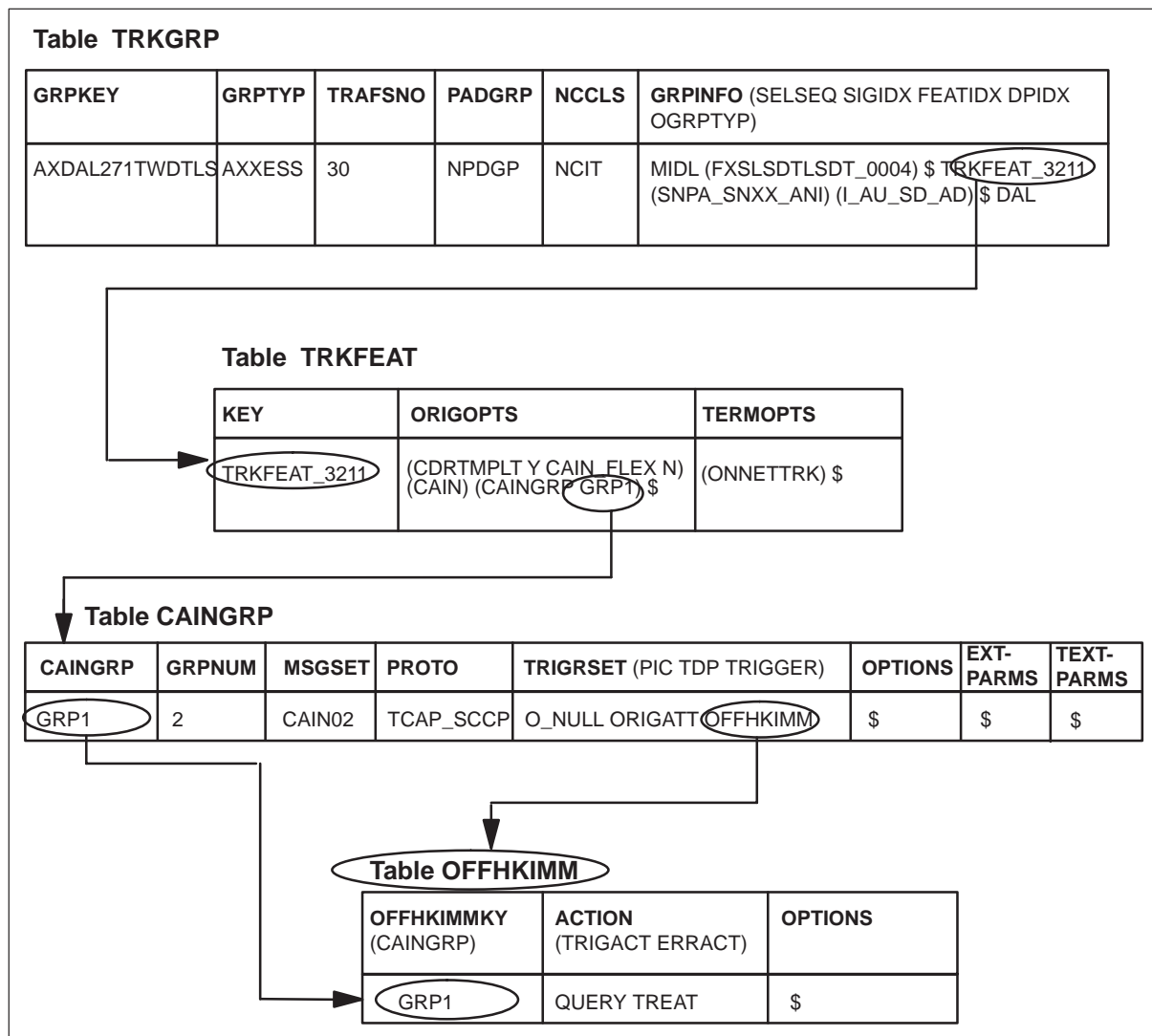
Datafill

The following figure shows how a subscription table interacts with the *Off_Hook_Immediate* trigger table (OFFHKIMM).

Step 1: Familiarize yourself with the call model

PIC 1: O_Null (end)

Subscription-OFFHKIMM table interaction



Note: Refer to the *UCS DMS-250 NetworkBuilder Application Guide* for more information on provisioning the *Off_Hook_Immediate* trigger and table OFFHKIMM.

Step 1: Familiarize yourself with the call model

PIC 3: Collect_Information

Call processing enters **Collect_Information** once the following are completed:

- The originating agent is identified.
- CDR is started and contains the following data: ORIGTIME, ORIGDATE, ORIGMEM, and ORIGAMPM.
- The resources required to handle the call are allocated.
- Authorize origination is detected.

The majority of interaction with the FlexDial Framework occurs during **Collect_Information**. **Collect_Information** corresponds to the switch collecting the data provisioned in table FLEXDIAL.

O_Feature_Requested TDP

The **O_Feature_Requested** TDP is encountered when the address digits are collected.

Note 1: This TDP is encountered when the switch collects the digits that lead to the determination of the address associated with the terminating party.

Note 2: The **O_Feature_Requested** TDP can be encountered up to three times. For example, the switch may collect a UA number or speed number as the address before collecting the called party's address.

When **O_Feature_Requested** is encountered, the NetworkBuilder directs the switch to check for the following:

- call subscription to a CAIN group through table STDPRTCT, AUTHCODU, ANISCUSP (or ANIVAL and UNIPROF), FLEXTYPE, FLEXFEAT, TRKFEAT, TRKGRP, CALLATTR, or CAINPARAM

Note: Tables AUTHCODU, ANISCUSP (or ANIVAL and UNIPROF), TRKGRP and CALLATTR are not used for subscription on CAIN/FlexDial Interactions.

- enabled triggers (Triggers are enabled through table CAINGRP.)

Note: NetworkBuilder software supports the **O_Feature_Requested** trigger at **O_Feature_Requested**.

Note: AXCESS originations require the VALIDATE option on the ADDR and ADDRPARAM collectables in order to trigger at **O_Feature_Requested**.

Step 1: Familiarize yourself with the call model

PIC 3: Collect_Information (continued)

O_Feature_Requested trigger

ATTENTION

The *O_Feature_Requested* trigger requires the CAIN0508 SOC option. Refer to the *UCS DMS-250 Software Optionality Control User's Manual* for more information about SOCs.

Reorigination identifier

The reorigination option (REORIG) is only available at *O_Feature_Requested* and forces a reoriginated call to automatically query with no additional trigger criteria evaluations. Reoriginated calls trigger once a new address is collected. The following reorigination behavior and *O_Feature_Requested* interaction apply to the use of the reorigination option:

- Once the address is collected on a reoriginated call, the switch queries the SCP. SCP digit requirements are fulfilled through conversational digit collection directed by the SCP.

Note 1: FlexDial provisioning may instruct the switch to collect additional digits upon reorigination. Depending on FlexDial provisioning, the switch may recollect non-address digits that can affect NetworkBuilder queries. For more information, refer to Chapter 3. "Call processing."

Note 2: *O_Feature_Requested* datafill is not consulted again. A query is sent based on the original call. The address sent in the query is the number associated with the reoriginated call.

- Upon reorigination, normal call processing resets the account code and the PIN. Therefore, the reoriginated call's query to the SCP will not contain an account code or a PIN. However, account codes and pins may be included in the query if:
 - the switch collected an account code and PIN as directed by table FLEXDIAL for the original call
 - the switch collected an account code and PIN as directed by table FLEXDIAL for the reoriginated call
 - the switch collected an account code and PIN as directed by *O_Feature_Requested* datafill

Step 1: Familiarize yourself with the call model

PIC 3: Collect_Information (continued)

Additional information

The following apply to reoriginated calls at *O_Feature_Requested*:

- When the *O_Feature_Requested* ADDR collectable is not executed on the original call, normal (non-NetworkBuilder) reorigination occurs and dialplan execution continues at the index into table FLEXDIAL (provisioned by the REORGACT option (in table FLEXFEAT) on the associated ANI validation performed on the call).
- If the pretranslator option is datafilled, the specified pretranslator is used on all reoriginated calls (NetworkBuilder or non-NetworkBuilder).

CDR population

The following table shows the collectables with their associated CDR field.

CDR fields

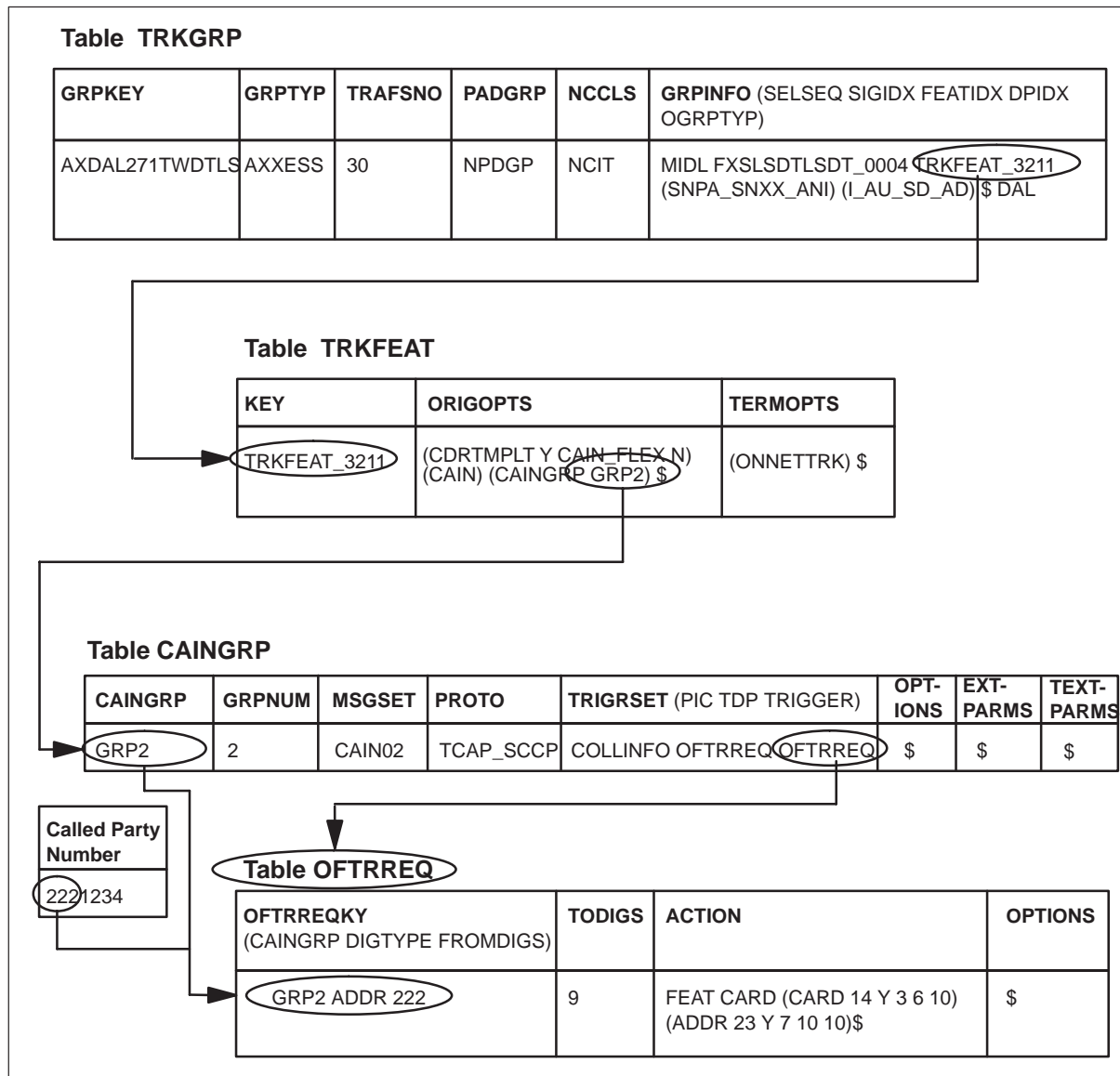
<i>O_Feature_Requested</i> collectable	CDR field
CARD	BILLNUM
AUTH	BILLNUM
PIN	PINDIGS
ACCT	ACCTCD
ADDR	CALLEDNO
—end—	

Datafill

The following figure shows how a subscription table interacts with the *O_Feature_Requested* trigger table (OFTRREQ).

Step 1: Familiarize yourself with the call model PIC 3: Collect_Information (continued)

Subscription-OFTRREQ table interaction



Note: Refer to the *UCS DMS-250 NetworkBuilder Application Guide* for more information on provisioning the *O_Feature_Requested* trigger and table OFTRREQ.

Step 1: Familiarize yourself with the call model
PIC 3: Collect_Information (continued)

Tollfree_Service trigger

Subscribing to the Tollfree_Service trigger

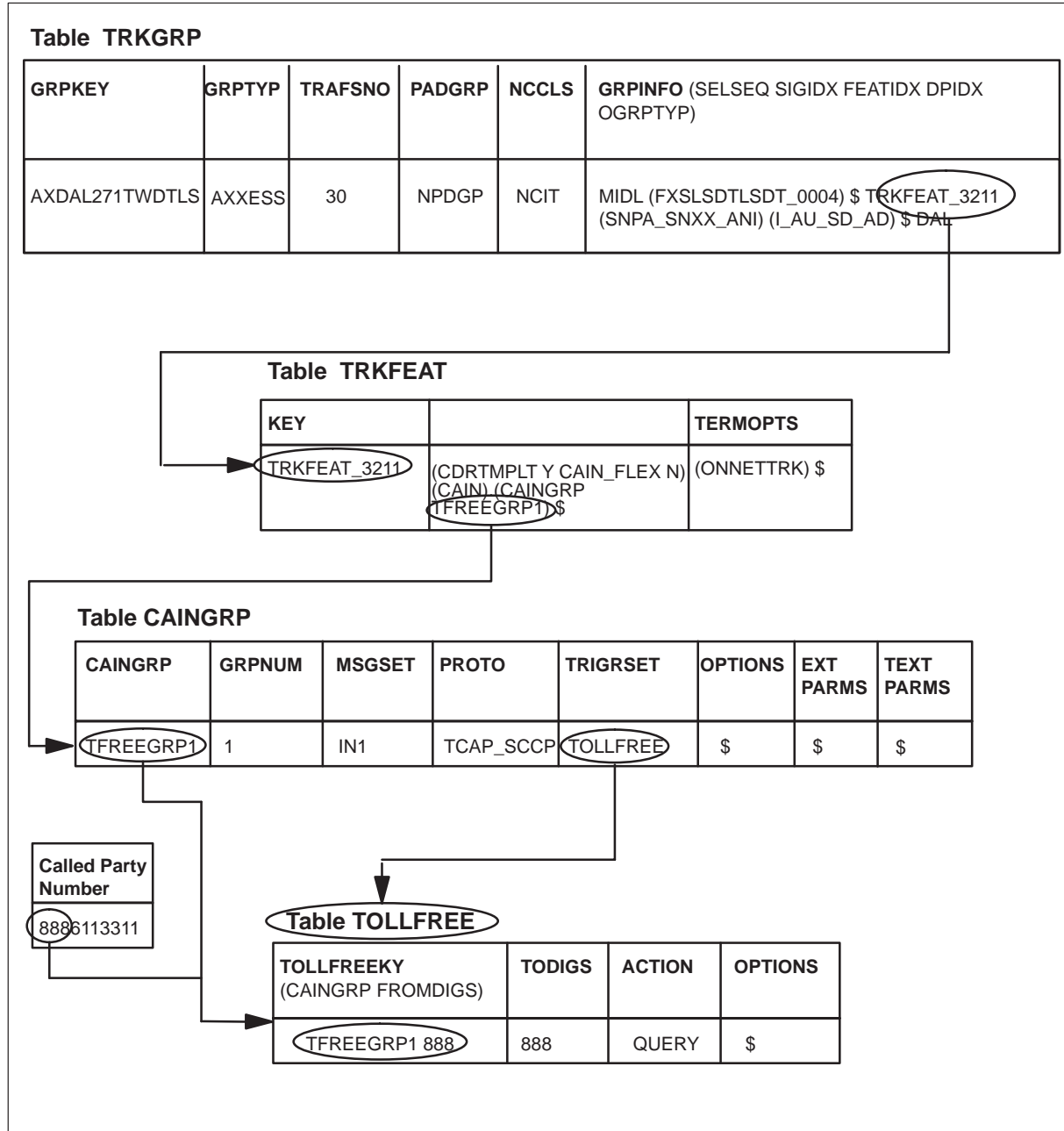
Subscription to the *Tollfree_Service* trigger is available on an agent basis (table TRKFEAT).

Datafill

The following figure shows how a subscription table interacts with the *Tollfree_Service* trigger table (TOLLFREE).

Step 1: Familiarize yourself with the call model
PIC 3: Collect_Information (continued)

Subscription-TOLLFREE table interaction



Step 1: Familiarize yourself with the call model

PIC 3: Collect_Information (continued)

Provisioning the Tollfree_Service trigger

At the CI prompt

- 1 Provision the originating agent as CAIN-capable
- 2 Subscribe to a CAIN group (table STDPRTCT, authcode tables, ANI tables, TRKGRP, or CAINPARAM)
- 3 Datafill CAIN group trigger subscription to the (TOLLFREE) trigger set (table CAINGRP).
- 4 Enter table TOLLFREE.
- 5 Define the trigger criteria for a CAIN group by using the following format:

>ADD tollfreeky todigs action options

where

- tollfreeky** is comprised of three subfields: CAINGRP and FROMDIGS, where
- CAINGRP is the CAIN group that enables the trigger (from table CAINGRP).
- FROMDIGS is the first number used to define the range of digits (0 to 9, *, #).
- todigs** is the second number used to define the range of digits (0 to 9, *, #).
- action** is the trigger action taken when the specified digits are within the FROMDIGS-TODIGS range (BLOCK, IGNORE, QUERY).

- 6 Datafill the OPTIONS, where

- options** is a vector of options. Currently, only the TRTMT option is supported.
- OPTION is the TRTMT option. Enter the TRTMT when the ACTION is BLOCK. When the ACTION is not BLOCK enter \$.
- TREATMENT is the treatment code. Enter the TRTMT when the OPTION is TRTMT. When the TRTMT option is not provisioned this option is not available.

Sample entry: **>ADD tfreegrp1 888 888 block trtmt \$**

Tollfree_Service trigger criteria is defined.

Step 1: Familiarize yourself with the call model

PIC 3: Collect_Information (continued)

Info_Collected TDP

Call processing encounters *Info_Collected* when the following conditions are met:

- dialing plan is identified and executed
- IAM received (SS7 originations)
- the switch has collected and validated digits (as directed by datafill in table FLEXDIAL)

When *Info_Collected* is encountered, NetworkBuilder directs the switch to check the following:

- call subscription to a CAIN group through table STDPRTCT, AUTHCODU, ANISCUSP (or ANIVAL and UNIPROF), FLEXTYPE, FLEXFEAT, TRKFEAT, TRKGRP, CALLATTR, or CAINPARM

Note: Tables AUTHCODU, ANISCUSP (or ANIVAL and UNIPROF), TRKGRP and CALLATTR are not used for subscription on CAIN/FlexDial interactions.

- enabled triggers (Triggers are enabled through table CAINGRP.)

Note: NetworkBuilder software supports the *Shared_Interoffice_Trunk*, *Offhook_Delay*, and *PRI_B-Channel* triggers at *Info_Collected*.

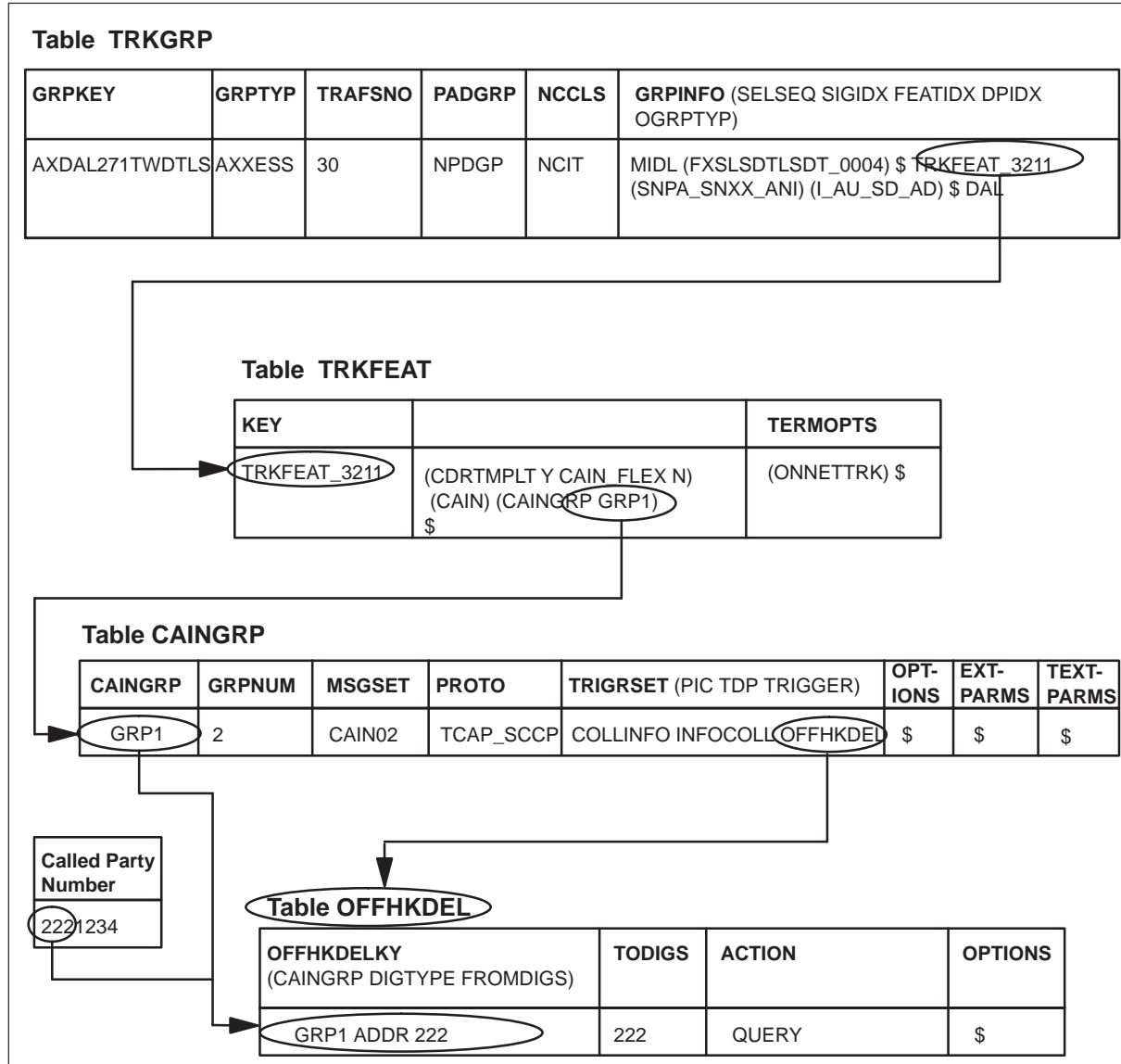
Offhook_Delay trigger

Datafill

The following figure shows how a subscription table interacts with the *Offhook_Delay* trigger table (OFFHKDEL).

Step 1: Familiarize yourself with the call model
PIC 3: Collect_Information (continued)

Subscription-OFFHKDEL table interaction



Shared_Interoffice_Trunk trigger

ATTENTION

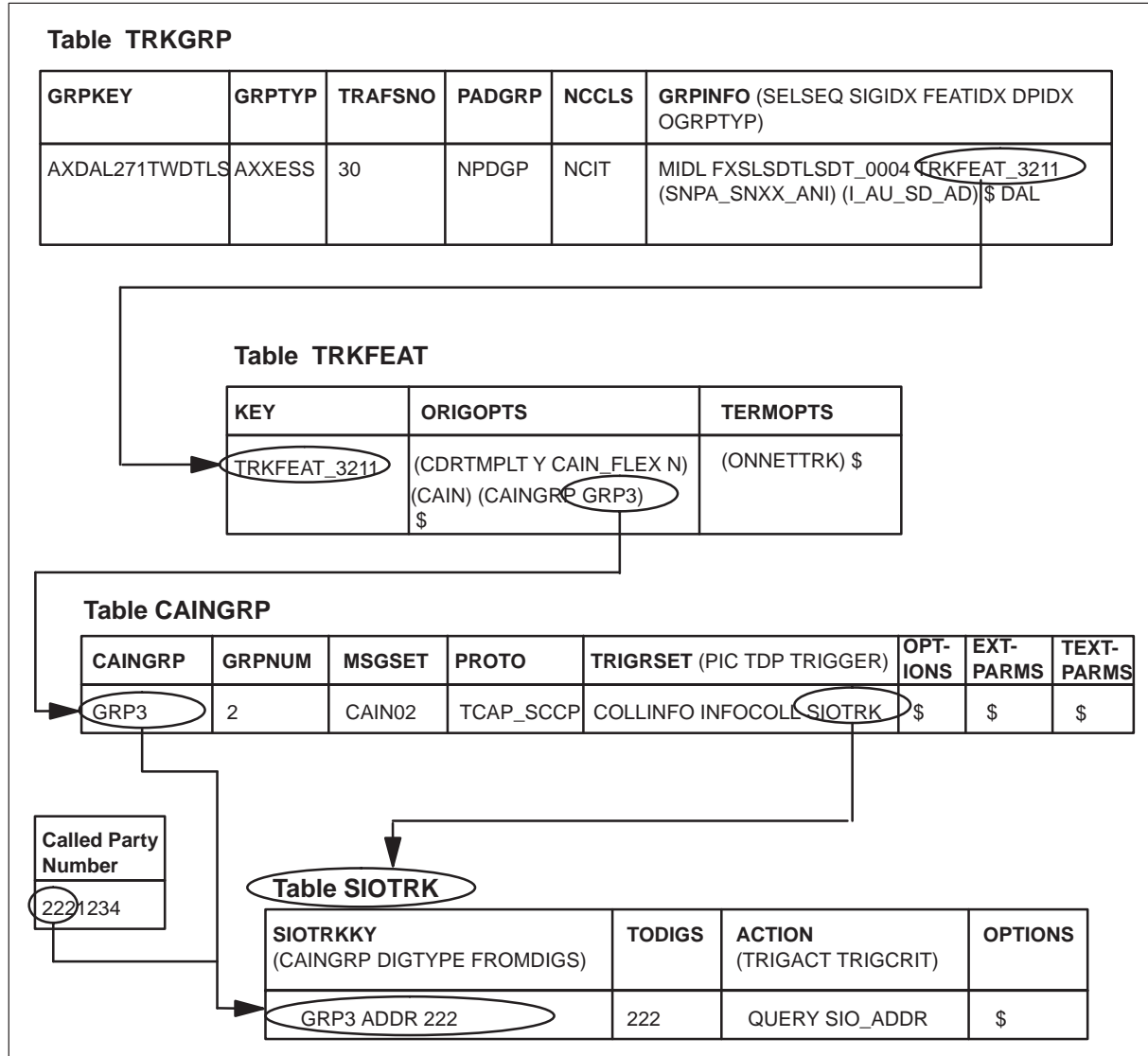
The *Shared_Interoffice_Trunk* trigger requires the CAIN0503 SOC option. Refer to the *UCS DMS-250 Software Optionality Control User's Manual* for more information about SOCs.

Step 1: Familiarize yourself with the call model PIC 3: Collect_Information (continued)

Datafill

The following figure shows how a subscription table interacts with the *Shared_Interoffice_Trunk* trigger table (SIOTRK).

Subscription-SIOTRK table interaction



Note: Refer to the *UCS DMS-250 NetworkBuilder Application Guide* for more information on provisioning the *Shared_Interoffice_Trunk* trigger and table SIOTRK.

Step 1: Familiarize yourself with the call model

PIC 3: Collect_Information (end)

PRI_B-Channel trigger

ATTENTION

CAIN/FlexDial interaction does not support the *PRI_B-Channel* trigger.

Note: Refer to the *UCS DMS-250 Software Optionality Control User's Manual* for more information about SOCs.

Step 1: Familiarize yourself with the call model

PIC 4: Analyze_Information

Call processing enters **Analyze_Information** when the switch begins analyzing and translating the collected digits to determine a route index. Specifically, call processing enters **Analyze_Information** once the dialing plan is completed and the switch has:

- collected data from the initial address message (IAM) of the originating SS7 agent
 - collected digits from a PTS agent
 - collected data from the SETUP message of the originating PRI agency
- Note:* FlexDial does not support mimicking of PRI originations.
- performed N00 digit validation
 - performed FlexDial-collected digit validation

ATTENTION

NetworkBuilder call processing always encounters **Info_Analyzed**, whether or not a route index is identified through normal translations (where the called party number is provisioned in a translation table such as HNPACONT). By not identifying a route index, the switch allows the SCP to control routing. When the switch does not identify a route index, the SCP must provide a route index in an **Analyze_Route** message. If the SCP responds with a **Continue** message, treatment is set to indicate a translation failure.

Call processing performs any remaining digit validation during **Analyze_Information**.

Info_Analyzed TDP

Call processing encounters **Info_Analyzed** once the switch has attempted to identify a route index. Specifically, call processing encounters **Info_Analyzed** when

- the FlexDial Framework completes execution of the dialing plan
- if the switch did not collect address digits (the dialplan may not request them), potential route choices are calculated from the collected information or FlexDial datafill may have identified a route by executing the ROUTE collectable
- in order to trigger, some digits must be available (either dialed or filed)

Step 1: Familiarize yourself with the call model

PIC 4: Analyze_Information (continued)

Instead of immediately seizing an outgoing agent, NetworkBuilder directs the switch to check for the following:

- call subscription to a CAIN group through table STDPRTCT, FLEXTYPE, FLEXFEAT, TRKFEAT, or CAINPARAM

Note: Tables AUTHCODU, ANISCUSP (or ANIVAL and UNIPROF), TRKGRP and CALLATTR are not used for subscription on CAIN/FlexDial interactions.

- enabled triggers (Triggers are enabled through table CAINGRP.)

Note: NetworkBuilder software supports the *Specific_Feature_Code*, *Customized_Dialing_Plan*, *Specific_Digit_String*, and *Office_Code* triggers at *Info_Analyzed*.

Specific_Feature_Code trigger

ATTENTION

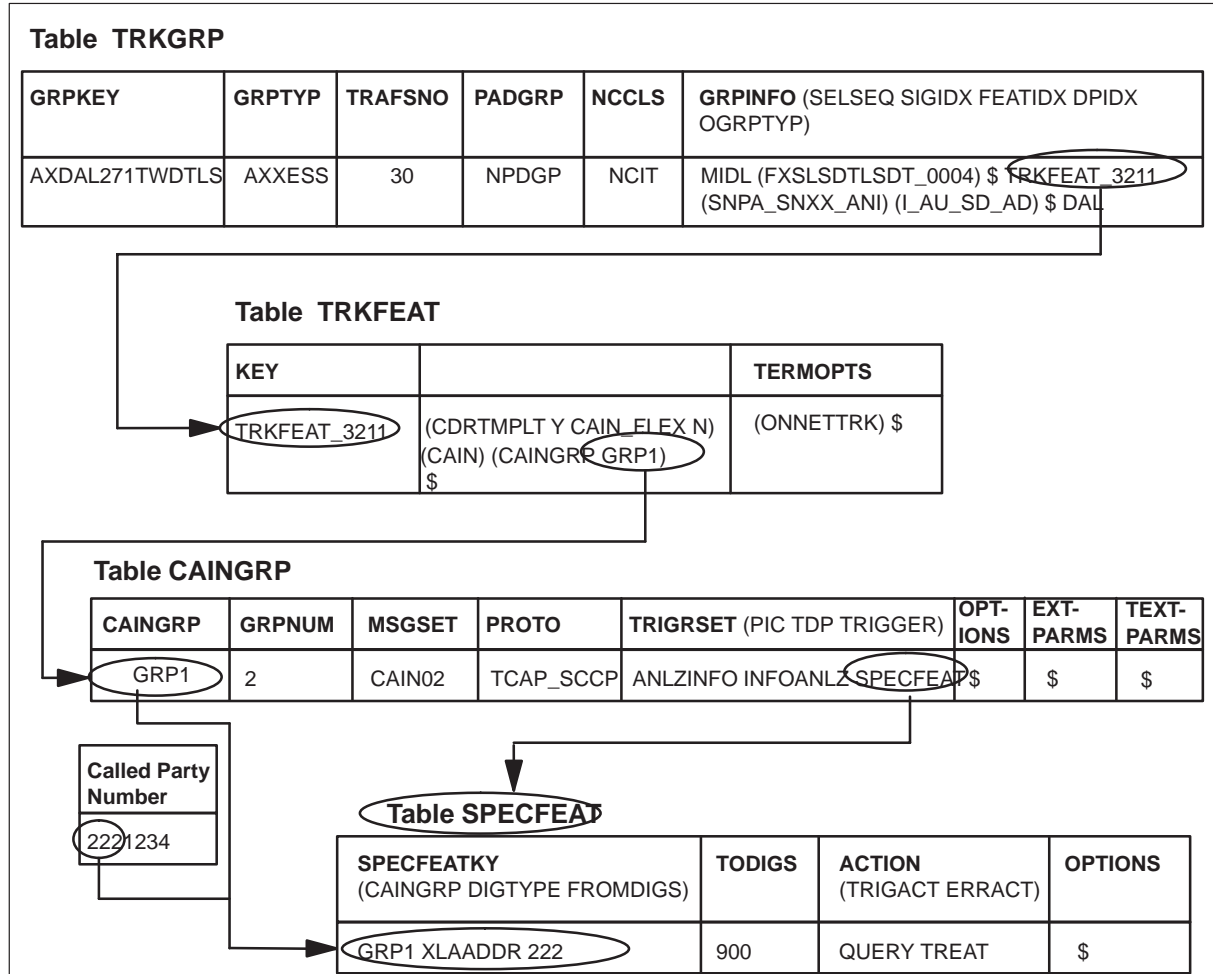
The *Specific_Feature_Code* trigger requires the CAIN0500 SOC option. Refer to the *UCS DMS-250 Software Optionality Control User's Manual* for more information about SOCs.

Datafill

The following figure shows how a subscription table interacts with the *Specific_Feature_Code* trigger table (SPECFEAT).

Step 1: Familiarize yourself with the call model PIC 4: Analyze_Information (continued)

Subscription-SPECFEAT table interaction



Step 1: Familiarize yourself with the call model
PIC 4: Analyze_Information (continued)

Customized_Dialing_Plan trigger

ATTENTION

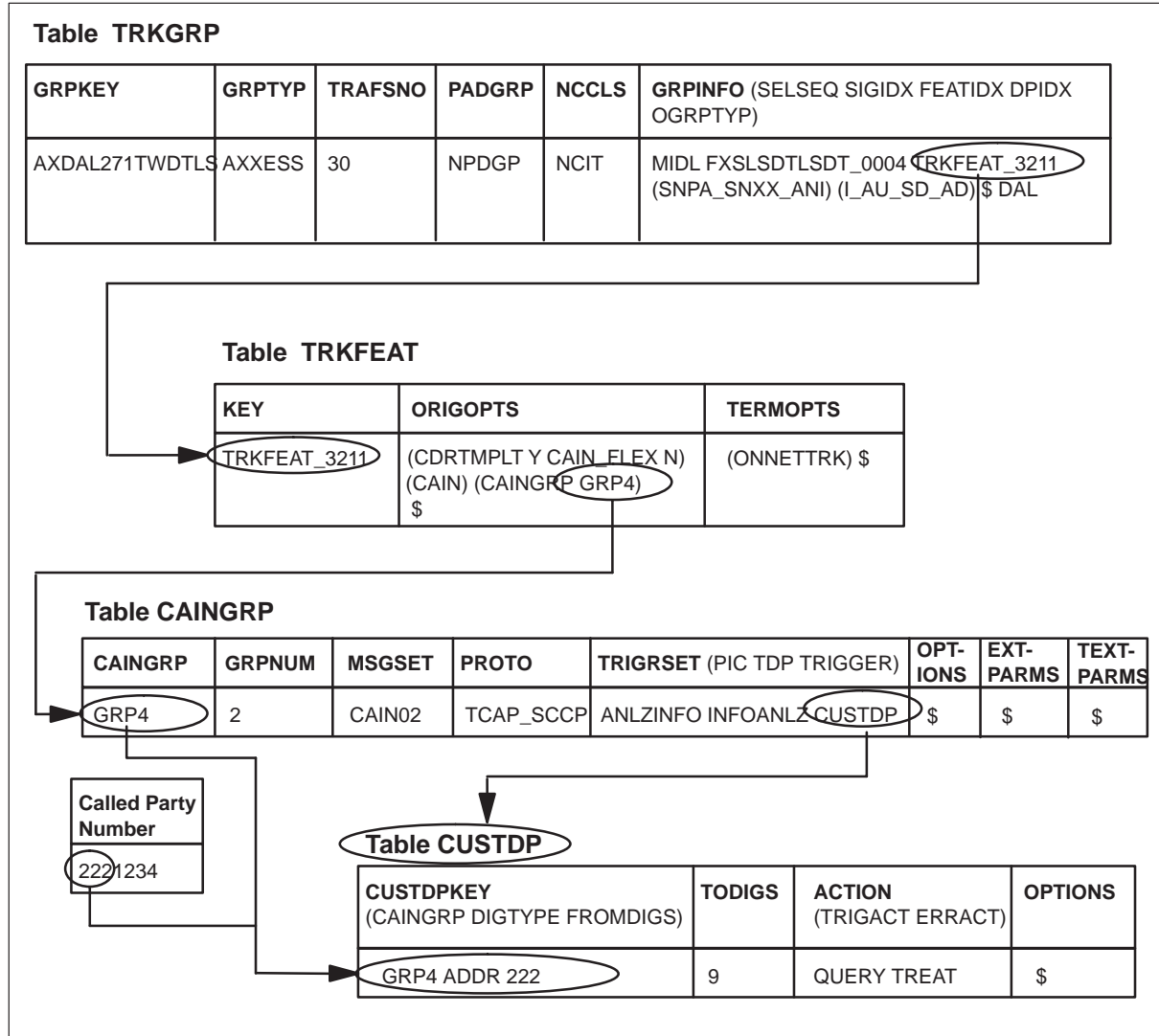
The *Customized_Dialing_Plan* trigger requires the CAIN0500 SOC option. Refer to the *UCS DMS-250 Software Optionality Control User's Manual* for more information about SOCs.

Datafill

The following figure shows how a subscription table interacts with the *Customized_Dialing_Plan* trigger table (CUSTDP).

Step 1: Familiarize yourself with the call model PIC 4: Analyze_Information (continued)

Subscription-CUSTDP table interaction



Note: Refer to the *UCS DMS-250 NetworkBuilder Application Guide* for more information on provisioning the *Customized_Dialing_Plan* trigger and table CUSTDP.

Step 1: Familiarize yourself with the call model
PIC 4: Analyze_Information (continued)

Specific_Digit_String trigger

ATTENTION

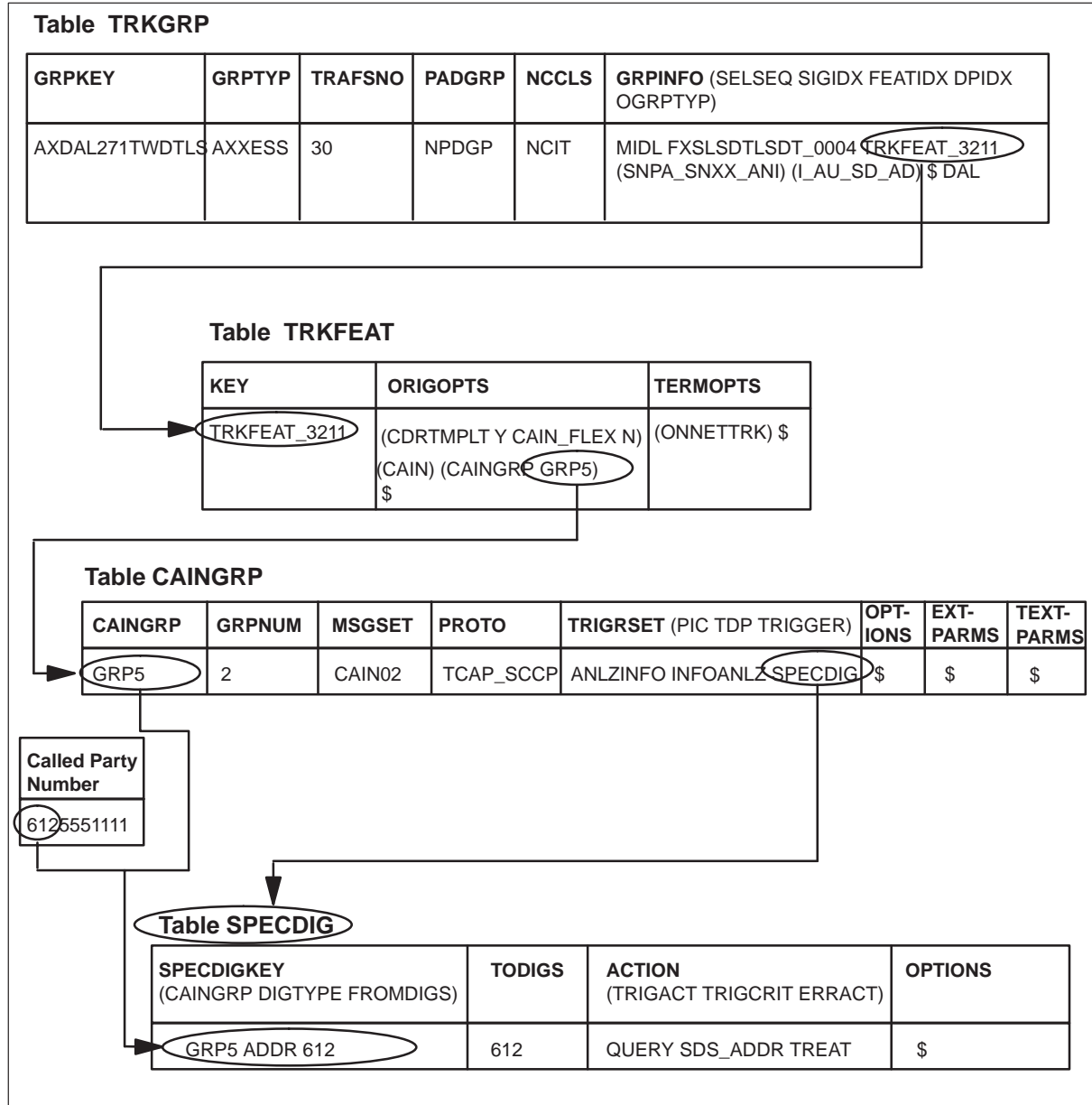
The *Specific_Digit_String* trigger requires the CAIN0501 SOC option. Refer to the *UCS DMS-250 Software Optionality Control User's Manual* for more information about SOCs.

Datafill

The following figure shows how the subscription tables interact with the *Specific_Digit_String* trigger table (SPECDIG).

Step 1: Familiarize yourself with the call model PIC 4: Analyze_Information (continued)

Subscription-SPECDIG table interaction



Note: Refer to the *UCS DMS-250 NetworkBuilder Application Guide* for more information on provisioning the *Specific_Digit_String* trigger and table SPECDIG.

Step 1: Familiarize yourself with the call model
PIC 4: Analyze_Information (continued)

Office_Code trigger

ATTENTION

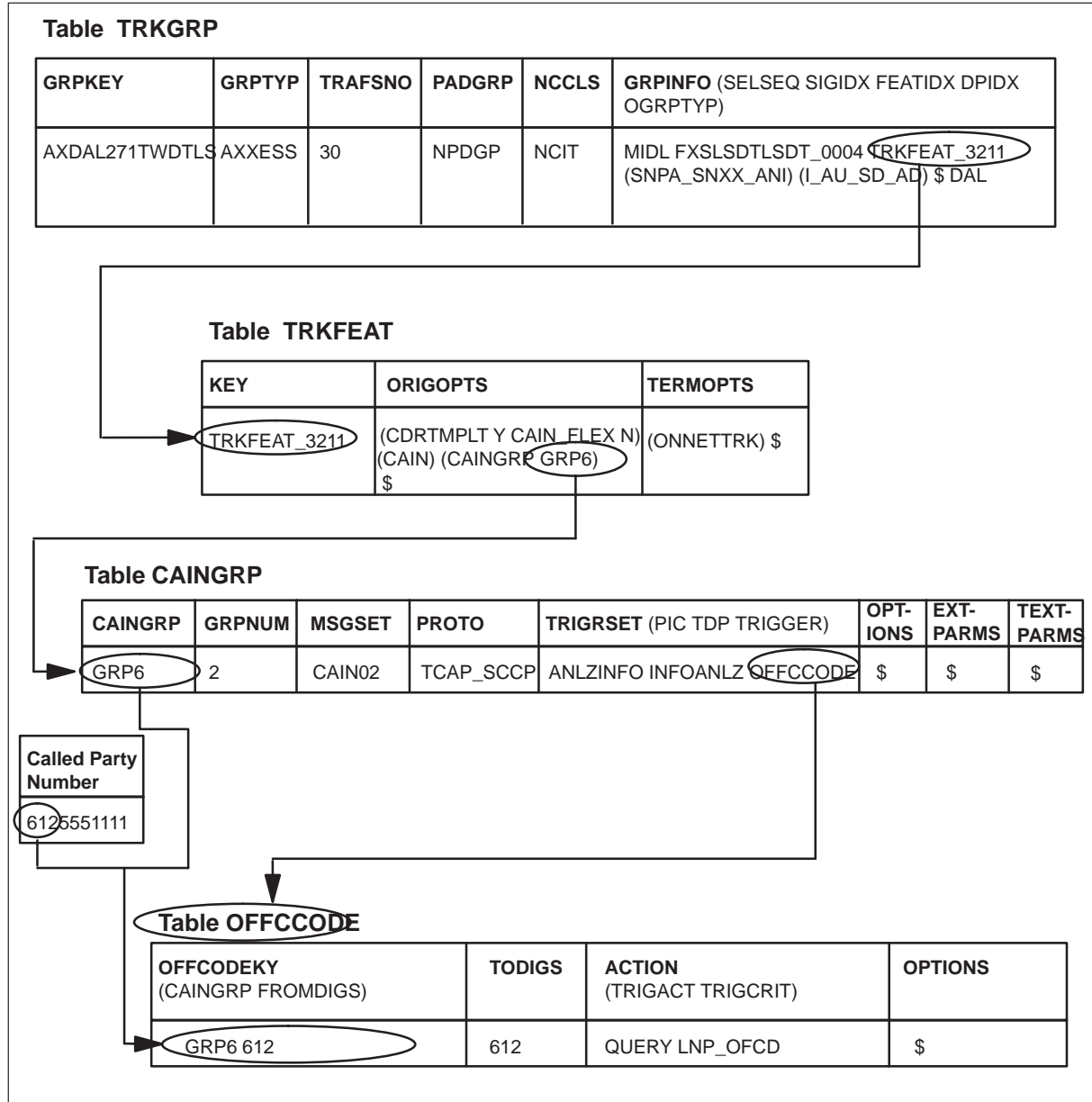
The *Office_Code* trigger requires the CAIN0700 SOC option. Refer to the *UCS DMS-250 Software Optionality Control User's Manual* for more information about SOCs.

Datafill

The following figure shows how the subscription tables interact with the *Office_Code* trigger table (OFFCCODE).

Step 1: Familiarize yourself with the call model
PIC 4: Analyze_Information (end)

Subscription-OFFCCODE table interaction



Note: Refer to the *UCS DMS-250 NetworkBuilder Application Guide* for more information on provisioning the *Office_Code* trigger and table OFFCCODE.

Step 1: Familiarize yourself with the call model

PIC 5: Select_Route

Call processing enters **Select_Route** once a route index is identified.

Network_Busy DP

The **Network_Busy** DP is encountered when the triggering switch has attempted all routes and is unable to terminate the call or when an SS7 release (REL) message or ISDN Release message is received indicating a network busy condition.

When **Network_Busy** is encountered, NetworkBuilder directs the switch to check for the following:

- call subscription to a CAIN group through table STDPRTCT, FLEXTYPE, FLEXFEAT, TRKFEAT, or CAINPARAM

Note: Tables AUTHCODU, ANISCUSP (or ANIVAL or UNIPROF), TRKGRP and CALLATTR are not used for subscription on CAIN/FlexDial interactions.

- enabled triggers (Triggers are enabled through table CAINGRP.)

Note: NetworkBuilder software supports the **Network_Busy** trigger at **Network_Busy**.

Network_Busy trigger/event

ATTENTION

The **Network_Busy** trigger requires the CAIN0506 SOC option. The **Network_Busy** event requires the CAIN0602 SOC option. Refer to the *UCS DMS-250 Software Optionality Control User's Manual* for more information about SOCs.

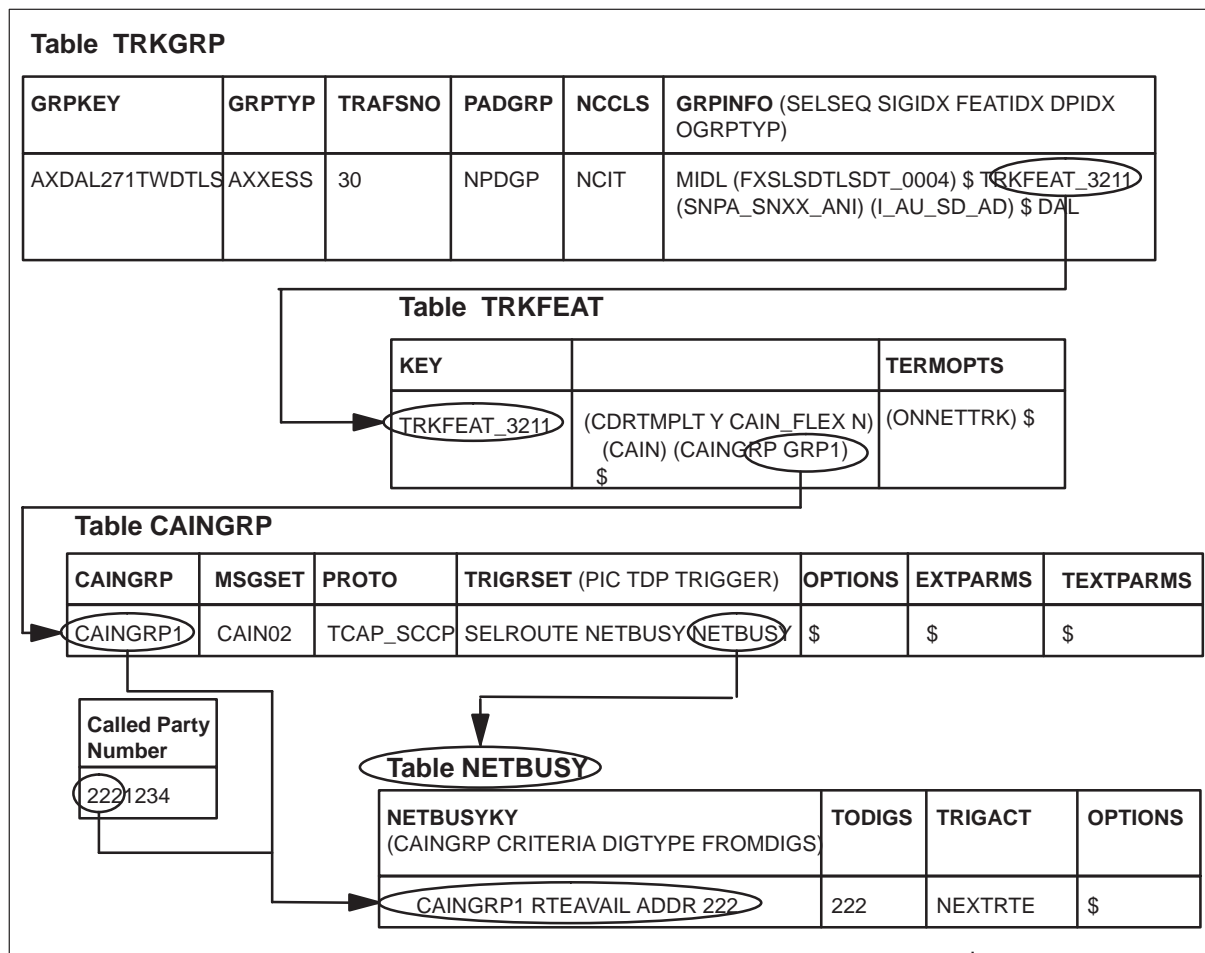
Network_Busy TDP- and EDP-Request

Datafill

The following figure shows how a subscription table interacts with the **Network_Busy** trigger table (NETBUSY).

Step 1: Familiarize yourself with the call model PIC 5: Select_Route (end)

Subscription-NETBUSY table interaction



Note: Refer to the *UCS DMS-250 NetworkBuilder Application Guide* for more information on provisioning the *Network_Busy* trigger and table NETBUSY.

Step 1: Familiarize yourself with the call model

PIC 7: Send_Call

Call processing enters **Send_Call** when the switch attempts to terminate the call.

O_Called_Party_Busy DP

O_Called_Party_Busy is encountered when the switch

- attempts to route to a busy AXXESS agent (with MONNETTRK=Y in table TRKFEAT)
- receives an indication (SS7 release [REL] or ISDN Release message) from the terminating agent that the end-user is busy

When **O_Called_Party_Busy** is encountered, NetworkBuilder directs the switch to check for the following:

- call subscription to a CAIN group through table STDPRTCT, FLEXTYPE, FLEXFEAT, TRKFEAT, or CAINPARM

Note: Tables AUTHCODU, ANISCUSP (or ANIVAL and UNIPROF), TRKGRP and CALLATTR are not used for subscription on CAIN/FlexDial interactions.

- enabled triggers/events (Triggers are enabled through table CAINGRP.)

Note: NetworkBuilder software supports the **O_Called_Party_Busy** trigger at **O_Called_Party_Busy**.

O_Called_Party_Busy trigger/event

ATTENTION

The **O_Called_Party_Busy** trigger requires the CAIN0507 SOC option.
The **O_Called_Party_Busy** event requires the CAIN0602 SOC option.
Refer to the *UCS DMS-250 Software Optionality Control User's Manual* for more information about SOCs.

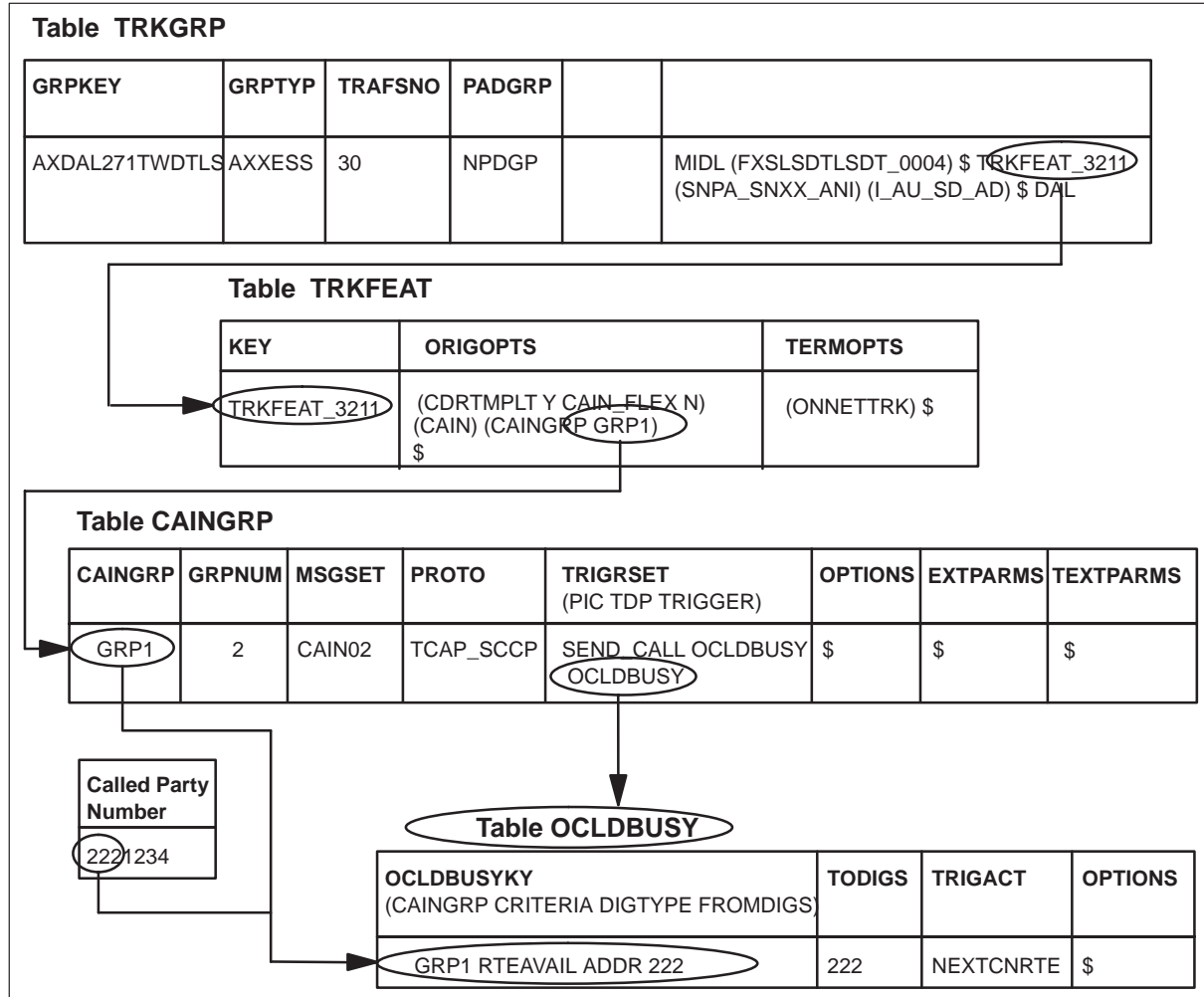
Datafill

The following figure shows how a subscription table interacts with the **O_Called_Party_Busy** trigger table (OCLDBUSY).

Step 1: Familiarize yourself with the call model

PIC 7: Send_Call (end)

Subscription-OCLDBUSY table interaction



Note: Refer to the *UCS DMS-250 NetworkBuilder Application Guide* for more information on provisioning the *O_Called_Party_Busy* trigger and table OCLDBUSY.

Step 1: Familiarize yourself with the call model

PIC 8: O_Alerting

Call processing enters **O_Alerting** when the terminating switch applies ringing and ring-back.

O_No_Answer DP

O_No_Answer is encountered when the called party does not answer within the provisioned amount of time.

When **O_No_Answer** is encountered as a TDP, NetworkBuilder directs the switch to check for the following:

- call subscription to a CAIN group through an SCP-determined group or through table STDPRTCT, authcode tables, ANI tables, TRKGRP, CALLATTR, or CAINPARM
- enabled triggers (Triggers are enabled through table CAINGRP.)

Note: NetworkBuilder software supports the *O_No_Answer* trigger at the *O_No_Answer* TDP.

Long call duration

The *O_No_Answer* trigger interacts with the long call timer as follows:

- If the long call timer expires before the *O_No_Answer* timer, the call is sent to treatment or FlexDial performs the action provisioned in the NOANSDUR option (table FLEXFEAT) and the *O_No_Answer* timer is cancelled.
- If the *O_No_Answer* timer expires before the long call timer, **O_No_Answer** is encountered, and call processing checks the trigger criteria. If the criteria is met and you have provisioned an action of QUERY, NEXTRTE, or NEXTCNRTE, the long call timer is canceled. Otherwise, the long call timer continues.

If you have provisioned an action of QUERY and the SCP returns a **Continue** response message, call processing restarts the long call timer using the datafilled value.

Step 1: Familiarize yourself with the call model

PIC 8: O_Alerting (continued)

O_No_Answer trigger/event

ATTENTION

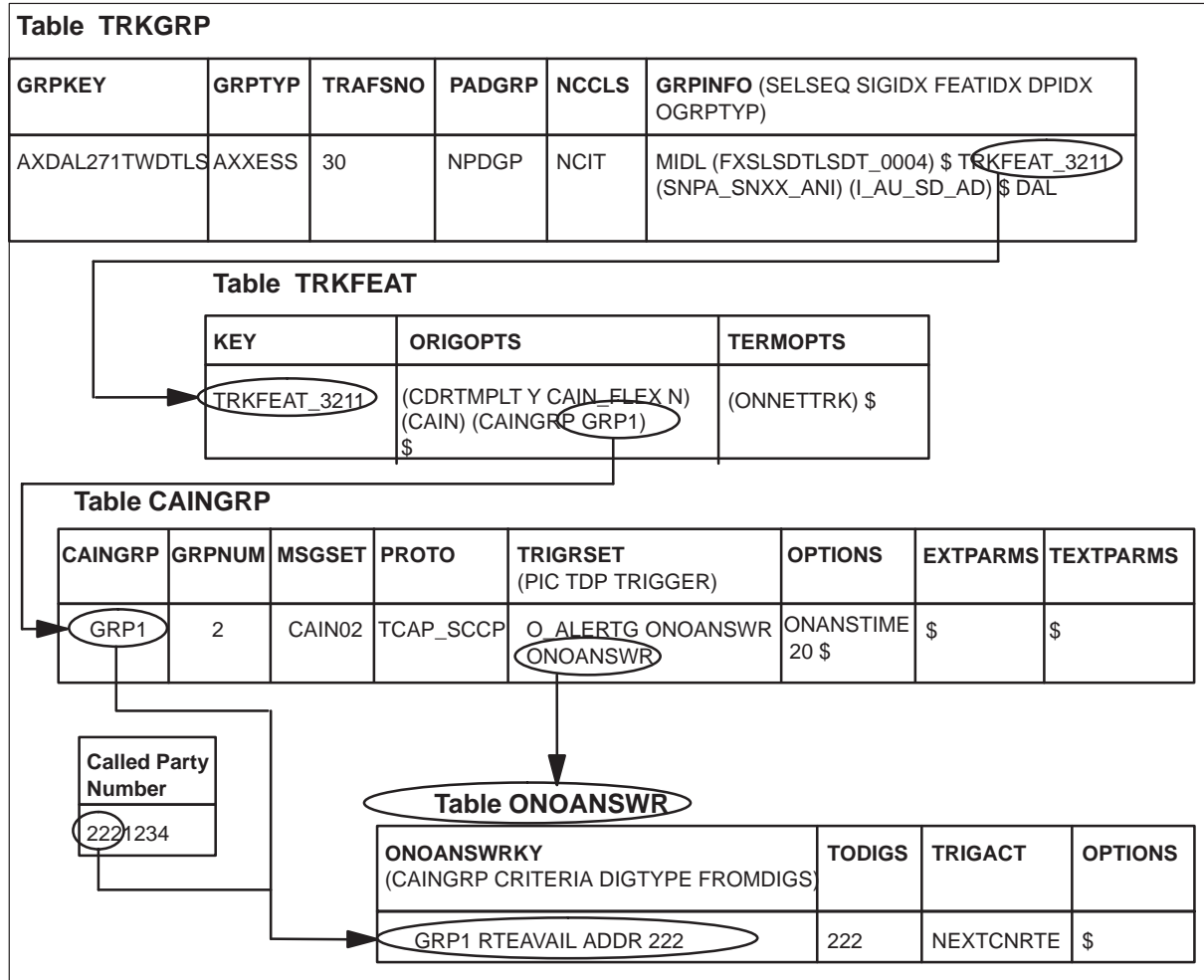
The *O_No_Answer* trigger requires the CAIN0508 SOC option. The *O_No_Answer* event requires the CAIN0602 SOC option. Refer to the *UCS DMS-250 Software Optionality Control User's Manual* for more information about SOCs.

Datafill

The following figure shows how a subscription table interacts with the *O_No_Answer* trigger table (ONOANSWR).

Step 1: Familiarize yourself with the call model
PIC 8: O_Alerting (end)

Subscription-ONOANSWR table interaction



Note: Refer to the *UCS DMS-250 NetworkBuilder Application Guide* for more information on provisioning the *O_No_Answer* trigger and table ONOANSWR.

Step 2: Familiarize yourself with related OA&M

In order to fully understand, implement, and maintain the interactions between CAIN and FlexDial, you need to be familiar with the logs, operational measurements, data schema, and treatments used to process CAIN/FlexDial calls.

This section provides a brief overview of the following:

- Treatment – the method by which a call is disconnected or ended. Refer to *UCS DMS-250 Data Schema Reference Manual*, tables TMTMAP and CSEMAP for a full list of treatment codes.
- Billing – billing records for calls on the switch. Refer to *UCS DMS-250 Billing Records Application Guide* for more billing information.

For information on logs, refer to

- *UCS DMS-250 NetworkBuilder Application Guide* for information on NetworkBuilder logs
- *UCS DMS-250 FlexDial Application Guide* for FlexDial logs
- *UCS DMS-250 Logs Reference Manual*.

For information on operational measurements, refer to

- *UCS DMS-250 NetworkBuilder Application Guide* for information on NetworkBuilder OMs
- *UCS DMS-250 FlexDial Application Guide* for FlexDial OMs
- *UCS DMS-250 Operational Measurements Reference Manual*.

For information on data schema, refer to

- *UCS DMS-250 NetworkBuilder Application Guide* or *UCS DMS-250 Data Schema Reference Manual*. for information on NetworkBuilder data schema
- *UCS DMS-250 FlexDial Application Guide* for FlexDial data schema

Step 2: Familiarize yourself with related OA&M Treatments

Normal call processing

When the switch performs digit collection during normal call processing, errors may occur that cause the switch to set certain treatments (such as PDIL) against the call. When treatment is set, the switch generates the appropriate TRK, OCC, or FLEX302 log and applies the treatment to the call.

NetworkBuilder call processing

When a digit collection error occurs (directed by the switch, NetworkBuilder, or FlexDial), the switch may set a delayed (validation failures) or an immediate (resource problems) treatment. However, when treatment (associated with digit collection) is set on a call occurring during **Collect Information** or **Analyze Information** and the associated trigger criteria is met, NetworkBuilder call processing allows querying of the SCP.

When the trigger action is QUERY, NetworkBuilder suppresses the generation of the treatment logs (TRK, OCC, or FLEX302) because the SCP may collect information and make decisions that allow the call to proceed.

When the trigger action is BLOCK, the switch applies AINF treatment and terminates the call. When the trigger action is IGNORE, CONT_NOTRIG, or LEAVE_TDP, the switch applies the treatment set before the switch evaluated the trigger criteria.

When the trigger criteria is not met, the appropriate logs (TRK, OCC, or FLEX302) are generated and treatment is applied to the call.

FlexDial and treatments

During FlexDial processing the switch may set delayed treatments for failed validation of FlexDial digit collectables. Refer to the *UCS DMS-250 FlexDial Framework Application Guide* for more information.

Note: Refer to the *UCS DMS-250 Data Schema Reference Manual* for more information on treatments.

The FlexDial framework controls the treatment application by tagging the treatment as immediate or delayed.

Immediate treatment indicates software exception conditions. The switch sets the appropriate treatment such as PSIG, PDIL, NOSR, FNAL, and AINF.

Step 2: Familiarize yourself with related OA&M Treatments (end)

Note: When the switch sets AINF due to a BLOCK action at *O_Feature_Requested*, FlexDial reacts as if immediate treatment was set.

Delayed treatment indicates validation failures associated with the subscriber number, such as multi-COS screening and AINF, with account validation failures through tables FLEXVAL and FLEXFEAT. The treatment is typically provisioned in FlexDial datafill. The FlexDial framework continues to process the call.

Refer to *UCS DMS-250 Data Schema Reference Manual*, tables TMTMAP and CSEMAP for a full list of treatment codes.

Step 2: Familiarize yourself with related OA&M Billing



CAUTION

Changes may affect site functionality

Changes to the billing system require updates to engineering parameters that may affect site functionality. Any changes to the billing system may affect downstream processing of billing records. Nortel recommends that only experienced personnel make changes to the billing system.

NetworkBuilder/FlexDial population of CDR fields

Table 2-1 shows how CAIN/FlexDial interactions affect the population of CDR fields.

Table 2-1
CDR population

Field	NetworkBuilder population	FlexDial collectable
ACCTCD	<i>AMADigitsDialedWC</i> in an <i>Analyze_Route</i> , <i>Continue</i> , or <i>Disconnect</i>	SUBR or SUBRPARM with BILLFLD ACCTCT option (table FLEXFEAT)
ANISP	<i>ChargeNumber</i> in an <i>Analyze_Route</i> , when the NOA is I2ANI or ANI	SUBR or SUBRPARM with BILLFLD ANISP option (table FLEXTYPE)
Note: Refer to the <i>UCS DMS-250 NetworkBuilder Application Guide</i> and the <i>UCS DMS-250 Billing Records Application Guide</i> for more information.		
—continued—		

Step 2: Familiarize yourself with related OA&M Billing (continued)

Table 2-1
CDR population (continued)

Field	NetworkBuilder population	FlexDial collectable
BILLNUM	<ol style="list-style-type: none"> 1 <i>AMADigitsDialedWC</i> in an <i>Analyze_Route</i>, <i>Continue</i>, or <i>Disconnect</i> 2 <i>ChargeNumber</i> in an <i>Analyze_Route</i>, when the NOA is N00, MCCS, or AUTH 	<ol style="list-style-type: none"> 1 SUBR or SUBRPARM with BILLFLD BILLNUM option (table FLEXFEAT) 2 Validation of subscriber number with CLDPBILL option (table FLEXFEAT). Contains the called address. 3 ADDR or ADDRPARM flagged as called party billed
CALLEDNO	<i>CalledPartyID</i> in an <i>Analyze_Route</i>	ADDR or ADDRPARM upon validation
CIC	<i>AMADigitsDialedWC</i> in an <i>Analyze_Route</i> , <i>Continue</i> , or <i>Disconnect</i>	CIC, CICPARM, SUBR, SUBRPARM with BILLFLD CIC option (table FLEXTYPE)
CLGPTYNO	<i>CallingPartyID</i> in an <i>Analyze_Route</i>	SUBR or SUBRPARM with BILLFLD CLGPTYNO option (table FLEXTYPE)
INC_INTL	NetworkBuilder call processing does not populate this field.	SUBR or SUBRPARM with BILLFLD CLGPTYNO option (table FLEXTYPE)
OPART	<i>servTranslationScheme</i> extension parameter in an <i>Analyze_Route</i>	SUBR or SUBRPARM
OUTPUTLNO	<ol style="list-style-type: none"> 1 <i>OutputpulseNumber</i> in an <i>Analyze_Route</i> 2 <i>AlternateOutputpulseNo</i> in the <i>GenericAddressList</i> in an <i>Analyze_Route</i> 	ADDR or ADDRPARM
<p>Note: Refer to the <i>UCS DMS-250 NetworkBuilder Application Guide</i> and the <i>UCS DMS-250 Billing Records Application Guide</i> for more information.</p>		
—continued—		

Step 2: Familiarize yourself with related OA&M Billing (end)

Table 2-1
CDR population (continued)

Field	NetworkBuilder population	FlexDial collectable
PINDIGS	<i>AMADigitsDialedWC</i> in an <i>Analyze_Route</i> , <i>Continue</i> , or <i>Disconnect</i>	SUBR or SUBRPARM with BILLFLD PINDIGS option (table FLEXFEAT)
PORTEDNO	<i>CalledPartyID</i> in an <i>Analyze_Route</i> for LNP queries only	The FlexDial Framework does not populate this field.
PRESIND	<i>CallingPartyID</i> in an <i>Analyze_Route</i>	SUBR or SUBRPARM with BILLFLD CLGPTYNO option (table FLEXTYPE)
TPART	<i>servTranslationScheme</i> extension parameter in an <i>Analyze_Route</i>	SUBR or SUBRPARM
<p>Note: Refer to the <i>UCS DMS-250 NetworkBuilder Application Guide</i> and the <i>UCS DMS-250 Billing Records Application Guide</i> for more information.</p>		
—end—		

Step 3: Be familiar with agent setup messaging

To better understand NetworkBuilder/FlexDial call processing, you need to be familiar with the setup messaging for the AXXESS agent.

AXXESS originations and terminations

The following table lists the supported triggers on AXXESS agents:

Supported triggers

Trigger	Supported signaling			Mimicked agents (Note 1)						
	DS1	FXS/FXO	CCS7	DAL	EDAL	DAL TIE	ONAL (2-wire)	ONAL (4-wire)	ONAT	FGD (PTS and SS7)
<i>Off_Hook_Immediate</i>		√		√	√	-	√	-	-	-
<i>O_Feature_Requested</i>	√	√	√	√	√	√	√	√	√	√
<i>Tollfree_Service</i>	√	√	√	√	√	√	√	√	√	√
<i>Off_Hook_Delay</i>	√	√	√	√	√	√	√	√	√	√
<i>Shared_Interoffice_Trunk</i>	√	√	√	√	√	√	√	√	√	√
<i>PRI_B-Channel</i>	√	√	√	-	-	-	-	-	-	-
<i>Specific_Feature_Code</i>	√	√	√	√	√	√	√	√	√	√
<i>Customized_Dialing_Plan</i>	√	√	√	√	√	√	√	√	√	√
<i>Specific_Digit_String</i>	√	√	√	√	√	√	√	√	√	√
<i>Office_Code</i> (Note 2)	√	√	√	√	√	√	√	√	√	√
<i>Network_Busy</i>	√	√	√	√	√	√	√	√	√	√

Note: Not available for Global-IMT, PTS IMT, and SS7 IMT originations.

—continued—

Step 3: Be familiar with agent setup messaging (continued)**Supported triggers** (continued)

Trigger	Supported signaling			Mimicked agents (Note 1)						
	DS1	FXS/FXO	CCS7	DAL	EDAL	DAL TIE	ONAL (2-wire)	ONAL (4-wire)	ONAT	FGD (PTS and SS7)
<i>O_Called_Party_Busy</i>	√	√	√	√	√	√	√	√	√	√
<i>O_No_Answer</i>	√	√	√	√	√	√	√	√	√	√
<i>Termination_Attempt</i>	√	√	√	√	√	√	√	√	√	√
Note: Not available for Global-IMT, PTS IMT, and SS7 IMT originations.										
—end—										

NetworkBuilder software does not support non-AXXESS EDALs, ONALs, or ONATs. However, by using the FlexDial AXXESS agent, you can define agents that mimic EDAL, ONAL, or ONAT protocols. Since NetworkBuilder software supports the AXXESS agent, AXXESS agents that mimic EDALs, ONALs, or ONATs are supported.

The following data may be outputted to the switch through PTS AXXESS originations:

- called party number
- ANI
- information digits
- CIC

Refer to the *UCS DMS-250 FlexDial Framework Application Guide* for more information on AXXESS originations and terminations.

Step 3: Be familiar with agent setup messaging (end)

Step 4: Define CCS7 connectivity

Refer to the *UCS DMS-250 NetworkBuilder Application Guide* for details on defining CCS7 connectivity.

Step 5: Define resource allocation requirements

Refer to the *UCS DMS-250 NetworkBuilder Application Guide* for details on defining resource allocation requirements.

Step 6: Datafill the AXXESS agent as CAIN-capable

Before any NetworkBuilder/FlexDial processing takes place, the originating or terminating AXXESS agent must be provisioned as CAIN-capable.

Note: Refer to the *UCS DMS-250 NetworkBuilder Application Guide* for non-AXXESS agent provisioning information.

ATTENTION

When the CAIN option is not provisioned for the originating AXXESS agent, or CAIN option is not provisioned for the terminating AXXESS agent, the call is not evaluated for the corresponding NetworkBuilder services.

Note: AXXESS agent signaling is provisioned in table TRKSIG. Table TRKGRP provides the index into table TRKSIG.

The following triggers are supported on originating AXXESS agents:

- *Off_Hook_Immediate*

Note: The originating AXXESS agent must be provisioned with FXS- or FXO-type signaling, similar to legacy two-wire DAL agents.

- *O_Feature_Requested*
- *Tollfree_Service*
- *Off_Hook_Delay*

Note: The originating AXXESS agent must be provisioned with FXS- or FXO-type signaling, similar to legacy two-wire DAL agents.

- *Shared_Interoffice_Trunk*
- *Specific_Feature_Code*
- *Customized_Dialing_Plan*
- *Specific_Digit_String*
- *Office_Code*
- *Network_Busy*
- *O_Called_Party_Busy*
- *O_No_Answer*
- *O_IEC_Reorigination*
- *Termination_Attempt*

Step 6: Datafill the AXXESS agent as CAIN-capable (continued)

Note: The *PRI_B-Channel* trigger is not supported on AXXESS agents.

AXXESS datafill requirements

Before any NetworkBuilder services can be used, you must provision the feature set for the originating or terminating AXXESS agent as CAIN-capable using the CAIN option in table TRKFEAT (Trunk Feature).

Note: The index to table TRKFEAT is provisioned in table TRKGRP.

TRKFEAT

AXXESS agents are datafilled in table TRKGRP which provides an index into table TRKFEAT. Datafill in table TRKFEAT identifies the activated feature set for the agent on an originating (ORIGOPTS) and terminating (TERMOPTS) basis. Make the agent CAIN-capable by datafilling the CAIN option in the ORIGOPTS option field or TCAIN option in the TERMOPTS option field of table TRKFEAT.

Step 6: Datafill the required agents as CAIN-capable originating AXCESS agents (end)

At the CI prompt

- 1 Enter table TRKFEAT.
- 2 Position on the feature set requiring originating CAIN capabilities.
- 3 Change the tuple to reflect originating call model CAIN capability by adding the CAIN option by using the following format:

>CHA key origopts termopts

where

key is the name of the feature set (up to 16 alphanumeric characters) indexed by table TRKGRP.

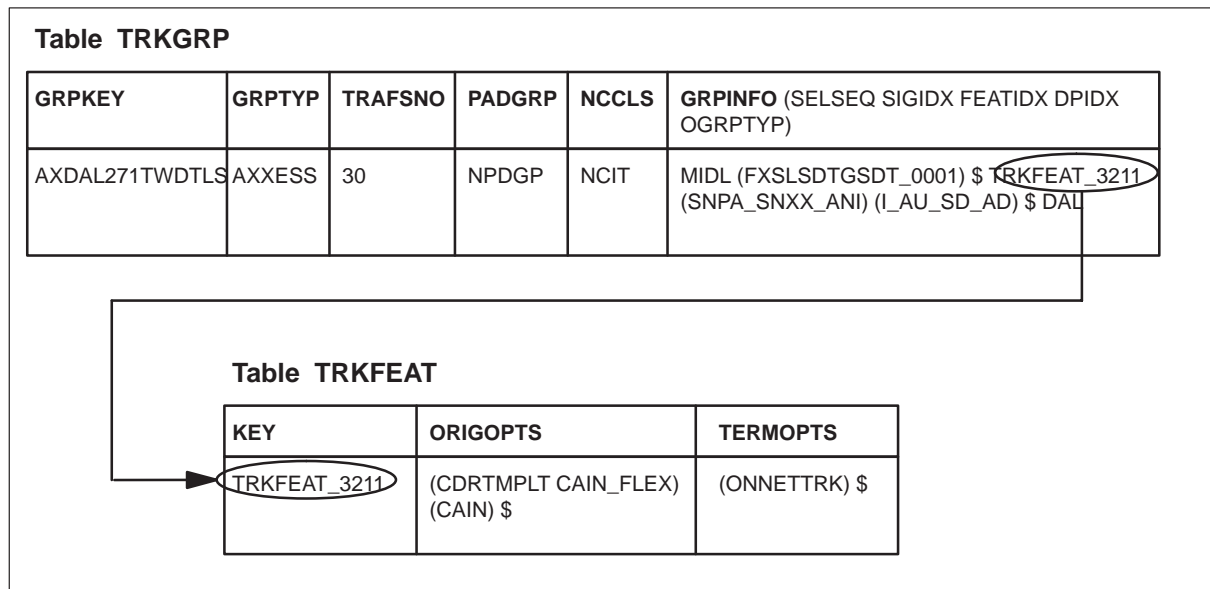
origopts is the option you want to enable on the originating agent (CAIN).

termopts is the option you want to enable on the terminating agent.

Sample entry: **CHA trkfeat_2323 cdrtmpl cain_flex CAIN \$ onnettrk \$**

The agent is now capable of using NetworkBuilder call processing.

Relationship between TRKGRP and TRKFEAT



Step 8: Choose the type of subscription

Once you've identified the TDPs that require specific services provided by the SCP, you need to determine the calls that require the service.

Any call originating or terminating on a supported, provisioned agency can subscribe to NetworkBuilder services.

Subscription methods

As data is collected for a call, subscription methods are stored for later use by NetworkBuilder call processing. For example, when a call goes off-hook, the following occurs:

- 1 NetworkBuilder checks table CAINPARM for subscription to a CAIN group (using the CAIN_OFFICE_GROUP parameter in table CAINPARM) and stores the group for use later in the call. If the call subscribes to NetworkBuilder services, the switch generates a CAIN902 log indicating the subscription method.
- 2 NetworkBuilder checks the originating agent (through table TRKFEAT) for CAIN-capability and subscription to a CAIN group (using the CAINGRP option). If the call subscribes to NetworkBuilder services, the switch generates a CAIN902 log indicating the subscription method.

As the call progresses through the call model and enters **Collect Information**, the switch initiates digit collection as directed by normal call processing, FlexDial call processing, or NetworkBuilder call processing. For example, the following occurs:

- 1 NetworkBuilder checks the collected subscriber data (authorization codes, ANIs, addresses, SUBR collectables, SUBRPARM collectables, ADDR collectables, ADDRPARM collectables, or CALLTYPE collectable) for subscription to a CAIN group and stores the group for use later in the call. If the call subscribes to NetworkBuilder services, the switch generates a CAIN902 log indicating the subscription method.

The switch stores up to six subscription groups, one for each precedence level.

Precedence levels

With the NetworkBuilder (CAIN)/FlexDial interaction, a new method of discussing subscription is introduced. The following table maps the new method to the previous method:

Step 8: Choose the type of subscription (continued)**Subscription methods**

UCS07 and higher	Prior to UCS07
Precedence Level 1	SCP-returned
Precedence Level 2	na
Precedence Level 3	Authorization code
Precedence Level 4	ANI
Precedence Level 5	Agent
Precedence Level 6	Office
—end—	

Precedence Level 1 subscription

This subscription level refers to subscription determined by the SCP. The SCP can return a `cainGroup` extension parameter in an **Analyze_Route**, providing SCP control over subscription. When the SCP returns the `cainGroup` extension parameter, the CAIN group overwrites any previously stored CAIN group associated with Precedence Level 1 subscription.

Note: Precedence Level 1 subscription requires the CAIN601 SOC option. This is the only type of subscription that can change after leaving **Collect_Information**.

Precedence Level 2 subscription

This subscription level refers to subscription based on the called party's address. The switch performs one of the following to obtain address digits:

- Normal call processing – address is collected as determined by the originating agent and the associated hard-coded dialing plan. Subscription is determined based on the collected address for the call.
- FlexDial call processing – address is collected as directed by datafill in table FLEXDIAL. The ADDR and ADDRPARM collectables are used to determine Precedence Level 2 subscription.

Note: In some cases, the FlexDial Framework may collect multiple addresses. The last collected address digits are always used to determine this NetworkBuilder subscription method.

- NetworkBuilder call processing – address is collected as directed by datafill in table OFTRREQ. When the PRTNM option is provisioned in table OFTRREQ, the `o_Feature_Requested` ADDR collectible is used to determine Precedence Level 2 subscription from table STDPRTCT.

Step 8: Choose the type of subscription (continued)

Note: In some cases, NetworkBuilder may collect an address after the FlexDial Framework has collected an address. In this case, the NetworkBuilder-collected address is used to determine the subscription method. The last collected address digits are always used to determine the NetworkBuilder subscription method.

Provision Precedence Level 2 subscription against the appropriate selector (ES, UA, UAX, CT, and NT) in subtable STDPRT of table STDPRTCT.

Note: AXXESS originations do not support the UA pretranslator.

Precedence Level 3 subscription

This subscription level refers to subscription based on one of the following:

- authorization codes – subscription is provisioned in one of the authcode tables (AUTHCODU, AUTHCDU2, AUTHCDU3, AUTHCDU4, or AUTHCDU5).
- SUBR or SUBRPARAM collectable that indexes a FLEXTYPE tuple where the CAINFLG option is set and a FLEXFEAT tuple where the CAINGRP option is set.

Note 1: If the CAINGRP option is provisioned in table FLEXFEAT, subscription is updated; if the CAINGRP option is not provisioned, subscription is cancelled if the CAINFLG is the same as the previously defined subscription method.

Note 2: The FlexDial Framework performs digit validation as the digits are collected. Whenever split validation is performed, the last CAIN group processed during split validation is used to determine the particular collectables impact on subscription.

- CALLTYPE collectable – subscription is provisioned in table FLEXFEAT.

Note: Whenever the FlexDial Framework executes SUBR, SUBRPARAM, and CALLTYPE collectables, the last collectable affecting a Precedence Level 3 subscription is used for subscription to NetworkBuilder services.

Precedence Level 4 subscription

This subscription level refers to subscription based on one of the following:

- ANI – subscription is provisioned in table ANISCUSP (or ANIVAL and UNIPROF).

Step 8: Choose the type of subscription (continued)

- SUBR or SUBRPARM collectables that indexed a FLEXTYPE tuple where the CALLING option is set and the CAINFLG option is not. The CAINGRP option is required on the associated FLEXFEAT tuple.

Note 1: If the CAINGRP option is provisioned in table FLEXFEAT, subscription is updated; if the CAINGRP option is not provisioned, subscription is canceled (if previously established).

Note 2: The CAINFLG option takes precedence over the CALLING option when both are present in a FLEXTYPE tuple.

Note 3: The FlexDial Framework performs digit validation as the digits are collected. Whenever split validation is performed, the last CAIN group processed during split validation is used to determine the particular collectables impact on subscription.

Precedence Level 5 subscription

This subscription level refers to subscription based on the originating agent. Subscription based on an AXXESS agent is provisioned in table TRKFEAT (field ORIGOPTS or TERMOPTS); non-AXXESS agents (except PRI) are provisioned in table TRKGRP; PRI agents are provisioned in table CALLATTR.

Precedence Level 6 subscription

This subscription level refers to default office subscription. Office-wide subscription is provisioned in table CAINPARM.

The following table shows the order that triggers are evaluated when a CAIN group is stored for each subscription method.

Step 8: Choose the type of subscription (continued)

Trigger evaluation order

TDP	Trigger	Table	Subscription method					
			Precedence Level 1	Precedence Level 2	Precedence Level 3	Precedence Level 4	Precedence Level 5	Precedence Level 6
<i>Origination_Attempt</i>	<i>Off_Hook_Immediate</i>	OFFHKIMM	na	na	na	na	1	2
<i>O_Feature_Requested</i>	<i>O_Feature_Requested</i>	OFTRREQ	na	1	2	3	4	5
<i>Info_Collected</i>	<i>Tollfree_Service</i> (note 3)	TOLLFREE	1 (note 2)	2	3	4	5	6
		OFFHKDEL	7 (note 2)	8	9	10	11	12
	<i>Offhook_Delay</i>	SIOTRK	13	14	15	16	17	18
	<i>Shared_Interoffice_Trunk</i>		(note 2)					
	<i>PRI_B-Channel</i> (note 4)	PRIBCHNL	13 (note 2)	14	15	16	17	18
<i>Info_Analyzed</i>	<i>Specific_Feature_Code</i>	SPECFEAT	1 (note 2)	5	9	13	17	21
	<i>Customized_Dialing_Plan</i>	CUSTDP	2 (note 2)	6	10	14	18	22
	<i>Specific_Digit_String</i>	SPECDIG	3 (note 2)	7	11	15	19	23
	<i>Office_Code</i>	OFFCCODE	4 (note 2)	8	12	16	20	24
<p>Note 1: Only AXCESS originations allow re-evaluation of collectable-based CAIN group subscription upon reorigination; also applies to non-AXCESS agents in the case of address digits.</p> <p>Note 2: An SCP-returned CAIN group, specified before reorigination, is available at these TDPs upon reorigination.</p> <p>Note 3: NetworkBuilder supports Bellcore's <i>TR-NWT-000533</i> toll-free service specifications. The ability to subscribe to multiple CAIN groups allows for a service integration of the CAIN triggers and the IN/1 <i>Tollfree_Service</i> trigger defined in <i>TR-NWT-000533</i>.</p> <p>Note 4: The <i>PRI_B-Channel</i> trigger is not supported on AXCESS calls.</p>								
—continued—								

Step 8: Choose the type of subscription (continued)**Trigger evaluation order** (continued)

TDP	Trigger	Table	Subscription method					
			Precedence Level 1	Precedence Level 2	Precedence Level 3	Precedence Level 4	Precedence Level 5	Precedence Level 6
<i>Network_Busy</i>	<i>Network_Busy</i>	NETBUSY	1	2	3	4	5	6
<i>O_Called_Party_Busy</i>	<i>O_Called_Party_Busy</i>	OCLDBUSY	1	2	3	4	5	6
<i>O_No_Answer</i>	<i>O_No_Answer</i>	ONOANSWR	1	2	3	4	5	6
<i>O_Mid_Call</i>	<i>O_IEC_Reorigination</i>	OIECREO	1	2	3	4	5	6
<i>T_Null</i>	<i>Termination_Attempt</i>	TERMATT	na	na	na	na	1	na

Note 1: Only AXCESS originations allow re-evaluation of collectable-based CAIN group subscription upon reorigination; also applies to non-AXCESS agents in the case of address digits.
Note 2: An SCP-returned CAIN group, specified before reorigination, is available at these TDPs upon reorigination.
Note 3: NetworkBuilder supports Bellcore's *TR-NWT-000533* toll-free service specifications. The ability to subscribe to multiple CAIN groups allows for a service integration of the CAIN triggers and the IN/1 *Tollfree_Service* trigger defined in *TR-NWT-000533*.
Note 4: The *PRI_B-Channel* trigger is not supported on AXCESS calls.

—end—

Changes to NetworkBuilder subscription

During a call, NetworkBuilder subscription can be changed due to:

- execution of CLRFTRS collectable – when the FlexDial Framework processes a CLRFTRS collectable
- digit validation failures – RESET failure action – during digit validation “reset to” failure action may require that collectable-based subscription (through the CALLTYPE, ADDR, ADDRPARAM, or CLRFTRS collectables) revert back to previous values for SUBR/SUBRPARAM collectables. Refer to the *UCS DMS-250 FlexDial Framework Application Guide* for more information on the “reset to” failure action.

Note: Only Precedence Levels 2 through 4 are affected.

Step 8: Choose the type of subscription (continued)

- multiple collection of same-type digits – due to the flexibility of the FlexDial Framework, same-type digits can be collected multiple times. Therefore, different CAIN groups may be associated with the call at different times.

Note: In addition, at certain times within the call, re-entry scenarios may be defined. Refer to Chapter 3, “Call processing,” for more information on re-entry scenarios.

Digit validation failures

As FlexDial collectables are executed, the switch validates the collected digits. The switch can validate the ADDR and ADDRPARAM collectables through multi-COS screening or table STDPRTCT. The switch validates the OLI and OLIPARM collectables through table STDPRTCT. SUBR and SUBRPARM collectables are validated through tables FLEXFEAT and FLEXVAL.

Note: For more information on FlexDial digit validation, refer to the *UCS DMS-250 FlexDial Framework Application Guide*.

Because subscription is available on the ADDR, ADDRPARAM, SUBR, and SUBRPARM collectables, failed digit validation can have an affect on subscription. When digit validation passes, the CAIN group is recorded and call processing continues. However, depending on the datafill associated with the call, failed validation can change subscription.

The following table shows potential validation failures and corresponding actions for the SUBR and SUBRPARM collectables

Step 8: Choose the type of subscription (continued)

Table 1
Validation failures of SUBR and SUBRPARM collectables

Table FLEXFEAT datafill of the OPTION field	Table FLEXDIAL associated datafill of the OPTION1 refinement for SUBR and SUBRPARM collectables	Switch action
CASUAL	CASUBLK option is not provisioned	<ul style="list-style-type: none"> • CASUAL number validation and screening is not performed • CAIN group is recorded • remaining options datafilled in table FLEXFEAT are applied as necessary
	CASUBLK TRMT N AINF	<ul style="list-style-type: none"> • CASUAL number validation and screening is performed • Validation fails • CAIN group is recorded • remaining options datafilled in table FLEXFEAT are applied as necessary • FlexDial Framework sets delayed treatment
<p>Note: Refer to the <i>UCS DMS-250 FlexDial Framework Application Guide</i> for more information on datafilling tables FLEXFEAT and FLEXDIAL.</p>		
<p>—continued—</p>		

Step 8: Choose the type of subscription (continued)

Table 1
Validation failures of SUBR and SUBRPARM collectables (continued)

Table FLEXFEAT datafill of the OPTION field	Table FLEXDIAL associated datafill of the OPTION1 refinement for SUBR and SUBRPARM collectables	Switch action
CITYCODE	CASUBLK VALIDATE CITYVAL option is not provisioned	<ul style="list-style-type: none"> • CASUAL number validation and screening is performed • Validation fails • CAIN group is recorded • See Table 2 for datafill of the OPTION2 refinement and the associated failure action • City code and CAIN group are recorded • remaining options datafilled in table FLEXFEAT are applied as necessary
<p>Note: Refer to the <i>UCS DMS-250 FlexDial Framework Application Guide</i> for more information on datafilling tables FLEXFEAT and FLEXDIAL.</p>		
<p>—continued—</p>		

Step 8: Choose the type of subscription (continued)

Table 1
Validation failures of SUBR and SUBRPARM collectables (continued)

Table FLEXFEAT datafill of the OPTION field	Table FLEXDIAL associated datafill of the OPTION1 refinement for SUBR and SUBRPARM collectables	Switch action
	CITYVAL DPIDX	<ul style="list-style-type: none"> • For successful validations: <ul style="list-style-type: none"> — CAIN group is recorded — remaining options datafilled in table FLEXFEAT are applied as necessary • For failed validations: <ul style="list-style-type: none"> — CAIN group is recorded — remaining options datafilled in table FLEXFEAT are ignored — Interaction is modified as directed by the DPIDX
<p>Note: Refer to the <i>UCS DMS-250 FlexDial Framework Application Guide</i> for more information on datafilling tables FLEXFEAT and FLEXDIAL.</p>		
<p>—continued—</p>		

Step 8: Choose the type of subscription (continued)

Table 1
Validation failures of SUBR and SUBRPARM collectables (continued)

Table FLEXFEAT datafill of the OPTION field	Table FLEXDIAL associated datafill of the OPTION1 refinement for SUBR and SUBRPARM collectables	Switch action
	CITYVAL TRMT	<ul style="list-style-type: none"> • For successful validations: <ul style="list-style-type: none"> — CAIN group is recorded — remaining options datafilled in table FLEXFEAT are applied as necessary • For failed validations: <ul style="list-style-type: none"> — CAIN group is recorded — remaining options datafilled in table FLEXFEAT are ignored — FlexDial Framework sets delayed treatment
<p>Note: Refer to the <i>UCS DMS-250 FlexDial Framework Application Guide</i> for more information on datafilling tables FLEXFEAT and FLEXDIAL.</p>		
<p>—continued—</p>		

Step 8: Choose the type of subscription (continued)

Table 1
Validation failures of SUBR and SUBRPARM collectables (continued)

Table FLEXFEAT datafill of the OPTION field	Table FLEXDIAL associated datafill of the OPTION1 refinement for SUBR and SUBRPARM collectables	Switch action
FAILVAL TRMT	<p>CITYVAL VALIDATE</p> <p>not applicable</p>	<ul style="list-style-type: none"> • For successful validations: <ul style="list-style-type: none"> — CAIN group is recorded — remaining options datafilled in table FLEXFEAT are applied as necessary • For failed validations: <ul style="list-style-type: none"> — CAIN group is recorded — See Table 2 for datafill of the OPTION2 refinement and the associated failure action • For failed validations: <ul style="list-style-type: none"> — CAIN group is recorded — remaining options datafilled in table FLEXFEAT are ignored — FlexDial Framework sets delayed treatment
<p>Note: Refer to the <i>UCS DMS-250 FlexDial Framework Application Guide</i> for more information on datafilling tables FLEXFEAT and FLEXDIAL.</p>		
<p>—continued—</p>		

Step 8: Choose the type of subscription (continued)

Table 1
Validation failures of SUBR and SUBRPARM collectables (continued)

Table FLEXFEAT datafill of the OPTION field	Table FLEXDIAL associated datafill of the OPTION1 refinement for SUBR and SUBRPARM collectables	Switch action
FAILVAL FAILACT	not applicable	<ul style="list-style-type: none"> • For failed validations: <ul style="list-style-type: none"> — CAIN group is recorded — See Table 2 for datafill of the OPTION2 refinement and the associated failure action
FLDONLY	not applicable	<ul style="list-style-type: none"> • For successful validations (all digits filed): <ul style="list-style-type: none"> — CAIN group is recorded — remaining options datafilled in table FLEXFEAT are applied as necessary • For failed validations (at least one digit not filed): <ul style="list-style-type: none"> — CAIN group is recorded — See Table 2 for datafill of the OPTION2 refinement and the associated failure action
<p>Note: Refer to the <i>UCS DMS-250 FlexDial Framework Application Guide</i> for more information on datafilling tables FLEXFEAT and FLEXDIAL.</p>		
<p>—continued—</p>		

Step 8: Choose the type of subscription (continued)

Table 1
Validation failures of SUBR and SUBRPARM collectables (continued)

Table FLEXFEAT datafill of the OPTION field	Table FLEXDIAL associated datafill of the OPTION1 refinement for SUBR and SUBRPARM collectables	Switch action
<p>MLTCOSID (with the COSOVE option datafilled in table TRKFEAT)</p> <p>Note: Only applicable for ADDR and ADDRPARM collectables validated through table STDPRTCT where call is calling party billed.</p>	<p>not applicable</p>	<ul style="list-style-type: none"> • CAIN group is recorded • remaining options datafilled in table FLEXFEAT are applied as necessary • interaction is modified as necessary according to the index into table FLEXDIAL (provisioned by the COSOVE option in table TRKFEAT)
<p>MLTCOSID (without the COSOVE option in table TRKFEAT)</p> <p>Note: Only applicable for ADDR and ADDRPARM collectables validated through table STDPRTCT where call is calling party billed.</p>	<p>not applicable</p>	<ul style="list-style-type: none"> • CAIN group is recorded • remaining options datafilled in table FLEXFEAT are applied as necessary • FlexDial Framework sets delayed treatment
<p>Note: Refer to the <i>UCS DMS-250 FlexDial Framework Application Guide</i> for more information on datafilling tables FLEXFEAT and FLEXDIAL.</p>		
<p>—end—</p>		

Step 8: Choose the type of subscription (continued)

Table 2
Validation failures of SUBR and SUBRPARAM collectables

Table FLEXDIAL datafill of the OPTION2 refinement for SUBR and SUBRPARAM collectables	Switch action
TRMT	<ul style="list-style-type: none"> • datafilled treatment is set as the delayed treatment • CAIN group is recorded • remaining options datafilled in table FLEXFEAT are ignored
RESET WDIGACT NOPROMPT	<ul style="list-style-type: none"> • datafilled prompt is not reapplied for the digit collectable
RESET WDIGACT PROMPT	<ul style="list-style-type: none"> • RESET prompt (if provisioned) or the original prompt (if provisioned) is reapplied for the digit collectable
RESET WDIGACT PREVIOUS	<ul style="list-style-type: none"> • call is reset to the previously executed digit collectable
RESET WDIGACT PREPRMPT	<ul style="list-style-type: none"> • call is reset to the previously executed digit collectable that identifies a prompt or reset prompt option
RESET WODIGACT SAME	<ul style="list-style-type: none"> • process the same collectable again • replay prompt (if provisioned)
RESET WODIGACT PREVIOUS	<ul style="list-style-type: none"> • call is reset to the previously executed digit collectable
RESET WODIGACT PREPRMPT	<ul style="list-style-type: none"> • call is reset to the previously executed digit collectable that identifies a prompt or reset prompt option
RESET WODIGACT DPIDX	<ul style="list-style-type: none"> • index the provisioned FLEXDIAL tuple and begin collectable processing (replaces existing collectable list)
<p>Note: Refer to the <i>UCS DMS-250 FlexDial Framework Application Guide</i> for more information on datafilling table FLEXDIAL.</p>	
<p>—continued—</p>	

Step 8: Choose the type of subscription (continued)

The following table shows potential validation failures and corresponding actions for the CALLTYPE collectables

Table 3
Validation failures of CALLTYPE collectable

Table FLEXFEAT datafill of the OPTION field	Switch action when the CALLTYPE collectable is datafilled in table FLEXDIAL
CASUAL	<ul style="list-style-type: none"> • Option not applicable • CAIN group is recorded • remaining options datafilled in table FLEXFEAT are applied as necessary
CITYCODE	<ul style="list-style-type: none"> • City code and CAIN group are recorded • remaining options datafilled in table FLEXFEAT are applied as necessary
FAILVAL TRMT	<ul style="list-style-type: none"> • CAIN group is recorded • remaining options datafilled in table FLEXFEAT are applied as necessary • FlexDial Framework sets delayed treatment
FAILVAL FAILACT	<ul style="list-style-type: none"> • Option not applicable (switch FLEX301 trouble log generated) • CAIN group is recorded • remaining options datafilled in table FLEXFEAT are applied as necessary
FLDONLY	<ul style="list-style-type: none"> • Option is not applicable • CAIN group is recorded • remaining options datafilled in table FLEXFEAT are applied as necessary
<p>Note: Refer to the <i>UCS DMS-250 FlexDial Framework Application Guide</i> for more information on datafilling tables FLEXFEAT and FLEXDIAL.</p>	

Step 8: Choose the type of subscription (continued)

Table 3
Validation failures of CALLTYPE collectable (continued)

Table FLEXFEAT datafill of the OPTION field	Switch action when the CALLTYPE collectable is datafilled in table FLEXDIAL
<p>MLTCOSID (with the COSOVE option datafilled in table TRKFEAT)</p> <p>Note: Only applicable for ADDR and ADDRPARM collectables validated through table STDPRTCT where call is calling party billed.</p> <p>MLTCOSID (without the COSOVE option in table TRKFEAT)</p> <p>Note: Only applicable for ADDR and ADDRPARM collectables validated through table STDPRTCT where call is calling party billed.</p>	<ul style="list-style-type: none"> • CAIN group is recorded • remaining options datafilled in table FLEXFEAT are applied as necessary • interaction is modified as necessary according to the index into table FLEXDIAL (provisioned by the COSOVE option in table TRKFEAT) <ul style="list-style-type: none"> • CAIN group is recorded • remaining options datafilled in table FLEXFEAT are applied as necessary • FlexDial Framework sets delayed treatment
<p>Note: Refer to the <i>UCS DMS-250 FlexDial Framework Application Guide</i> for more information on datafilling tables FLEXFEAT and FLEXDIAL.</p>	

Step 8: Choose the type of subscription Precedence Level 2 (Address)

Multiple collection of same-type digits

Due to the flexibility of the FlexDial Framework, same-type digits can be collected multiple times possibly causing the associated CAINGRP to change throughout the life of the call. The following table shows the effects on subscription by repeated, same-level, collectable execution.

Subscription based on repeated same level collectable execution

Precedence Level	Current collectable	Was a CAIN group identified prior to current collectable?	Is a CAIN group associated with current collectable?	Effect on subscription
2	ADDR or ADDRPARM collectable; ADDR collectible	No	No	No changes
	ADDR or ADDRPARM collectable; ADDR collectible	No	Yes	New CAIN group stored; overwrites previous group
	ADDR or ADDRPARM collectable; ADDR collectible	Yes	No	Cancelled – CAIN group is removed from memory. (Note)
	ADDR or ADDRPARM collectable; ADDR collectible	Yes	Yes	New CAIN group stored; overwrites previous group
3	SUBR or SUBRPARM with any CAINFLG option	No	No	No changes
<p>Note: Subscription is cancelled when the same type of digits are collected without identifying a CAIN group. For Precedence Level 3 SUBR and SUBRPARM collectables, digits are the same type when the CAINFLG option is the same.</p>				
—continued—				

Step 8: Choose the type of subscription Precedence Level 2 (Address) (continued)

Subscription based on repeated same level collectable execution (continued)

Precedence Level	Current collectable	Was a CAIN group identified prior to current collectable?	Is a CAIN group associated with current collectable?	Effect on subscription
	SUBR or SUBRPARM with any CAINFLG option	No	Yes	New CAIN group stored; overwrites previous group. The new CAIN group is associated with the SUBR or SUBRPARM collectable.
	CALLTYPE	No	No	No changes
	CALLTYPE	No	Yes	New CAIN group stored; overwrites previous group. The new CAIN group is associated with the CALLTYPE collectable.
	SUBR or SUBRPARM with same CAINFLG as prior collectable updating this level of subscription	Yes (SUBR or SUBRPARM)	No	Cancelled – CAIN group is removed from memory. (Note 1)
	SUBR or SUBRPARM with same CAINFLG as prior collectable updating this level of subscription	Yes (SUBR or SUBRPARM)	Yes	New CAIN group stored; overwrites previous group. The new CAIN group is associated with the SUBR or SUBRPARM collectable and based on the same CAINFLG.
	SUBR or SUBRPARM with different CAINFLG than prior collectable updating this level of subscription	Yes (SUBR or SUBRPARM)	No	No changes
<p>Note: Subscription is cancelled when the same type of digits are collected without identifying a CAIN group. For Precedence Level 3 SUBR and SUBRPARM collectables, digits are the same type when the CAINFLG option is the same.</p>				
—continued—				

Step 8: Choose the type of subscription Precedence Level 2 (Address) (continued)

Subscription based on repeated same level collectable execution (continued)

Precedence Level	Current collectable	Was a CAIN group identified prior to current collectable?	Is a CAIN group associated with current collectable?	Effect on subscription
	SUBR or SUBRPARM with different CAINFLG than prior collectable updating this level of subscription	Yes (SUBR or SUBRPARM)	Yes	New CAIN group stored; overwrites previous group. The new CAIN group is associated with the SUBR or SUBRPARM collectable and based on the new CAINFLG.
	CALLTYPE	Yes (SUBR or SUBRPARM)	No	No changes
	CALLTYPE	Yes (SUBR or SUBRPARM)	Yes	New CAIN group stored; overwrites previous group. The new CAIN group is associated with the CALLTYPE collectable.
	SUBR or SUBRPARM with any CAINFLG option	Yes (CALLTYPE)	No	No changes
	SUBR or SUBRPARM with any CAINFLG option	Yes (CALLTYPE)	Yes	New CAIN group stored; overwrites previous group. The new CAIN group is associated with the CALLTYPE collectable.
	CALLTYPE	Yes (CALLTYPE)	No	No changes
	CALLTYPE	Yes (CALLTYPE)	Yes	New CAIN group stored; overwrites previous group. The new CAIN group is associated with the CALLTYPE collectable.
<p>Note: Subscription is cancelled when the same type of digits are collected without identifying a CAIN group. For Precedence Level 3 SUBR and SUBRPARM collectables, digits are the same type when the CAINFLG option is the same.</p>				
—continued—				

Step 8: Choose the type of subscription Precedence Level 2 (Address) (continued)

Subscription based on repeated same level collectable execution (continued)

Precedence Level	Current collectable	Was a CAIN group identified prior to current collectable?	Is a CAIN group associated with current collectable?	Effect on subscription
4	SUBR or SUBRPARAM with CALLING option (no CAINFLG)	No	No	No changes
		No	Yes	New CAIN group stored; overwrites previous group.
		Yes	No	Cancelled – CAIN group is removed from memory. (Note 1)
		Yes	Yes	New CAIN group stored; overwrites previous group.
<p>Note: Subscription is cancelled when the same type of digits are collected without identifying a CAIN group. For Precedence Level 3 SUBR and SUBRPARAM collectables, digits are the same type when the CAINFLG option is the same.</p>				
—end—				

Triggers and subscription methods

For information on how the triggers map to the subscription methods available, refer to the *UCS DMS-250 NetworkBuilder Application Guide*.

CAIN902 log

NetworkBuilder call processing generates the CAIN902 log whenever subscription is updated. Be aware that generation of the CAIN902 log can impact real-time processing. Enable and disable the CAIN902 log through table CAINPARAM (CAIN900_LOGS_ENABLED). For more information on CAIN902 log, refer to the *UCS DMS-250 Logs Reference Manual*.

Step 8: Choose the type of subscription Precedence Level 2 (Address) (continued)

Provisioning NetworkBuilder subscription

The following table shows the subscription precedence level, the subscription method, applicable originating agents, and provisioning rules.

Provisioning NetworkBuilder subscription

Precedence level	Method	Originating agent	Provisioning
1	SCP	All	The SCP returns the <code>cainGroup</code> extension parameter in an <code>Analyze_Route</code> message.
2	ADDR	All	Datafill the CAINGRP option in subtable STDPRT of table STDPRTCT. NetworkBuilder supports the following pretranslator selectors (field PRERTSEL): CT, UA (non-AXXESS), ED, UAX (AXXESS), and NT (for IP and IN)
3	AUTH SUBR/ SUBRPARM	Non-AXXESS AXXESS	<p>Datafill the CAINGRP option in one of the authcode tables (AUTHCODU, AUTHCDU2, AUTHCDU3, AUTHCDU4, or AUTHCDU5)</p> <ol style="list-style-type: none"> 1 Datafill the SUBR or SUBRPARM collectables in table FLEXDIAL. 2 Datafill the CAINGRP option in the associated tuple in table FLEXFEAT. 3 Datafill the CAINFLG option in the associated tuple in table FLEXTYPE. <p>Note: The CAINFLG option is required for subscription to NetworkBuilder services. Datafilling the CAINFLG option allows you to identify authorization codes (AUTH), personal identification numbers (PIN), calling card numbers (MCCS), account codes (ACCT), and calling party addresses (CLGPTYADD) for subscription to NetworkBuilder services.</p>
—continued—			

Step 8: Choose the type of subscription Precedence Level 2 (Address) (continued)

Provisioning NetworkBuilder subscription (continued)

Precedence level	Method	Originating agent	Provisioning
	CALLTYPE	AXXESS	<ol style="list-style-type: none"> 1 Datafill the CALLTYPE collectable in table FLEXDIAL. 2 Datafill the CAINGRP option in the associated tuple in table FLEXFEAT.
4	ANI	Non-AXXESS	Datafill the CAINGRP option in table ANISCUSP (or ANIVAL and UNIPROF).
	SUBR/ SUBRPARM	AXXESS	<ol style="list-style-type: none"> 1 Datafill the SUBR or SUBRPARM collectables in table FLEXDIAL. 2 Datafill the CAINGRP option in the associated tuple in table FLEXFEAT. 3 Datafill the CALLING option in the associated tuple in table FLEXTYPE. <p>Note: If both the CAINFLG option and CALLING option are present in the FLEXTYPE tuple, the CAINFLG option takes precedence and any subscription associated with the call is a Precedence Level 3 subscription method.</p>
5	Agent	Non-AXXESS AXXESS	<p>Datafill the CAINGRP option in table TRKGRP.</p> <ol style="list-style-type: none"> 1 Datafill the agent in table TRKGRP. 2 Datafill the CAINGRP option in the associated tuple for the appropriate option vector in table TRKFEAT.
6	Office	All	Datafill parameter CAIN_OFFICE_GROUP in table CAINPARAM.
—end—			

Step 8: Choose the type of subscription Precedence Level 2 (Address) (end)

The following table shows the difference between a Precedence Level 3 SUBR/SUBRPARAM subscription and a Precedence Level 4 subscription method.

Precedence Level 3 and a Precedence Level 4 SUBR/SUBRPARAM subscription method

Precedence Level	Table FLEXFEAT	Table FLEXTYPE
3	CAINGRP option is datafilled	CAINFLG option is datafilled
4	CAINGRP option is datafilled	CALLING option is datafilled
no subscription	CAINGRP option is datafilled	Neither the CAINFLG nor the CALLING option is datafilled
—end—		

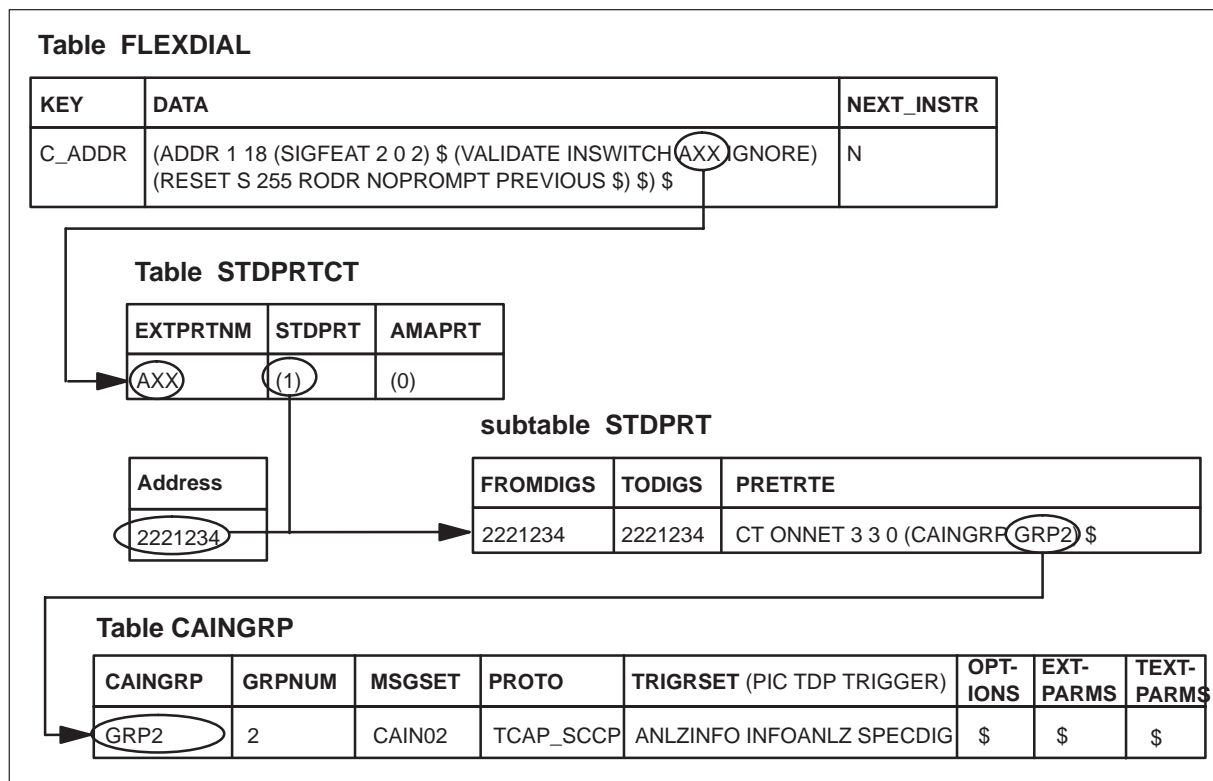
Step 8: Choose the type of subscription Precedence Level 2 (Address)

Subscribing to NetworkBuilder services (address)

Datafill Precedence Level 2 subscription in table STDPRTCT. The FlexDial Framework identifies pretranslator's associated with a call in either table FLEXDIAL (default) or in table MSGCTR.

The following example shows the interaction between tables FLEXDIAL, STDPRTCT, and CAINGRP.

Precedence Level 2 subscription-CAINGRP interaction



At the CI prompt

- 1 Enter table STDPRTCT.
- 2 Position on the pretranslator requiring NetworkBuilder services.
- 3 Enter subtable STDPRT.
- 4 Position on the address requiring NetworkBuilder originating call model (OCM) services.

Step 8: Choose the type of subscription Precedence Level 2 (Address) (end)

Note: In this example, an existing address subscribes to NetworkBuilder. New addresses can also subscribe to NetworkBuilder services.

- 5 Subscribe to NetworkBuilder by changing the tuple to include the CAINGRP option using the following format in the STDPRTOPT field:

> CAINGRP caingrp

where

caingrp is the CAIN group (defined in table CAINGRP) providing subscription services to the address.

Sample entry: **>CHA 2221234 2221234 CT ONNET 3 3 0 CAINGRP grp2 \$**

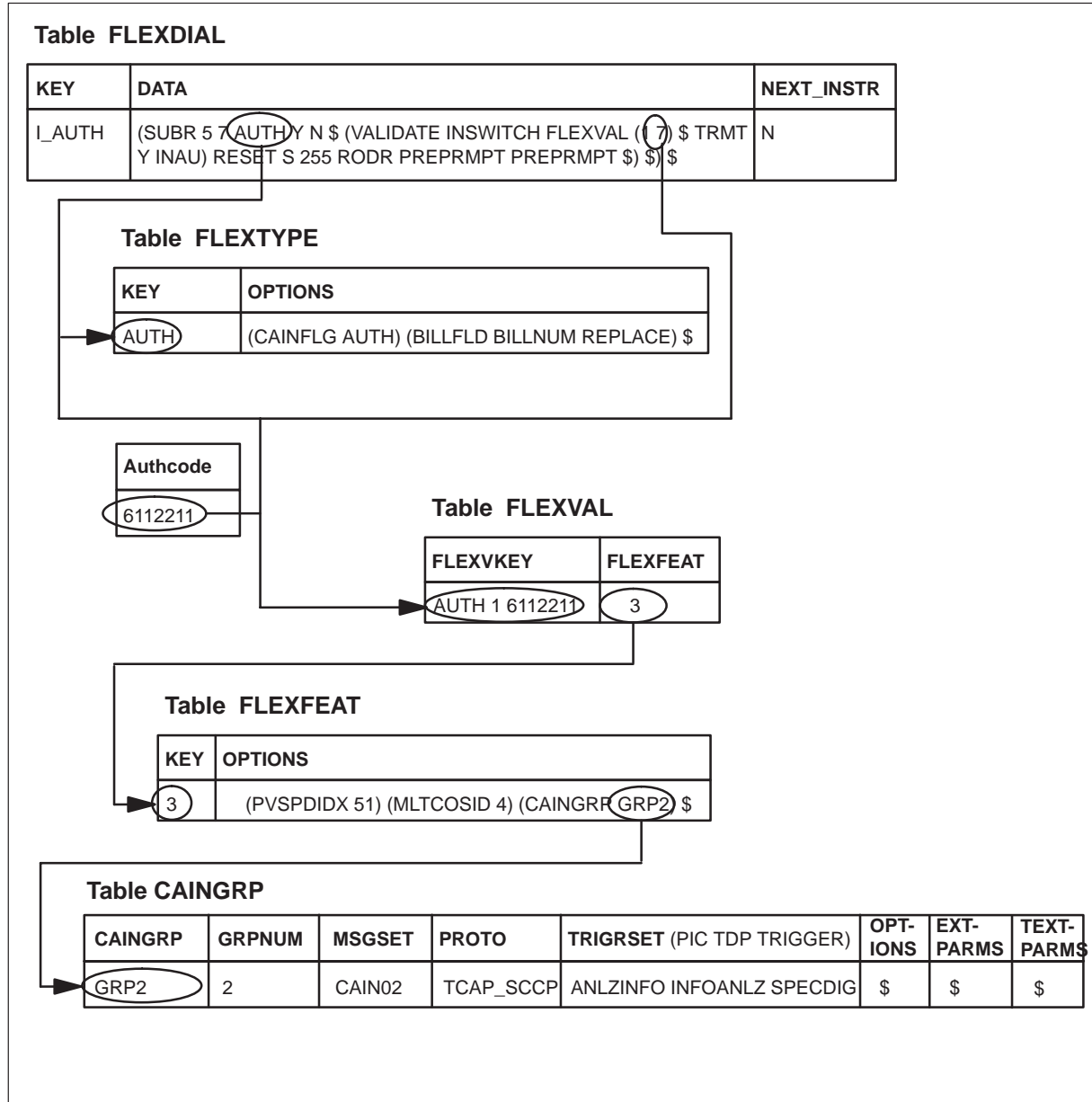
The address has subscribed to NetworkBuilder OCM services.

Step 8: Choose the type of subscription Precedence Level 3 (SUBR and SUBRPARM)

Subscribing to NetworkBuilder services (SUBR and SUBRPARM collectables)

The following example shows the interaction between tables FLEXDIAL, FLEXTYPE, FLEXVAL, FLEXFEAT, and CAINGRP.

Precedence Level 3 subscription-CAINGRP interaction



Step 8: Choose the type of subscription Precedence Level 3 (SUBR and SUBRPARM) (end)

At the CI prompt

- 1 Enter table FLEXTYPE.
- 2 Position on the tuple associated with the collectable requiring originating call model (OCM) NetworkBuilder services.
- 3 Set the CAINFLG option by changing the tuple using the following format in the OPTIONS field:

> CAINFLG cainflg

where

cainflg is the CAIN digit type associated with the collectable (ACCT, AUTH, MCCS, PIN, CLGPTYADD).

Sample entry: **>ADD auth CAINFLG auth BILLFLD billnum replace \$**

- 4 Enter table FLEXFEAT.
- 5 Position on the tuple associated with the collectable requiring NetworkBuilder services.
- 6 Subscribe to NetworkBuilder by changing the tuple to include the CAINGRP option using the following format in the OPTIONS field:

>CAINGRP caingrp

where

> CAINGRP caingrp

where

caingrp is the CAIN group (defined in table CAINGRP) providing subscription services to the address.

Sample entry: **>CHA 3 pvspidx 51 mltcoid 4 CAINGRP grp2 \$**

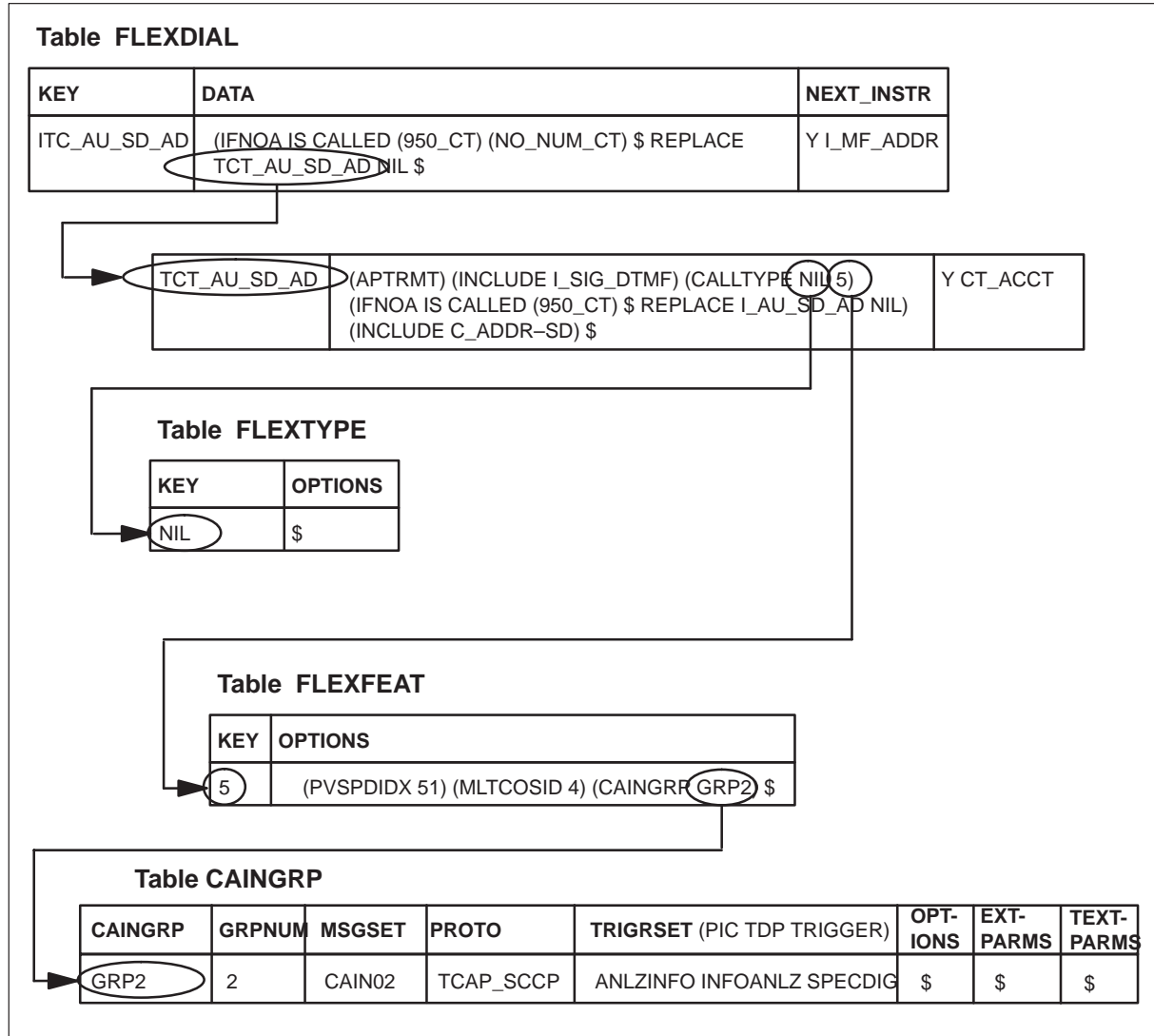
The SUBR or SUBRPARM collectable has subscribed to NetworkBuilder OCM services.

Step 8: Choose the type of subscription Precedence Level 3 (CALLTYPE)

Subscribing to NetworkBuilder services (CALLTYPE collectables)

The following example shows the interaction between tables FLEXDIAL, FLEXTYPE, FLEXFEAT, and CAINGRP.

Precedence Level 3 subscription-CAINGRP interaction



At the CI prompt

- 1 Enter table FLEXFEAT.

Step 8: Choose the type of subscription Precedence Level 3 (CALLTYPE) (end)

- 2 Position on the tuple associated with the collectable requiring NetworkBuilder originating call model (OCM) services.
- 3 Subscribe to NetworkBuilder by changing the tuple to include the CAINGRP option using the following format in the OPTIONS field:

>CAINGRP caingrp

where

> CAINGRP caingrp

where

caingrp is the CAIN group (defined in table CAINGRP) providing subscription services to the address.

Sample entry: **>CHA 5 pvspidx 51 mlcosid 4 CAINGRP grp2 \$**

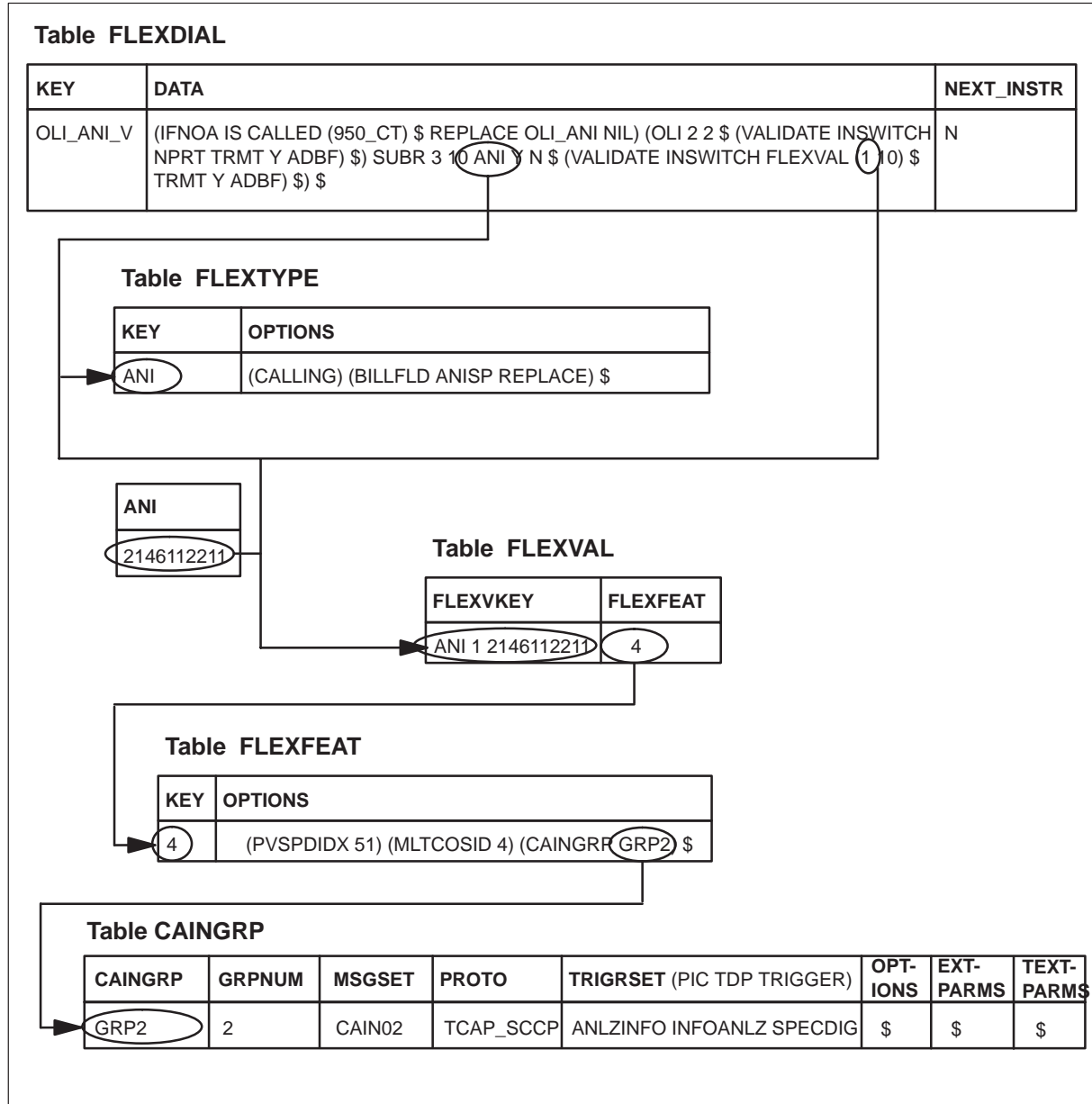
The CAINTYPE collectable has subscribed to NetworkBuilder OCM services.

Step 8: Choose the type of subscription Precedence Level 4 (SUBR and SUBRPARM)

Subscribing to NetworkBuilder services (SUBR and SUBRPARM collectables)

The following example shows the interaction between tables FLEXDIAL, FLEXTYPE, FLEXVAL, FLEXFEAT, and CAINGRP.

Precedence Level 3 subscription-CAINGRP interaction



Step 8: Choose the type of subscription Precedence Level 4 (SUBR and SUBRPARM) (end)

At the CI prompt

- 1 Enter table FLEXTYPE.
- 2 Position on the tuple associated with the collectable requiring NetworkBuilder originating call model (OCM) services.
- 3 Set the CALLING option by changing the tuple using the following format in the OPTIONS field:

> CALLING

Sample entry: **>ADD ani CALLING BILLFLD anisp replace \$**

- 4 Enter table FLEXFEAT.
- 5 Position on the tuple associated with the collectable requiring NetworkBuilder services.
- 6 Subscribe to NetworkBuilder by changing the tuple to include the CAINGRP option using the following format in the OPTIONS field:

> CAINGRP caingrp

where

caingrp is the CAIN group (defined in table CAINGRP) providing subscription services to the address.

Sample entry: **>CHA 4 pvspidx 51 mltcoid 4 CAINGRP grp2 \$**

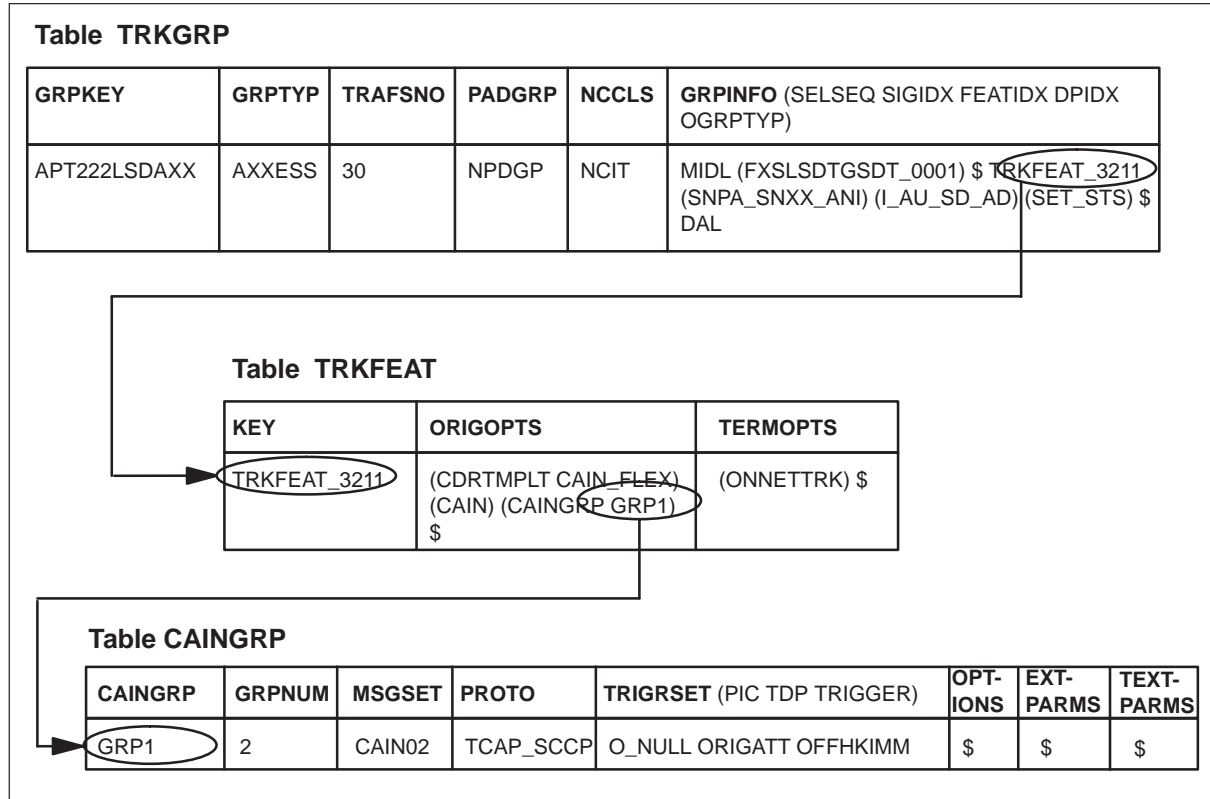
The SUBR or SUBRPARM collectable has subscribed to NetworkBuilder OCM services.

Step 8: Choose the type of subscription Precedence Level 5 (Agent)

Subscribing to NetworkBuilder services (AXCESS originating and terminating agencies)

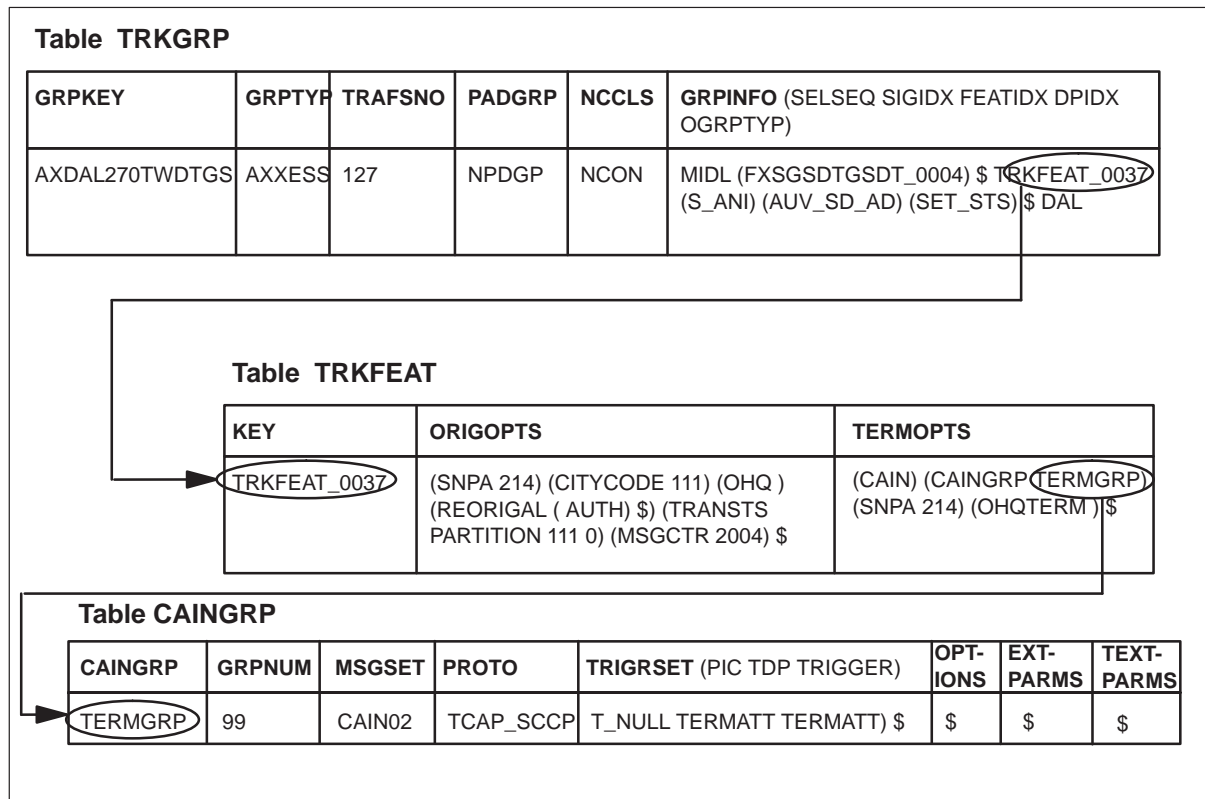
The following examples shows the interaction between tables TRKGRP, TRKFEAT, and CAINGRP.

AXCESS agent subscription – originating CAINGRP interaction



Step 8: Choose the type of subscription Precedence Level 5 (Agent) (continued)

AXXESS agent subscription – terminating CAINGRP interaction



At the CI prompt

- 1 Enter table TRKFEAT.
- 2 Position on the tuple associated with the agent requiring NetworkBuilder services.

Step 8: Choose the type of subscription Precedence Level 5 (Agent) (end)

ATTENTION

The CAIN and CAINGRP options are independent. Therefore, subscription to a CAIN group using the CAINGRP option does not require the CAIN option. However, all calls requiring NetworkBuilder services must originate from a CAIN-capable agency.

The TCAIN and TCAINGRP options are independent. Therefore, subscription to a CAIN group using the TCAINGRP option does not require the TCAIN option. However, all calls requiring TCAIN services must terminate to a TCAIN-provisioned agency.

- 3 Subscribe to NetworkBuilder by using the following format:

>REP key origopts termopts

where

key is the index datafilled in table TRKGRPs FEATIDX.
origopts are the originating agent options, including the CAINGRP option, where
CAINGRP is the option.
CAIN_GROUP_TYPE is the CAIN group (defined in table CAINGRP) providing subscription services to the call attribute.
termopts are the terminating agent options.

Sample entry: **>REP TRKFEAT_3211 CDRTMPLT cain_flex CAIN CAINGRP grp1 \$ ONNETTRK \$**

Sample entry: **>REP TRKFEAT_0037 SNPA 214 CITYCODE 111 OHQ REORIGAL AUTH \$ TRANSTS PARTITION 111 0 MSGCTR 2004 \$ SNPA 214 OHQTERM \$**

The agent has subscribed to NetworkBuilder services.

Step 9: Define the trigger criteria

ATTENTION

Triggers require the corresponding CAIN0500-series of SOC options. The *Office_Code* trigger requires CAIN0700 SOC option. The SS7 Inter-IMT and SS7 Global-IMT agents require CAIN0604 and CAIN0605 SOC options respectively, in order to trigger. Refer to the *UCS DMS-250 Software Optionality Control User's Manual* for more information about SOCs.

When NetworkBuilder call processing encounters a TDP, the following steps have been completed:

- 1 Data required to process the call as a NetworkBuilder call (at the current PIC, TDP, and trigger) has been gathered.
- 2 One or more subscription methods have been identified for the call.
- 3 Table CAINGRP enables the current trigger set.

Note: A trigger set identifies the PIC, TDP, and trigger.

Trigger evaluations

Trigger criteria

The following table lists each trigger and the digit types available for trigger criteria evaluation.

Triggers and trigger criteria

	Information digits (INFO) (Note 1)	ADIN (Note 5)	ANI or CLID (Note 2)	Translated address (XLADDR) (Note 3)	Address (ADDR) (Note 3)	CIC (Note 4)
<i>Off_Hook_Immediate</i>						
<i>O_Feature_Requested</i>	√		√		√	√
<i>Tollfree_Service</i> (Note 7)					√	
<i>Offhook_Delay</i>	√	√	√		√	√
<i>Shared_Interoffice_Trunk</i>	√	√	√		√	√
—continued—						

Step 9: Define the trigger criteria (continued)**Triggers and trigger criteria** (continued)

	Information digits (INFO) (Note 1)	ADIN (Note 5)	ANI or CLID (Note 2)	Translated address (XLADDR) (Note 3)	Address (ADDR) (Note 3)	CIC (Note 4)
<i>PRI_B-Channel</i> (Note 6)		√	√		√	
<i>Specific_Feature_Code</i>				√	√	
<i>Customized_Dialing_Plan</i>				√	√	
<i>Specific_Digit_String</i>	√	√	√	√	√	√
<i>Office_Code</i>				√		
<i>Network_Busy</i>	√	√	√	√	√	√
<i>O_Called_Party_Busy</i>	√	√	√	√	√	√
<i>O_No_Answer</i>	√	√	√	√	√	√
<i>O_IEC_Reorigination</i>	√	√	√		√	√
<i>Termination_Attempt</i>				√		
<p>Note 1: Datafill the OLI or OLIPARM collectables in table FLEXDIAL to collect information digits. Note 2: Datafill the SUBR or SUBRPARM collectable in table FLEXDIAL (with the corresponding CALLING option in table FLEXTYPE) to collect an ANI. Note 3: Datafill the ADDR or ADDRPARM collectables in table FLEXDIAL to collect an address. Note 4: Datafill the CIC or CICPARM collectables in table FLEXDIAL to collect a CIC. Note 5: AXCESS originations do not support the ADIN digit type. Note 6: CAIN/FlexDial interaction does not support the <i>PRI_B-Channel</i> trigger. Note 7: The digit type is implied but not provisioned.</p>						
—end—						

The last collected digits are always used as trigger criteria, with the following exceptions to address digits on any CAIN-capable originating agent:

- When the first address collected is a universal access number, the UA is evaluated and used as the address. When no query occurs and additional address digits are collected, the UA digits are used to populate the `universalAccess` extension parameter. The additional address digits are evaluated for the query.
- When the first address collected is a speed number, the translated speed number (through table SPEEDTAB) is used as trigger criteria.

Step 9: Define the trigger criteria (continued)

Note: This applies to public, private, and hotline speed numbers.

- When the first address collected is an N00 number, the trigger criteria is used against the N00 number. The XLADDR trigger criteria is used to analyze the real address for an N00 number.
- When the UIF executes the *O_Feature_Requested* ADDR collectible, the digits collected are evaluated against the ADDR trigger criteria. If the SCP does not return a *CalledPartyID*, then the ADDR collectable is also used for the XLAADDR digit type.

Error actions

For information on the valid error actions in each trigger table, refer to the *UCS DMS-250 NetworkBuilder Application Guide*.

Datafilling trigger options

For information on the valid trigger options in each trigger table, refer to the *UCS DMS-250 NetworkBuilder Application Guide*.

CAIN903 log

NetworkBuilder call processing generates the CAIN903 log whenever the trigger criteria is met. Be aware that generation of the CAIN903 log can impact real-time processing. Enable and disable the CAIN903 log through table CAINPARM (CAIN900_LOGS_ENABLED). Refer to the *UCS DMS-250 NetworkBuilder Application Guide* or the *UCS DMS-250 Logs Reference Manual* for more information on the CAIN903 log.

Step 10: Be familiar with digit collection

NetworkBuilder (CAIN)/FlexDial interact with the following methods of digit collection.

- FlexDial Framework collectables during **Collect_Information**.
- NetworkBuilder's user interaction framework (UIF) digit collection at *O_Feature_Requested*

Note: Refer to the *UCS DMS-250 NetworkBuilder Application Guide* for more information on digit collection for non-AXXESS agents.

FlexDial Framework collectables

NetworkBuilder interacts with the following Digit collectables:

- SUBR
- ADDR
- OLI

Note: You can provision “reset to” actions for SUBR, ADDR, and OLI collectables. Refer to *UCS DMS-250 FlexDial Framework Application Guide* for more information.

- CIC

Note: The SUBR collectable expands CIC processing for AXXESS agents and can replace the CIC collectable. Refer to *UCS DMS-250 FlexDial Framework Application Guide* for more information.

- SUBRPARAM
- ADDRPARAM
- OLIPARAM
- CICPARAM

Note: The SUBRPARAM collectable expands CIC processing for AXXESS agents and can replace the CIC collectable. Refer to *UCS DMS-250 FlexDial Framework Application Guide* for more information.

NetworkBuilder interacts with the following Calltype collectables:

- CALLTYPE
- CLRFTRS

FlexDial collectables are provisioned in table FLEXDIAL. Refer to the *UCS DMS-250 FlexDial Framework Application Guide* for more information.

Step 10: Be familiar with digit collection (continued)

SUBR and SUBRPARM collectables

The FlexDial Framework allows you to abstractly define subscriber numbers (through tables FLEXDIAL, FLEXFEAT, and FLEXTYPE), as opposed to viewing authcodes, ANIs, PINs, account codes, and calling card numbers traditionally (as hard-coded within the switch). Every aspect of the subscriber number is provided through datafill.

Since the FlexDial Framework does not distinguish between the different subscriber numbers, one type of validation (through tables FLEXVAL and FLEXFEAT) can be performed on all subscriber numbers.

However, NetworkBuilder call processing logic is based on traditional digit usage. Therefore, the FlexDial SUBR and SUBRPARM collectables must be mapped to traditional digit types.

The CAINFLG option

The CAINFLG option in table FLEXTYPE

- allows subscription based on the subscriber digits
- maps abstract FlexDial subscriber numbers to the traditional digit types recognized by NetworkBuilder. This mapping identifies the digit type and therefore, through hard-coded logic, the nature of address (NOA) used to populate outgoing message parameters

Note: FlexDial nature of address identifiers are not used by NetworkBuilder.

The following table shows the valid refinements for the CAINFLG option:

CAINFLG refinements

Refinement	NetworkBuilder interpretation	Valid Network-Builder NOAs
ACCT	Identifies the digits and NOA for population of the <i>AccessCode</i> parameter	ACCT
AUTH	Identifies the digits and NOA for population of the <i>ChargeNumber</i> parameter (see Note)	AUTH
<p>Note 1: For calling party billed calls, NetworkBuilder software populates the <i>ChargeNumber</i> parameter with the digits identified in table FLEXTYPE by the BILLFLD option that identifies BILLNUM as the billing field. The CAINFLG option should be specified for any digits used to populate the <i>ChargeNumber</i> parameter.</p>		

Step 10: Be familiar with digit collection (continued)**CAINFLG refinements**

Refinement	NetworkBuilder interpretation	Valid Network-Builder NOAs
CLGPTYADD	Identifies the digits normally delivered in the Calling Party Address parameter of the incoming IAM	based on incoming IAM
MCCS	Identifies the digits and NOA for population of the <i>ChargeNumber</i> parameter (see Note)	MCCS
PIN	Identifies the digits and NOA for population of the <i>CollectedDigits</i> parameter	PIN
<p>Note 1: For calling party billed calls, NetworkBuilder software populates the <i>ChargeNumber</i> parameter with the digits identified in table FLEXTYPE by the BILLFLD option that identifies BILLNUM as the billing field. The CAINFLG option should be specified for any digits used to populate the <i>ChargeNumber</i> parameter.</p>		
—end—		

The CALLING option

Use the CALLING option in table FLEXTYPE to identify the SUBR or SUBRPARAM collectable as the ANI for the call. NetworkBuilder checks these FlexDial-collected digits against datafilled trigger criteria for the ANI digit type.

Subscription

A call can subscribe to NetworkBuilder services through the SUBR and SUBRPARAM collectables. The following table shows the requirements for provisioning subscription based on the SUBR and SUBRPARAM collectables:

Subscription provisioning

Precedence Level	Table FLEXTYPE provisioning	Table FLEXFEAT provisioning	Description
3	CAINFLG option	CAINGRP option	Allows subscription based on a subscriber number
4	CALLING option	CAINGRP option	Allows subscription based on the SUBR or SUBRPARAM collectable as an ANI
—end—			

Step 10: Be familiar with digit collection (continued)

ADDR and ADDRPARM collectables

FlexDial uses the ADDR and ADDRPARM collectables to collect called party addresses. NetworkBuilder checks these FlexDial-collected digits against datafilled trigger criteria for the ADDR and XLAADDR digit type.

CIC and CICPARM collectables

FlexDial uses the CIC and CICPARM collectables to collect carrier identification codes. NetworkBuilder checks these FlexDial-collected digits against datafilled trigger criteria for the CIC digit type.

OLI and OLIPARM collectables

FlexDial uses the OLI and OLIPARM collectables to collect information digits. NetworkBuilder checks these FlexDial-collected digits against datafilled trigger criteria for the INFO digit type.

CALLTYPE collectables

FlexDial uses the CALLTYPE collectable to apply features (from table FLEXFEAT) to the call. NetworkBuilder allows subscription to a CAIN group through the CALLTYPE collectable.

Digit validation

Once the switch executes the FlexDial Framework collectables, digit validation is performed through tables FLEXTYPE and FLEXVAL for SUBR and SUBRPARM collectables; through table STDPRTCT for ADDR, ADDRPARM, OLI, and OLIPARM collectables; no validation is performed for CIC and CICPARM collectables.

UIF digit collection at O_Feature_Requested

When the user interaction framework (UIF) performs *O_Feature_Requested* digit collection, the collected digits are handled in one of the following ways:

- CAINFLG identifies subscription type
 - Parameter OFTRREQ_FLEXTYPE_MAP (table CAINPARM) is datafilled to provide a mapping of the *O_Feature_Requested* collectable to the associated FlexDial digit type.
 - The datafill in table CAINPARM provides an index into table FLEXTYPE.
 - The tuple in table FLEXTYPE contains the digit type (through the CAINFLG option) used to identify the type of subscription.
 - Digits are available for CAIN parameter population.

Step 10: Be familiar with digit collection (continued)

- CAINFLG value assumed by call processing
 - Parameter OFTRREQ_FLEXTYPE_MAP (table CAINPARAM) is datafilled to provide a mapping of the *O_Feature_Requested* collectable to the associated FlexDial digit type.
 - The datafill in table CAINPARAM provides an index into table FLEXTYPE.
 - The tuple in table FLEXTYPE does not contain the digit type (the CAINFLG option is not provisioned). Therefore, the following table identifies the assumed CAINFLG values

CAINFLG values

OFTRREQ collection type	Assumed CAINFLG value
AUTH	CAINFLG AUTH
ACCT	CAINFLG ACCT
PIN	CAINFLG PIN
ADDR	see Note
CARD	CAINFLG MCCS
<p>Note: FlexDial performs address collection through the ADDR and ADDRPARAM collectables which do not use the CAINFLG option. Therefore, no CAINFLG value is associated with the OFTRREQ collection type of ADDR.</p>	
—end—	

- Digits are available for CAIN parameter population.
- Digits are not available for CAIN parameter population.
 - Digit type is not defined in parameter OFTRREQ_FLEXTYPE_MAP (table CAINPARAM)

Note 1: *O_Feature_Requested* digit collection does not update Precedence Level 3 subscription.

Note 2: For UIF digit collection, NetworkBuilder does not interact with any other options in table FLEXTYPE.

Digit validation

The SCP performs digit validation for OFTRREQ-collected digits.

Step 10: Be familiar with digit collection (continued)

Digit mapping

You must map digits from FlexDial usage to NetworkBuilder, and vice versa.

FlexDial to NetworkBuilder

Map FlexDial digits for use by NetworkBuilder software through the CAINFLG option in table FLEXTYPE.

NetworkBuilder to FlexDial

Map NetworkBuilder digits and table RTEATTR (via SX selector) for use by the FlexDial Framework and NetworkBuilder queries later in the call model through the OFTRREQ_FLEXTYPE_MAP CAIN parameter in table CAINPARM.

Multiple sources of digit collection

Digit collection for a call originating on a CAIN-capable AXXESS agent can occur from two sources:

- FlexDial-initiated digit collection as directed by table FLEXDIAL
- when provisioned, UIF-initiated digit collection as directed by table OFTRREQ

The FlexDial Framework can execute a SUBR or SUBRPARM collectable to obtain authorization codes, account codes, PINs, addresses, and calling card numbers (as mapped by the CAINFLG option in table FLEXTYPE).

However, when trigger criteria is met at *O_Feature_Requested*, the UIF can also initiate digit collection (as datafilled in table OFTRREQ) for authorization codes, account codes, PINs, addresses, and calling card numbers.

ATTENTION

When multiple sets of digits are collected for a particular digit type, the last collected set is used to process the call provided the mapping is provisioned correctly.

Note: In some cases, call processing may re-enter the FlexDial Framework. Refer to Chapter 3, “Call processing,” for more information.

Step 10: Be familiar with digit collection (end)

Limitations and restrictions

- The FlexDial Framework does not allow “reset to” handling to extend back beyond the latest (re)entry into FlexDial. Therefore, unlike table FLEXDIAL framework digit collectables, entities outside of the FlexDial Framework (including *O_Feature_Requested*) are not required to “undo” their collection of digit information.
- Upon re-entry into the FlexDial Framework, the last collected digits are retained by the FlexDial Framework (whether or not they were collected by the FlexDial Framework). Therefore, any digits collected prior to re-entry, that are not re-collected during re-entry, are used to populate outgoing messages to the SCP.

Step 10: Be familiar with digit collection Mapping OFTRREQ to FlexDial (end)

When the UIF performs digit collection directed by table OFTRREQ, the associated digits stored in the FlexDial Framework must be updated to reflect the new digits collected by the UIF.

The OFTRREQ_FLEXTYPE_MAP parameter provides the mapping between OFTRREQ-collected digits and the associated digits used by the FlexDial Framework.

Define the OFTRREQ_FLEXTYPE_MAP parameter

At the CI prompt

- 1 Enter table CAINPARAM.
- 2 Replace the parameter by typing:
>REP OFTRREQ_FLEXTYPE_MAP parmval

where

parmval is the type of digit collected by the UIF (ACCT, AUTH, CARD, or PIN) and the index into table FLEXTYPE that is associated with the FlexDial digits.

Sample entry: **>REP OFTRREQ_FLEXTYPE_MAP acct flexacct \$**

The OFTRREQ_FLEXTYPE_MAP parameter is defined.

Call processing

Call processing is controlled by several different software pieces within the UCS DMS-250 software architecture. NetworkBuilder software is able to control a call and perform call processing functions throughout the life of a call. FlexDial software only has limited control of the call during **Collect_Information** or **Analyze_Information**.

For basic NetworkBuilder call processing, refer to the *UCS DMS-250 NetworkBuilder Application Guide*.

When FlexDial completes its processing portion of the call, the call is said to “exit” the FlexDial Framework and other software within the UCS DMS-250 switch continues the handling of the call. In certain scenarios, the call may re-enter the FlexDial Framework.

This chapter discusses the scenarios associated with exiting and re-entering the FlexDial framework.

Exit scenarios

During the life of a call, call processing at some point exits the FlexDial Framework to complete the call. Table 3-1 shows the different scenarios for exiting the FlexDial Framework and how the switch can handle the call after exiting.

Table 3-1
Exit scenarios (continued)

Scenario	Call completion
<ul style="list-style-type: none"> <li data-bbox="233 422 870 485">• The switch sets PSIG (reached through a FlexDial collectable where at least one digit is collected) <li data-bbox="233 562 938 625">• The FlexDial Framework sets PDIL (reached through the ADDR/ADDRPARM collectable) 	<p data-bbox="1024 422 1404 548">Call processing continues to Info_Collected. If trigger criteria is not met, the switch applies PSIG.</p> <p data-bbox="1024 562 1404 751">Call processing continues through the call model. However, if trigger criteria is not met during Collect_Information or Analyze_Information, the switch applies PDIL.</p>
—end—	

Re-entry scenarios

When certain situations occur, call processing can re-enter the FlexDial Framework. There are two types of re-entry:

- Recollection (RECOL) – returns call to collectable processing within the FlexDial Framework at the point where the exit occurred (must follow a TERMINATE collectable)
- Dialplan index (DPIDX) – returns call to the specified index into table FLEXDIAL, thereby re-entering the FlexDial Framework

All re-entry scenarios occur after successful termination of the call to the terminating agent. During re-entry scenarios, the following NetworkBuilder messaging scenarios may occur:

- The switch may send the same message (at different times within the life of the call), containing the same data, to the SCP.
- The switch may receive the same **Analyze_Route** message (in response to the first bullet), containing the same or different data, which can cause a looping of data between the switch and the SCP.
- The SCP may receive messages out of the expected order. For example, a call triggers at *Customized_Dialing_Plan* and the switch sends an **Info_Analyzed** message to the SCP. The SCP responds with an **Analyze_Route** message. The switch attempts to route the call and a re-entry scenario occurs, moving the call to **Collect_Information**. The call then triggers at *Shared_Interoffice_Trunk* and the switch sends an **Info_Collected** message to the SCP. The SCP has received an **Info_Analyzed** followed by an **Info_Collected**.

Note: The UCS08 software release allows EDPs to be activated in calls originating on AXXESS agents and integrates EDPs into the FlexDial call model. For more information on EDPs, refer to the *UCS DMS-250 NetworkBuilder Application Guide* for more information.

In FlexDial calls, if EDPs are active when the processing of collectables is resumed (due to the “RECOL” or “DPIDX” actions), the switch deactivates EDPs and sends a `close` message to the SCP with `closeCause` value `EDPsCompleted`. This allows the UCS DMS-250 SSP to encounter TDPs after the collectable processing is completed. If route advancing (the “RTEADV” action) occurs, EDPs remain active. Refer to the *UCS DMS-250 FlexDial Framework Application Guide* for more information on the RECOL, DPIDX, and RTEADV actions.

Because the FLEXSIM tool does not simulate NetworkBuilder call processing after the **Analyze_Information** PIC, there is no interaction between FLEXSIM and EDPs.

Recollection

When the RECOL re-entry scenario occurs, call processing returns control of the call to the FlexDial framework. RECOL occurs in one of the following situations:

- the long call duration timer expires.
- the switch receives a Release With Cause message for an SS7 termination.
- a successful termination for a TERMINATE CONTINUE scenario occurs.

FlexDial continues processing the call at the point where the exit originally occurred. For example, following the TERMINATE collectable.

Provisioning

RECOL is provisioned as follows:

- table FLEXDIAL – datafill the TERMINATE collectable with STOP
Note: This re-entry scenario requires that the call exited the FlexDial Framework after execution of a TERMINATE collectable datafilled as STOP. This exit scenario identifies the point where the switch can return control of the call to the FlexDial Framework and resume collectable execution.
- table FLEXFEAT – datafill the NOANSDUR option with the RECOL action

- table CSEMAP – datafill RECOL=Y against a particular release cause

Implied RECOL is provisioned as follows:

- table FLEXDIAL – datafill the TERMINATE collectable with CONTINUE

Note: Refer to the FlexDial recollection rules in the *UCS DMS-250 FlexDial Framework Application Guide*.

Exceptions

The RECOL re-entry scenario cannot occur in the following situations:

- The FlexDial Framework has executed all collectables.
- NetworkBuilder takes control of the call at **O_Feature_Requested** as follows:
 - FlexDial executes an ADDR or ADDRPARAM collectable that results in the call encountering **O_Feature_Requested**. Trigger criteria is met and a the switch sends a query to the SCP. FlexDial does not allow the RECOL re-entry scenario, because the FlexDial Framework did not control the exit.
- FlexDial executes the APTRMT collectable with the delayed treatment set.
- The switch sets PDIL (reached through a non-ADDR/ADDRPARAM collectable).
- The switch sets PSIG (reached through a FlexDial collectable where at least one digit is collected).
- The FlexDial Framework sets PDIL (reached though the ADDR/ADDRPARAM collectable).

Dialplan index

When the DPIDX re-entry scenario occurs, call processing re-enters the FlexDial framework as directed by the datafilled DPIDX.

Provisioning

- long call duration
 - table FLEXFEAT – datafill the NOANSDUR option with the DPIDX action
- reorigination

- table FLEXFEAT – datafill the REORGACT and provide the appropriate INDEX into table FLEXDIAL. Refer to the FlexDial reorigination rules in the *UCS DMS-250 FlexDial Framework Application Guide*.

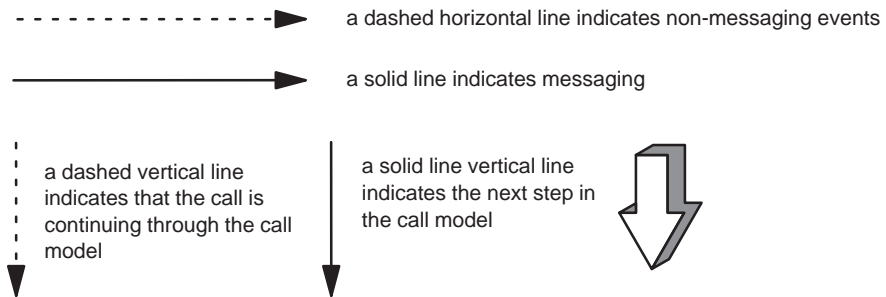
Note: Refer to the FlexDial recollection rules in the *UCS DMS-250 FlexDial Framework Application Guide*.

Exceptions

None

Examples

The following examples show the various re-entry scenarios and use the following conventions:



FlexDial-controlled reorigination

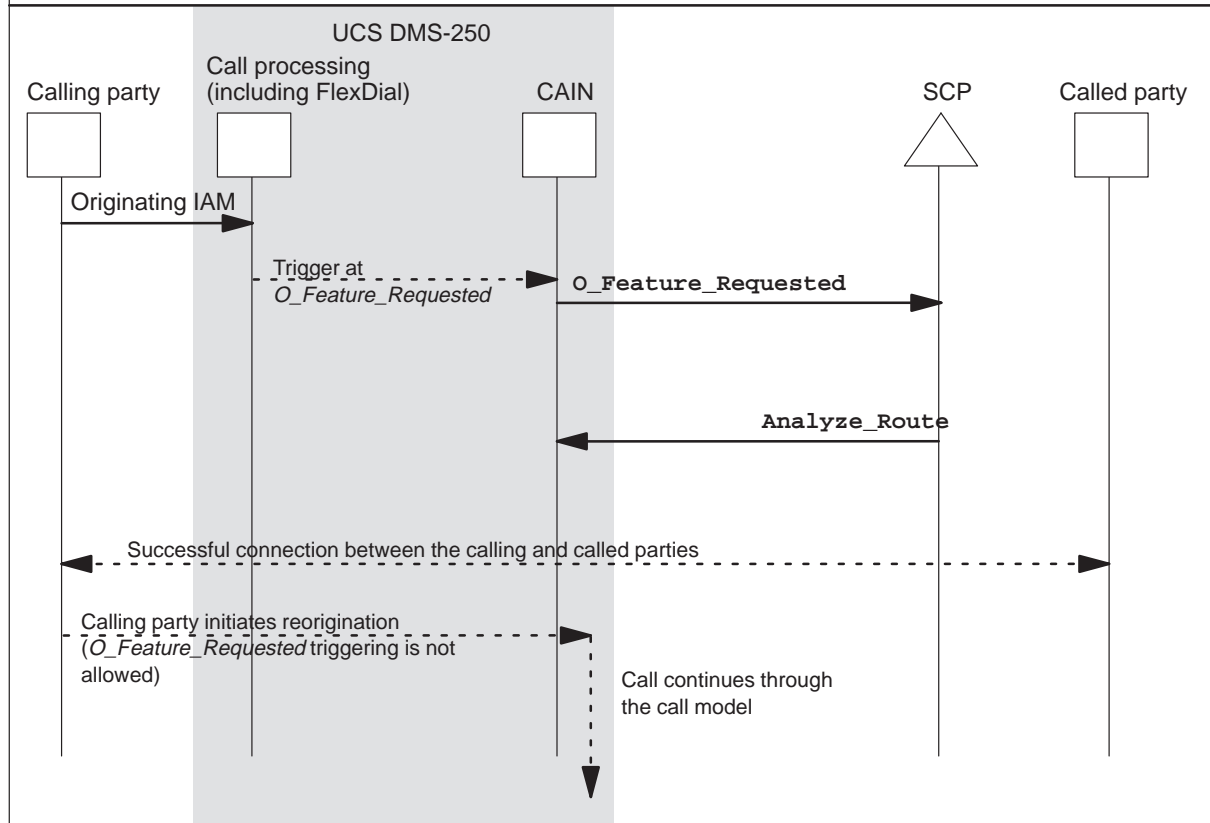
FlexDial Framework	CAIN Framework
EXIT CONDITION: Exited from ADDR or ADDRPARAM collectable execution due to O_Feature_Requested interactions.	HISTORY: Trigger criteria met at O_Feature_Requested with TRIGACT=FEAT Note: The REORIG option is not provisioned against the OFTRREQ tuple associated with the call.

Re-entry scenario: FlexDial-controlled reorigination
Related datafill:

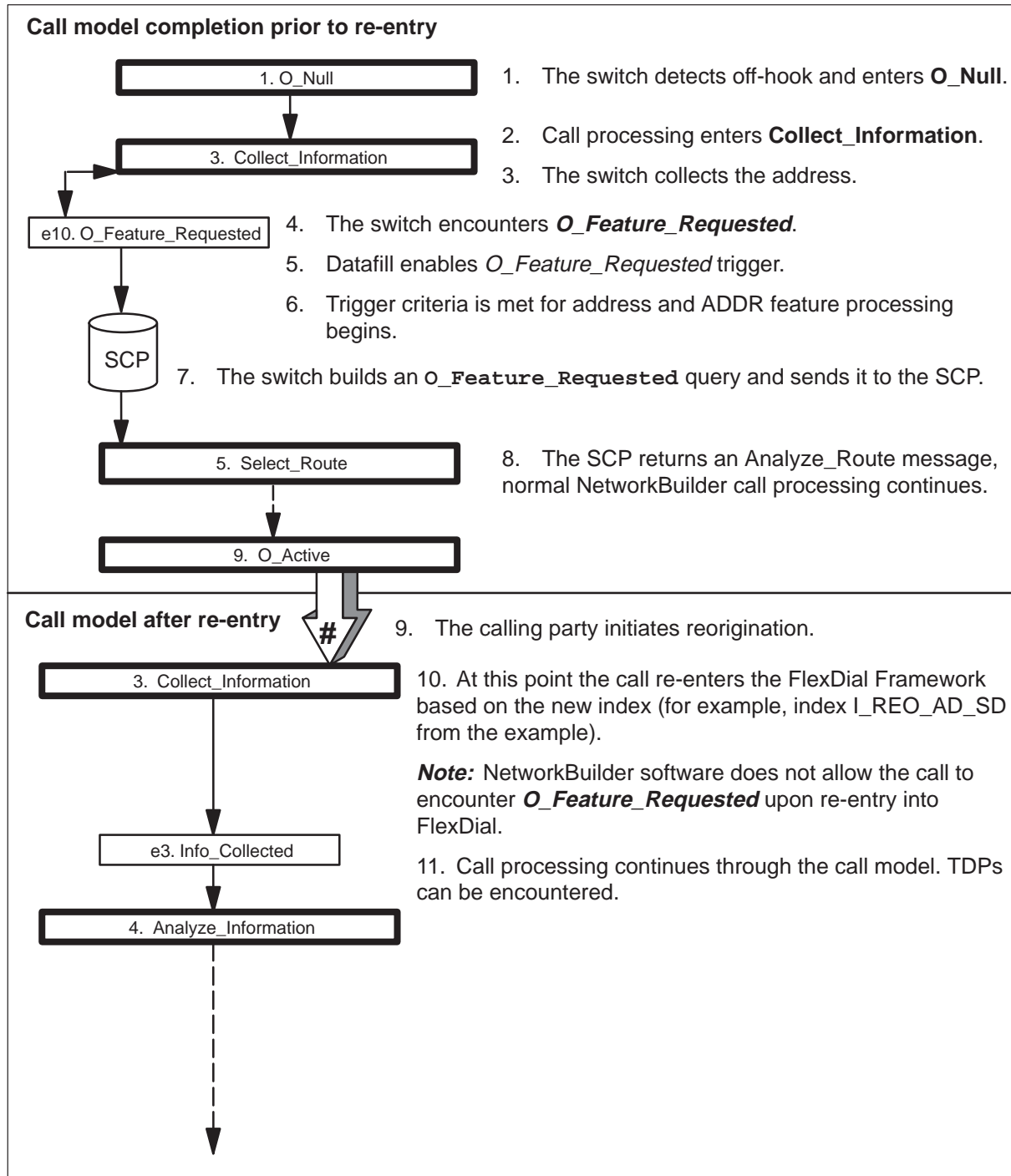
Table FLEXFEAT

KEY	OPTIONS
110111	1501 (DPIDX OFTRIDX APPEND) (REORGACT I_REO_AD_SD 0 15) (TRANSTS PARTITION 111 0) (PVSPDIDX 51) (CITYCODE 111) (CAINGRP AXXGRP) (REORGTYP ONKEY STR P 15 5) \$

Note: For reorigination to be allowed on the call, the REORGACT and REORGTYP options must be datafilled in table FLEXFEAT. The REORGACT option specifies the actions to take upon reorigination and identifies a tuple in table FLEXDIAL, an index into table MSGCTR, and a disconnect timer. The REORGTYP option specifies the reorigination activation type. Refer to the *UCS DMS-250 FlexDial Framework Application Guide* for more information.



FlexDial-controlled reorigination (cont)



OFTRREQ-controlled reorigination with O_Feature_Collected ADDR collectable

FlexDial Framework	CAIN Framework
EXIT CONDITION: Exited from ADDR or ADDRPARAM collectable execution due to O_Feature_Requested interactions.	HISTORY: Trigger criteria met at O_Feature_Requested with TRIGACT=FEAT and the REORIG option datafilled Note: The ADDR collectable is being used with a feature processor (for example, AUTH or CARD).

Re-entry scenario: OFTRREQ-controlled reorigination
Related datafill:

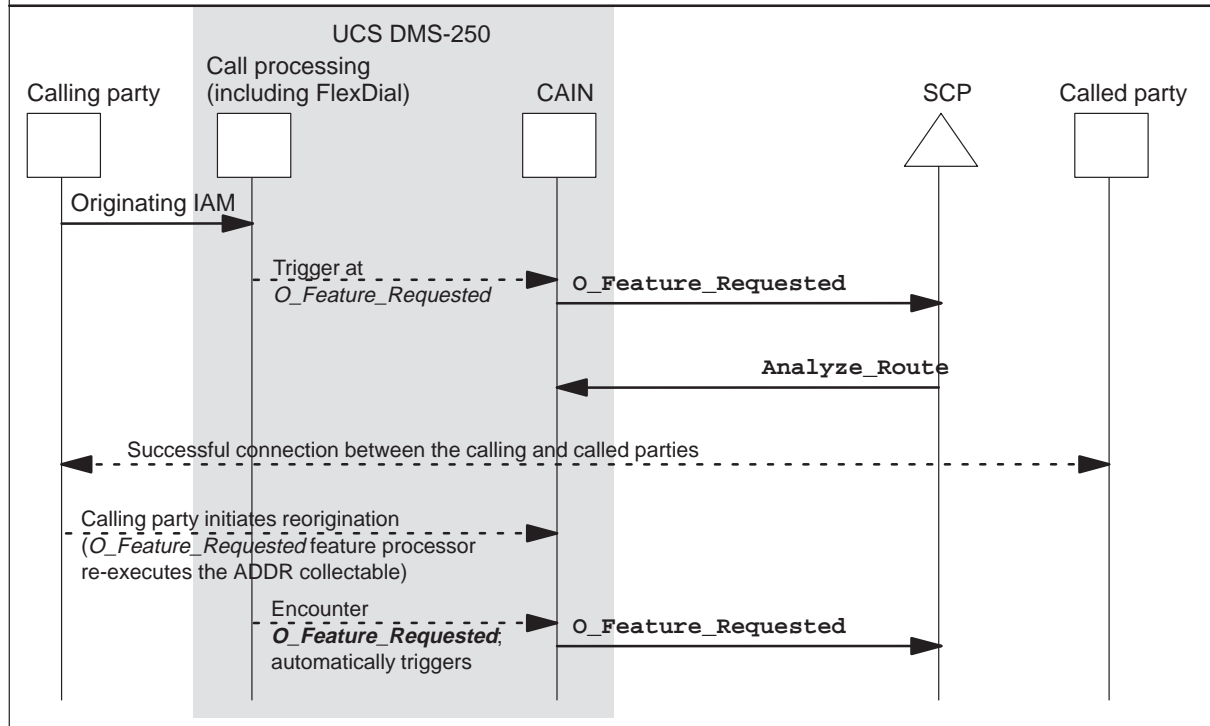
Table FLEXFEAT

KEY	OPTIONS
110111	1501 (DPIDX OFTRIDX APPEND) (REORGACT I_REO_AD_SD 0 15) (TRANSTS PARTITION 111 0) (PVSPDIDX 51) (CITYCODE 111) (CAINGRP AXXGRP) (REORGTYP ONKEY STR P 15 5) \$

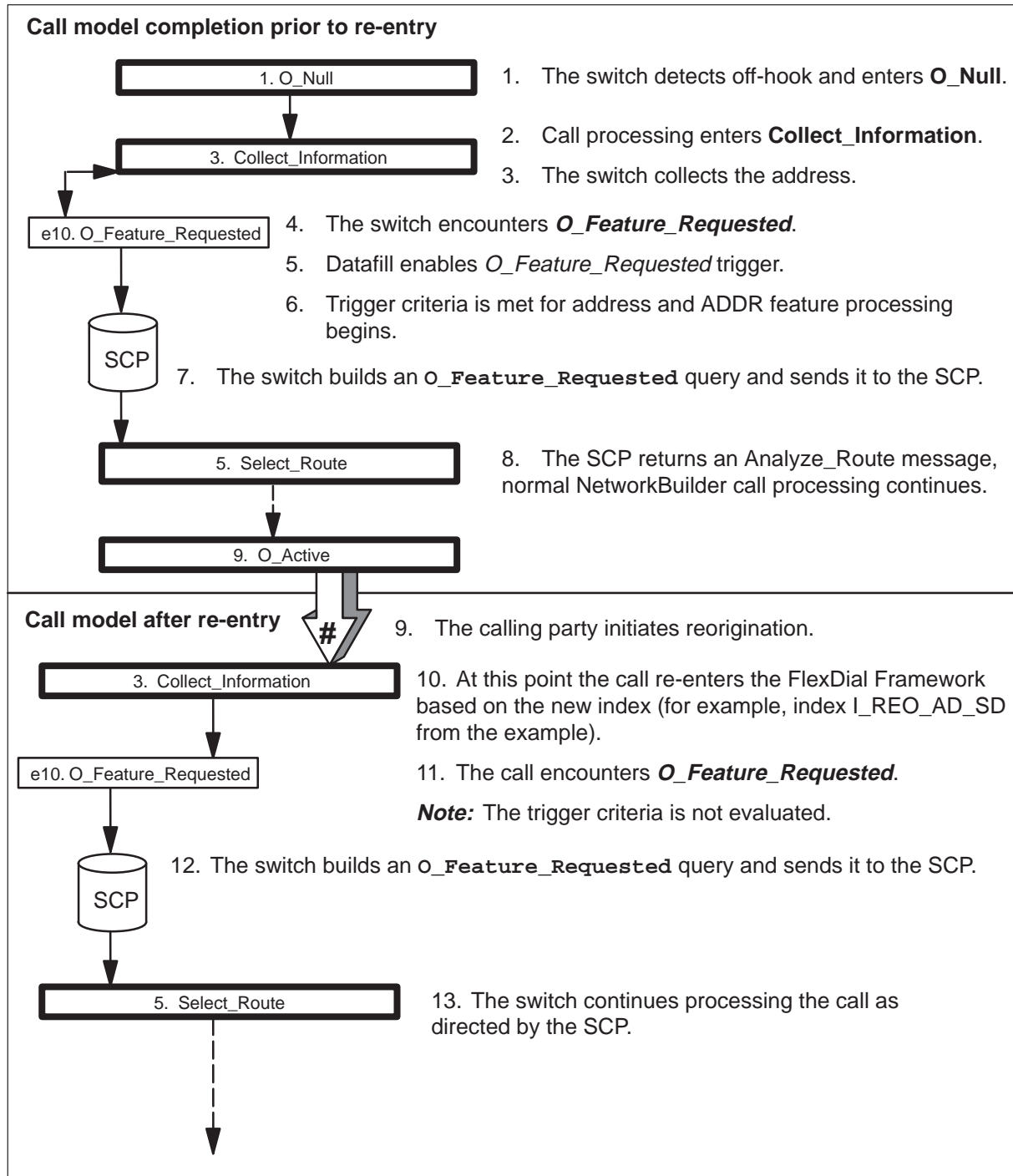
Table OFTRREQ

OFTRREQKY	TODIGS	ACTION	OPTIONS
AXXGRP AUTH 214	214	FEAT AUTH (AUTH 8 N 7 7 5)(ADDR 10 N 10 10 5) \$	REORIG \$

Note: For reorigination to be allowed on the call, the REORGACT and REORGTYP options must be datafilled in table FLEXFEAT. The REORGACT option specifies the actions to take upon reorigination and identifies a tuple in table FLEXDIAL, an index into table MSGCTR, and a disconnect timer. The REORGTYP option specifies the reorigination activation type. Refer to the *UCS DMS-250 FlexDial Framework Application Guide* for more information.



OFTRREQ-controlled reorigination with O_Feature_Collected ADDR collectable (cont)



Multiply-provisioned reorigination; O_Feature_Requested re-encountered

FlexDial Framework	CAIN Framework
EXIT CONDITION: Exited from ADDR or ADDRPARAM collectable execution due to O_Feature_Requested interactions.	HISTORY: Trigger criteria met at O_Feature_Requested with TRIGACT=FEAT
Reorigination behavior: ADDR or ADDRPARAM collectables executed as datafilled.	REORIG option is datafilled The OFTRREQ ADDR collectable is not provisioned.

Re-entry scenario: Multiply-provisioned reorigination

Related datafill:

Table	KEY	OPTIONS
FLEXFEAT	110111	1501 (DPIDX OFTRIDX APPEND) (REORGACT I_REO_AD_SD 0 15) (TRANSTS PARTITION 111 0) (PVSPDIDX 51) (CITYCODE 111) (CAINGRP AXXGRP) (REORGTYP ONKEY STR P 15 5) \$

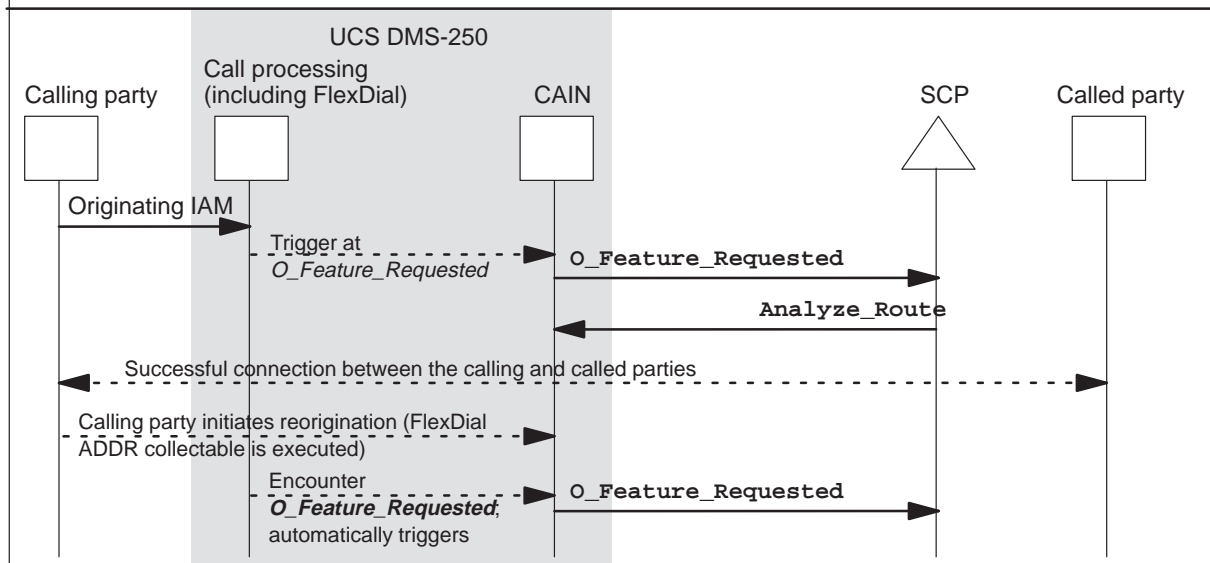
Table FLEXDIAL

KEY	DATA	NEXT_INSTR
I_REO_AD_SD	(INCLUDE I_SIG_DTMF) \$	Y C_ADDR_SD
C_ADDR_SD	(ADDR 1 18 (SIGFEAT 2 0 2) (PROMPT TONE STD SD 255 N) \$ (RESET S 255 RODR PREPRMPT PREPRMPT \$) (VALIDATE INSWITCH NPRT IGNORE) \$) \$	N

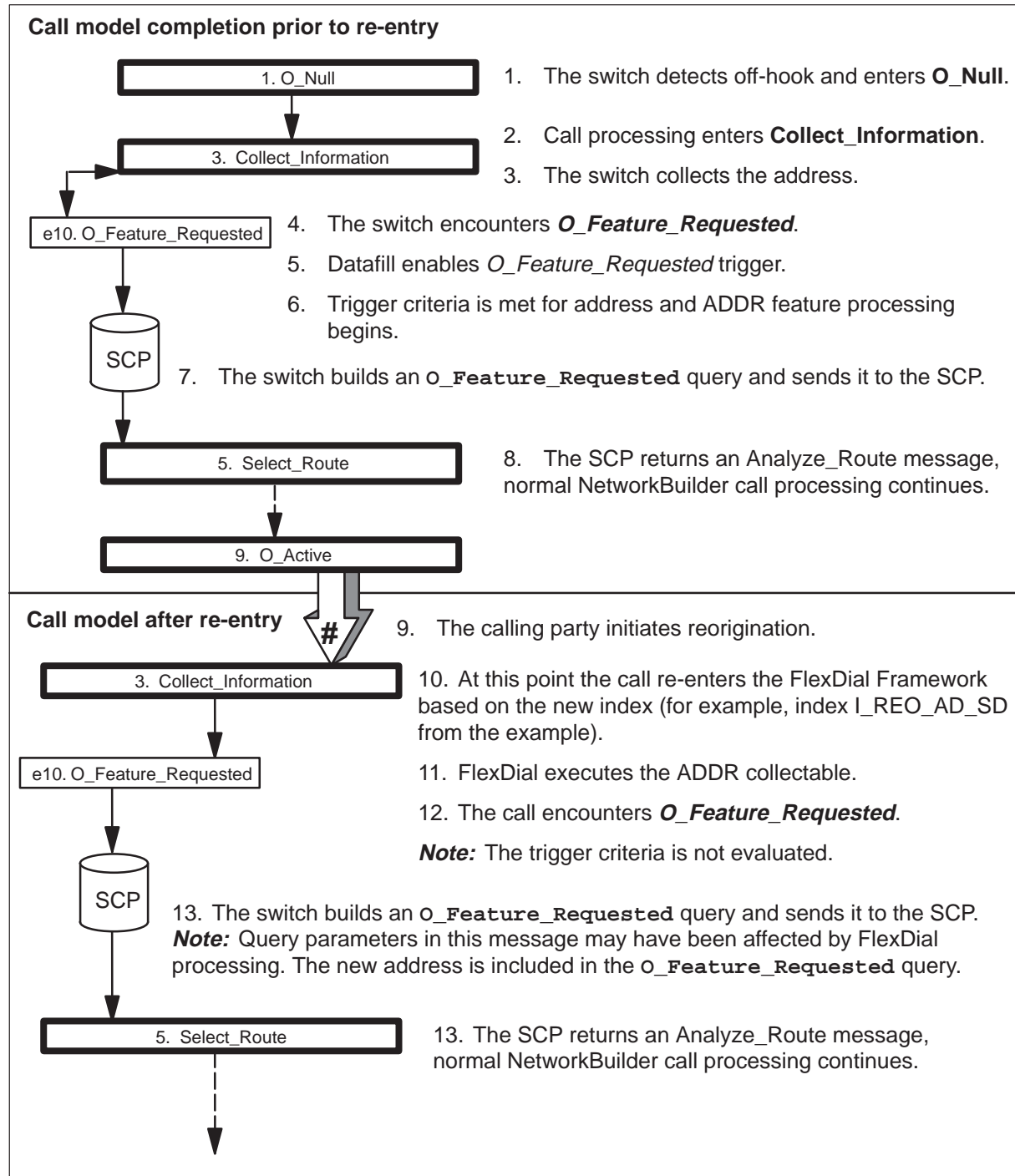
Table	OFTRREQKY	TODIGS	ACTION	OPTIONS
OFTRREQ	AXXGRP AUTH 214	214	FEAT ADDR \$	REORIG \$

Note: The ADDR in the OFTRREQ tuple is the ADDR processor, not the ADDR collectable.

Note: For reorigination to be allowed on the call, the REORGACT and REORGTYP options must be datafilled in table FLEXFEAT. The REORGACT option specifies the actions to take upon reorigination and identifies a tuple in table FLEXDIAL, an index into table MSGCTR, and a disconnect timer. The REORGTYP option specifies the reorigination activation type. Refer to the *UCS DMS-250 FlexDial Framework Application Guide* for more information.



Multiply-provisioned reorigination; O_Feature_Requested re-encountered (cont)



Multiply-provisioned reorigination; O_Feature_Requested is not encountered

FlexDial Framework	CAIN Framework
EXIT CONDITION: Exited from ADDR or ADDRPARM collectable execution due to O_Feature_Requested interactions.	HISTORY: Trigger criteria met at O_Feature_Requested with TRIGACT=FEAT and the REORIG option datafilled
Reorigination behavior: ADDR collectable is not executed. (For example, a ROUTE collectable may be executed).	REORIG option is datafilled
	The OFTRREQ ADDR collectable is not provisioned.

Re-entry scenario: OFTRREQ-controlled reorigination
 Related datafill:

Table FLEXFEAT	KEY	OPTIONS
	110111	1501 (DPIDX OFTRIDX APPEND) (REORGACT I_REO_ROUTE 0 15) (TRANSTS PARTITION 111 0) (PVSPDIDX 51) (CITYCODE 111) (CAINGRP AXXGRP) (REORGTYP ONKEY STR P 15 5) \$

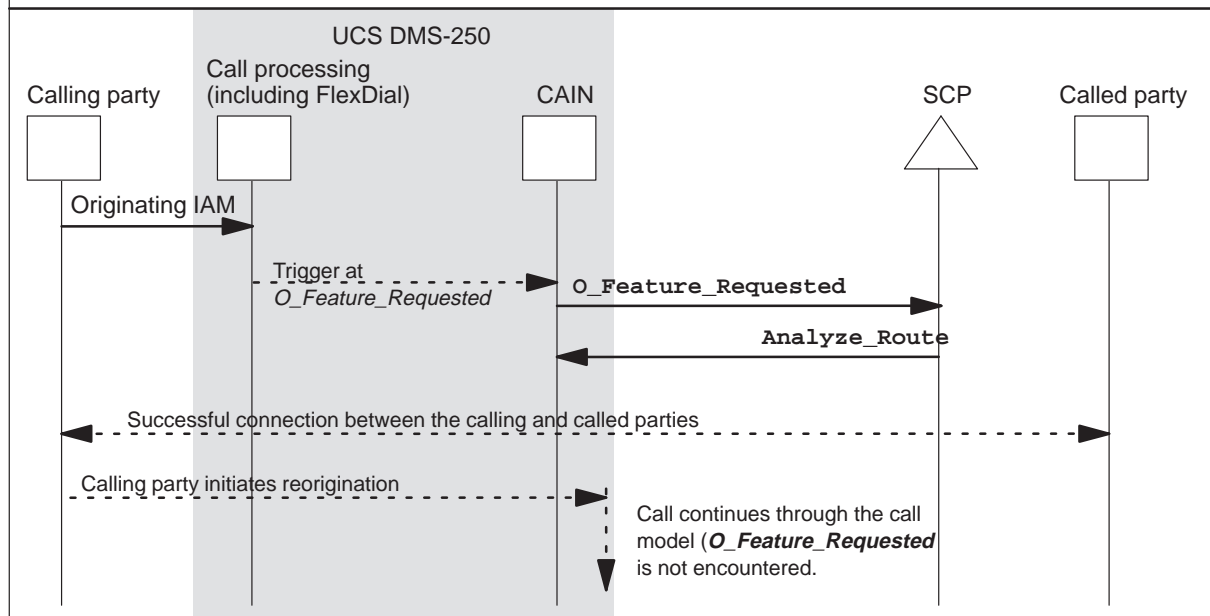
Table FLEXDIAL

KEY	DATA	NEXT_INSTR
I_REO_ROUTE	(ROUTE S D DAL214TWDTLS) \$	N

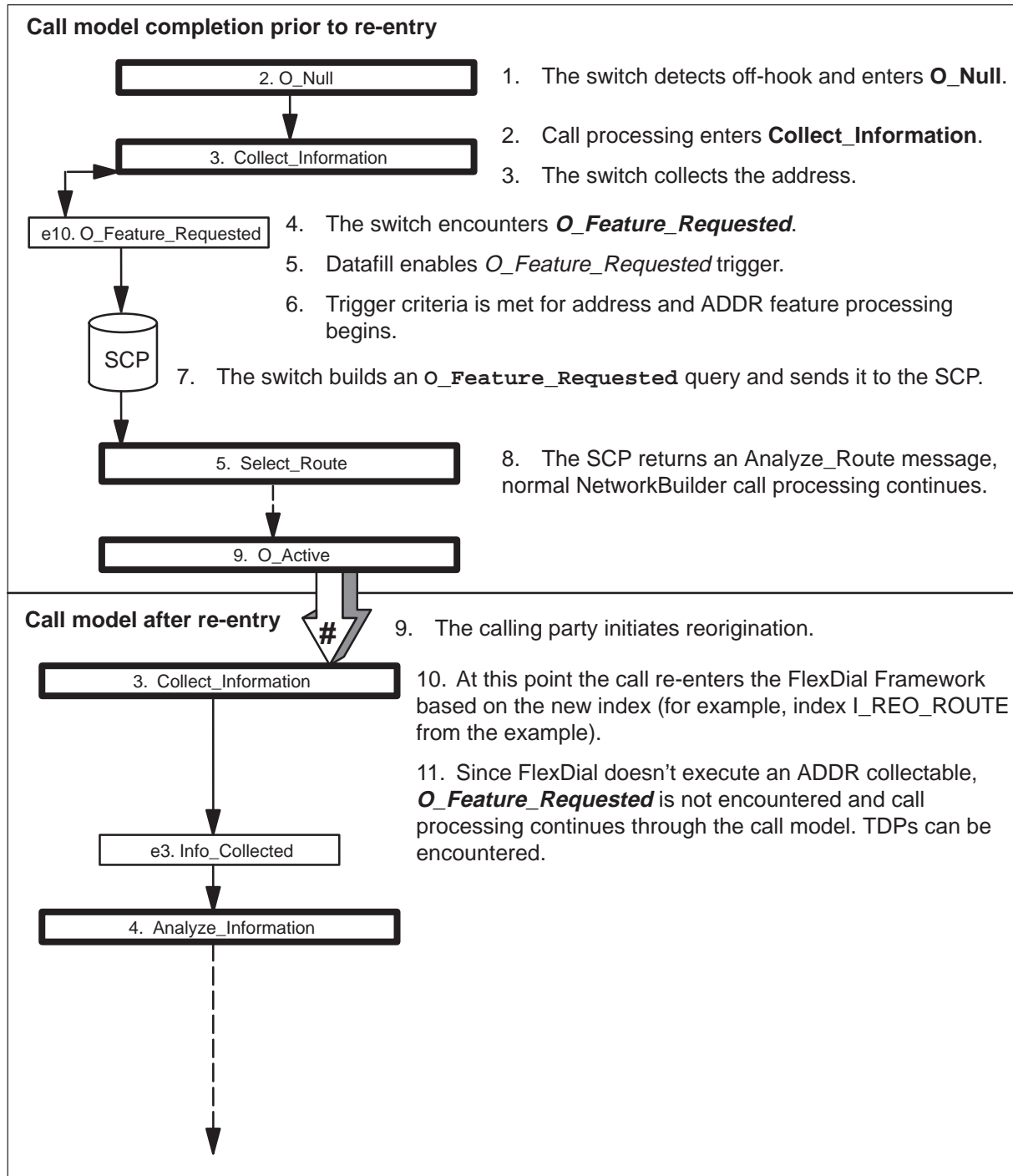
Table OFTRREQ	OFTRREQKY	TODIGS	ACTION	OPTIONS
	AXXGRP AUTH 214	214	FEAT ADDR \$	REORIG \$

Note: The ADDR in the OFTRREQ tuple is the ADDR processor, not the ADDR collectable.

Note: For reorigination to be allowed on the call, the REORGACT and REORGTYP options must be datafilled in table FLEXFEAT. The REORGACT option specifies the actions to take upon reorigination and identifies a tuple in table FLEXDIAL, an index into table MSGCTR, and a disconnect timer. The REORGTYP option specifies the reorigination activation type. Refer to the UCS DMS-250 FlexDial Framework Application Guide for more information.



Multiply-provisioned reorigination; O_Feature_Requested is not encountered (cont)



NonOFTRREQ-related reoriginations (normal reorigination)

FlexDial Framework	CAIN Framework
EXIT CONDITION: Exited due to collectable exhaustion or TERMINATE collectable.	HISTORY: Normal CAIN/FlexDial interactions throughout the progression of the call model.

Re-entry scenario: Multiply-provisioned reorigination

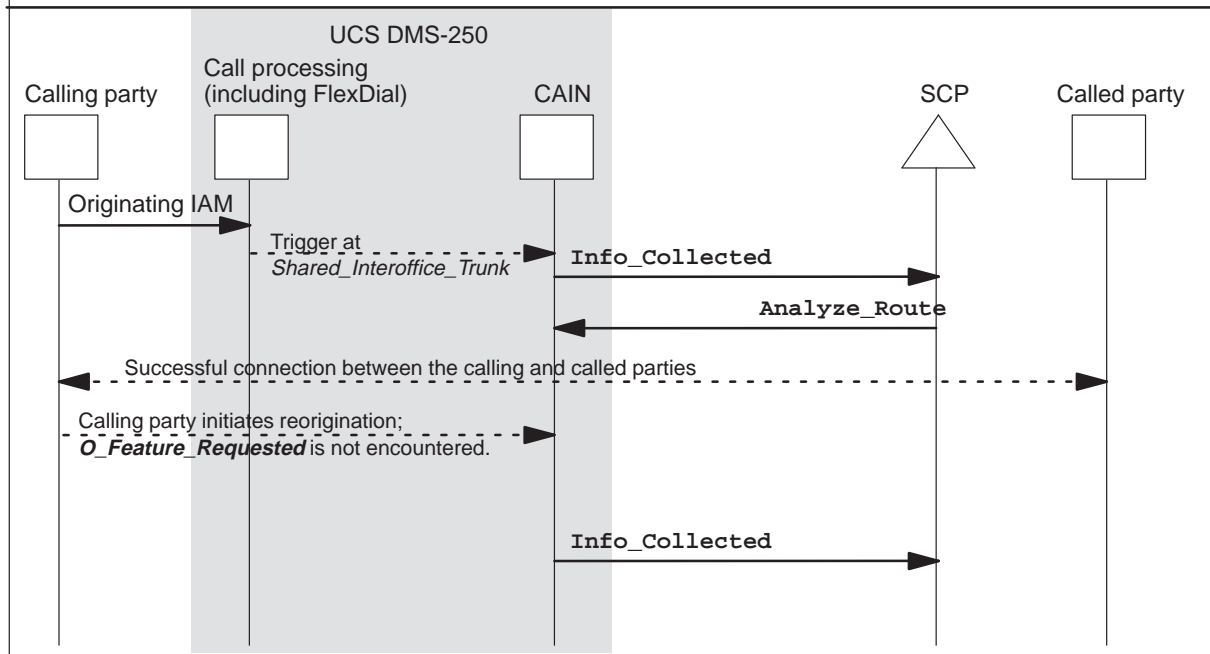
Related datafill:

Table	KEY	OPTIONS
FLEXFEAT	110111	1501 (DPIDX OFTRIDX APPEND) (REORGACT I_REO_AD_SD 0 15) (TRANSTS PARTITION 111 0) (PVSPDIDX 51) (CITYCODE 111) (CAINGRP AXXGRP) (REORGTYP ONKEY STR P 15 5) \$

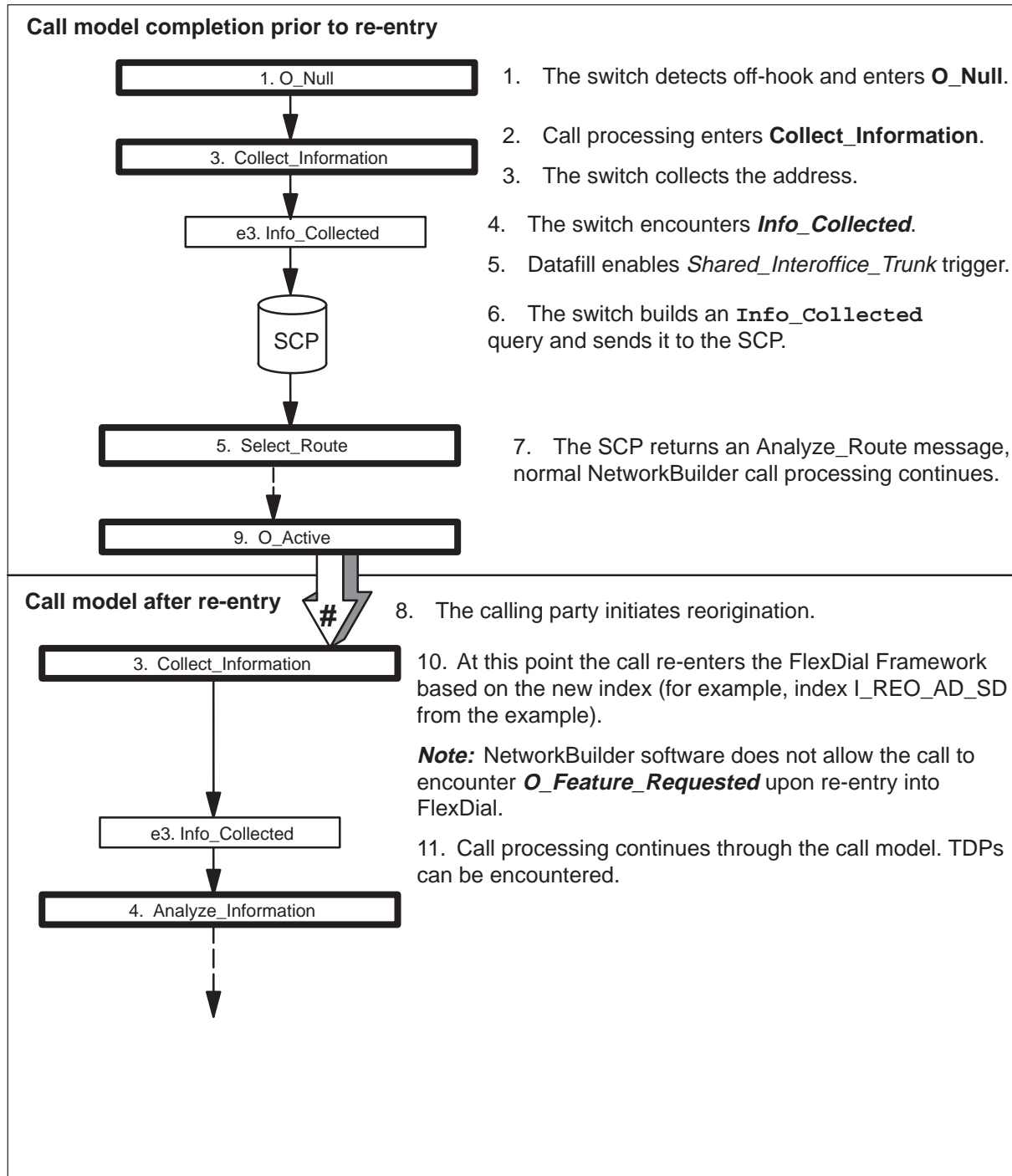
Table FLEXDIAL

KEY	DATA	NEXT_INSTR
I_REO_AD_SD	(INCLUDE I_SIG_DTMF) \$	Y C_ADDR_SD
C_ADDR_SD	(ADDR 1 18 (SIGFEAT 2 0 2) (PROMPT TONE STD SD 255 N) \$ (RESET S 255 RODR PREPRMPT PREPRMPT \$) (VALIDATE INSWITCH NPRT IGNORE) \$) \$	N

Note: For reorigination to be allowed on the call, the REORGACT and REORGTYP options must be datafilled in table FLEXFEAT. The REORGACT option specifies the actions to take upon reorigination and identifies a tuple in table FLEXDIAL, an index into table MSGCTR, and a disconnect timer. The REORGTYP option specifies the reorigination activation type. Refer to the *UCS DMS-250 FlexDial Framework Application Guide* for more information.



NonOFTRREQ-related reoriginations (normal reorigination) (cont)



RECOL re-entry based on TERMINATE with STOP with long call duration

FlexDial Framework	CAIN Framework
EXIT CONDITION: TERMINATE STOP collectable executed.	HISTORY: Trigger criteria met at Info_Collected or Info_Analyzed.
	Note: Triggering at O_Feature_Requested (with an action of FEAT) prevents re-entry into the FlexDial Framework.

Re-entry scenario: RECOL re-entry based on TERMINATE with STOP

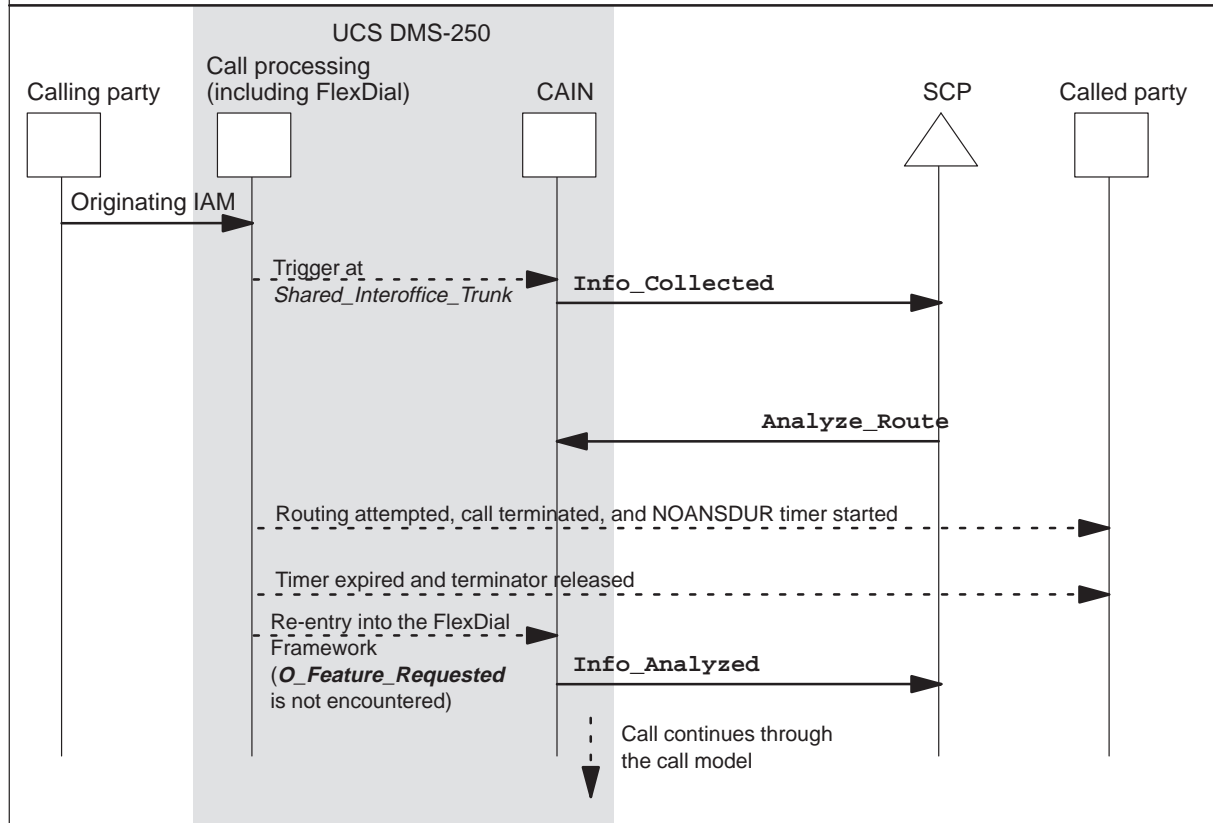
Related datafill:

Table TRKGRP

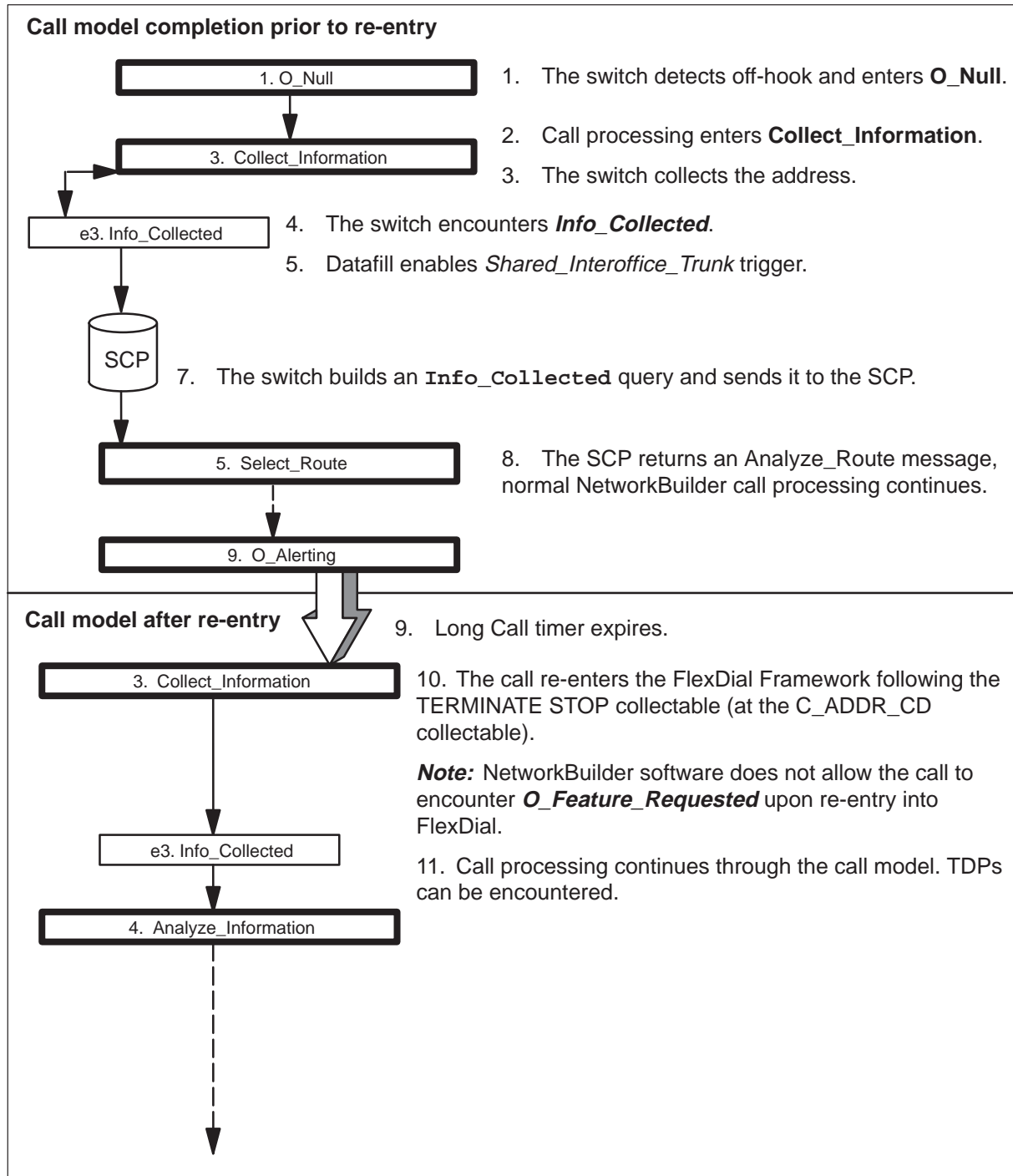
KEY
AXDAL273TWDTLS AXCESS 30 NPDGP MIDL FXS LS DT LSDT_0004 TRKFEAT_1503 (SNPA_SNXX_ANI) (I_AU_SD_AD) (TERM_STOP) \$ DAL

Table FLEXDIAL

KEY	DATA	NEXT_INSTR
TERM_STOP	(TERMINATE STOP \$) (INCLUDE C_ADDR_CD\$)	N



RECOL re-entry based on TERMINATE with STOP with long call duration (cont)



RECOL re-entry based on TERMINATE with STOP with CSEMAP

FlexDial Framework	CAIN Framework
EXIT CONDITION: TERMINATE STOP collectable executed.	HISTORY: Trigger criteria met at Info_Collected or Info_Analyzed. Note: Triggering at O_Feature_Requested (with an action of FEAT) prevents re-entry into the FlexDial Framework.

Re-entry scenario: RECOL re-entry based on TERMINATE with STOP with CSEMAP

Related datafill:

Table TRKGRP

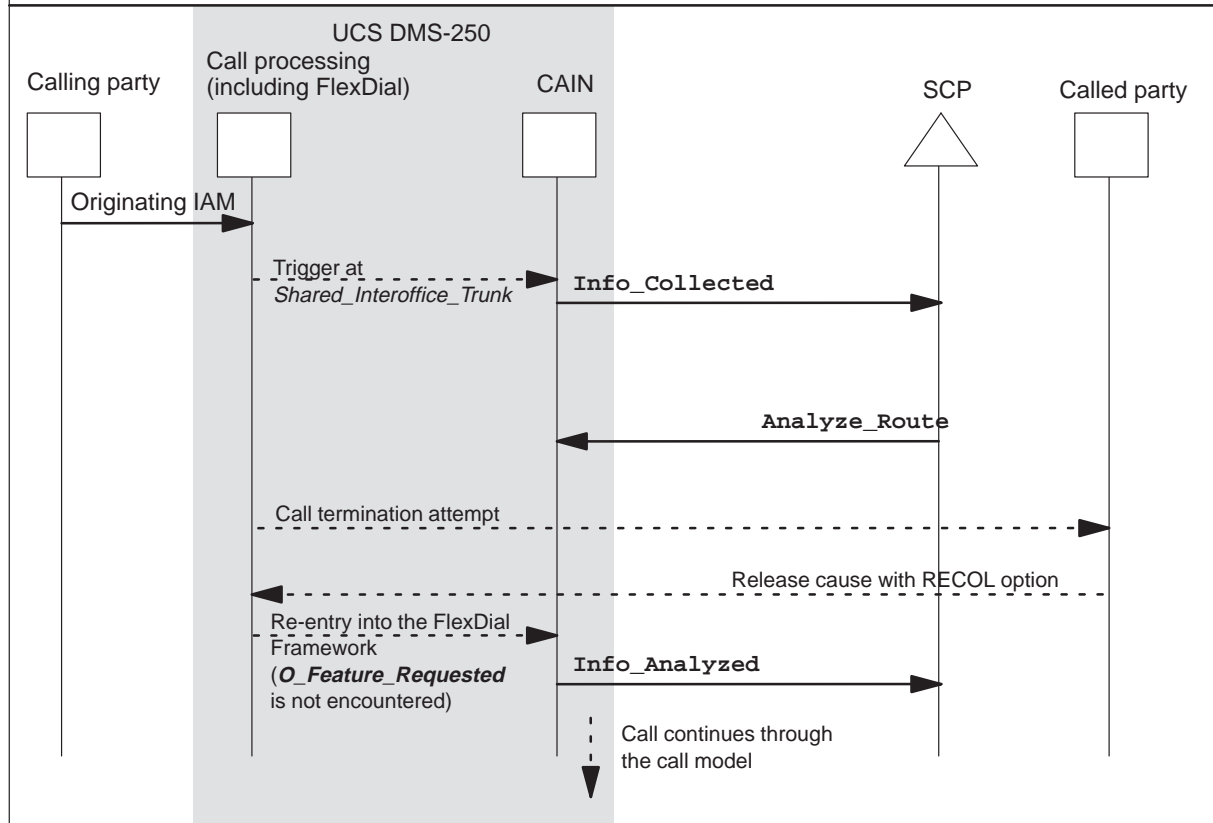
AXDAL273TWDTLS AXCESS 30 NPDGP MIDL FXS LS DT LSDT_0004
TRKFEAT_1503 (SNPA_SNX_ANI) (I_AU_SD_AD) (TERM_STOP) \$ DAL

Table FLEXDIAL

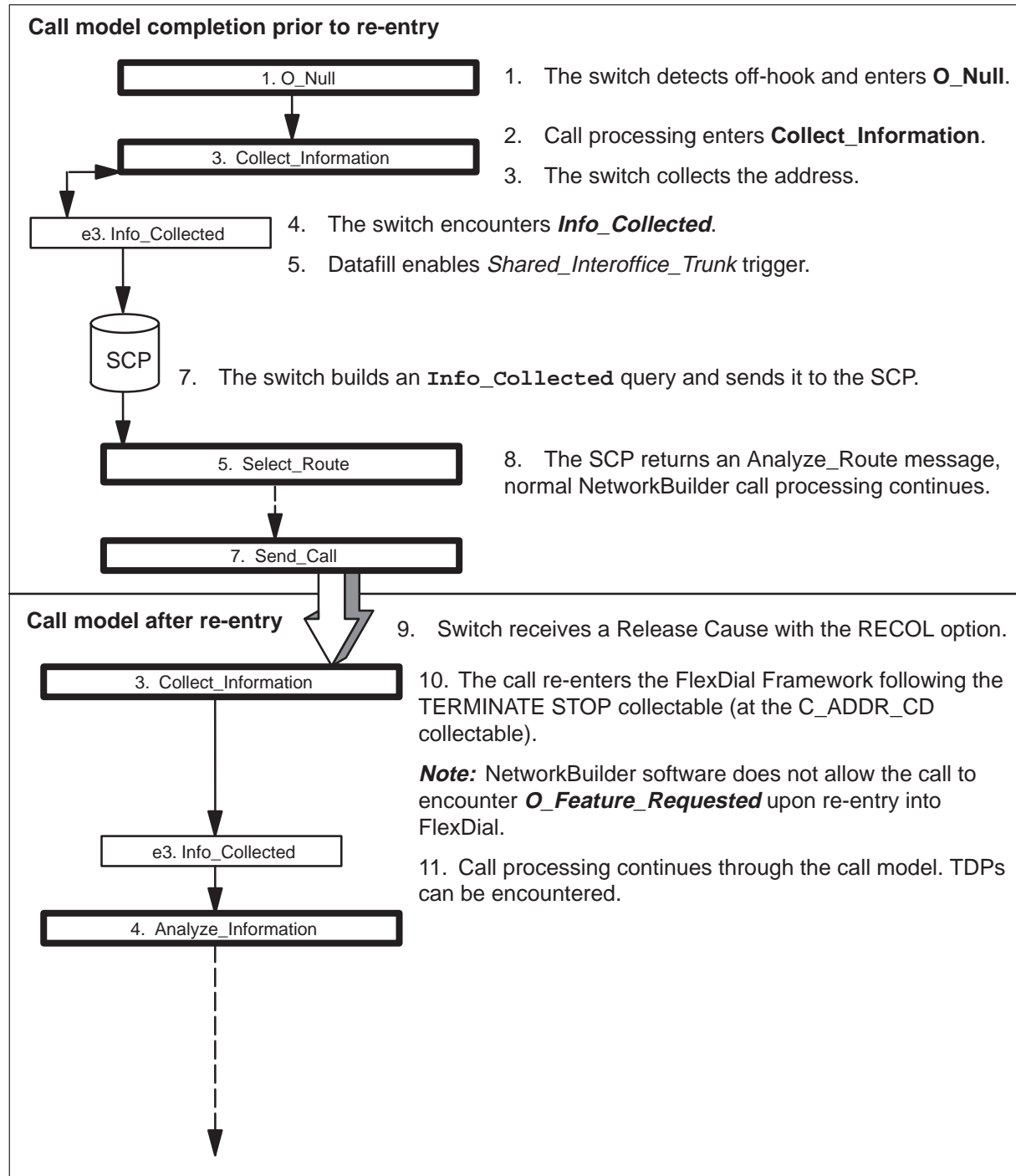
KEY	DATA	NEXT_INSTR
TERM_STOP	(TERMINATE STOP \$) (INCLUDE C_ADDR_CD\$)	N

Table CSEMAP

Q764 USERBUSY GNCT N Y \$



RECOL re-entry based on TERMINATE with STOP with CSEMAP (cont)



DPIDX re-entry based on long call duration

FlexDial Framework	CAIN Framework
EXIT CONDITION: TERMINATE STOP collectable executed.	HISTORY: Trigger criteria met at Info_Collected or Info_Analyzed. Note: Triggering at O_Feature_Requested (with an action of FEAT) prevents re-entry into the FlexDial Framework.

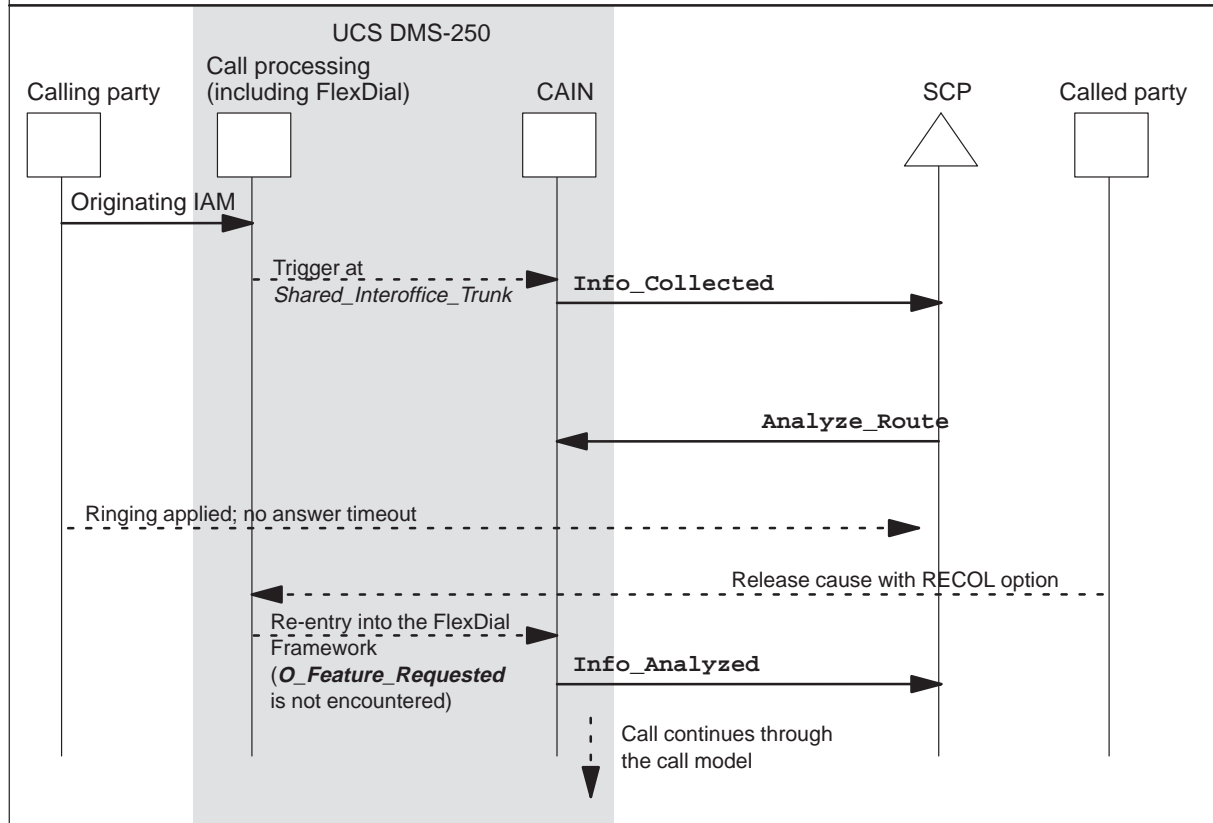
Re-entry scenario: RECOL re-entry based on long call duration
Related datafill:

Table TRKGRP

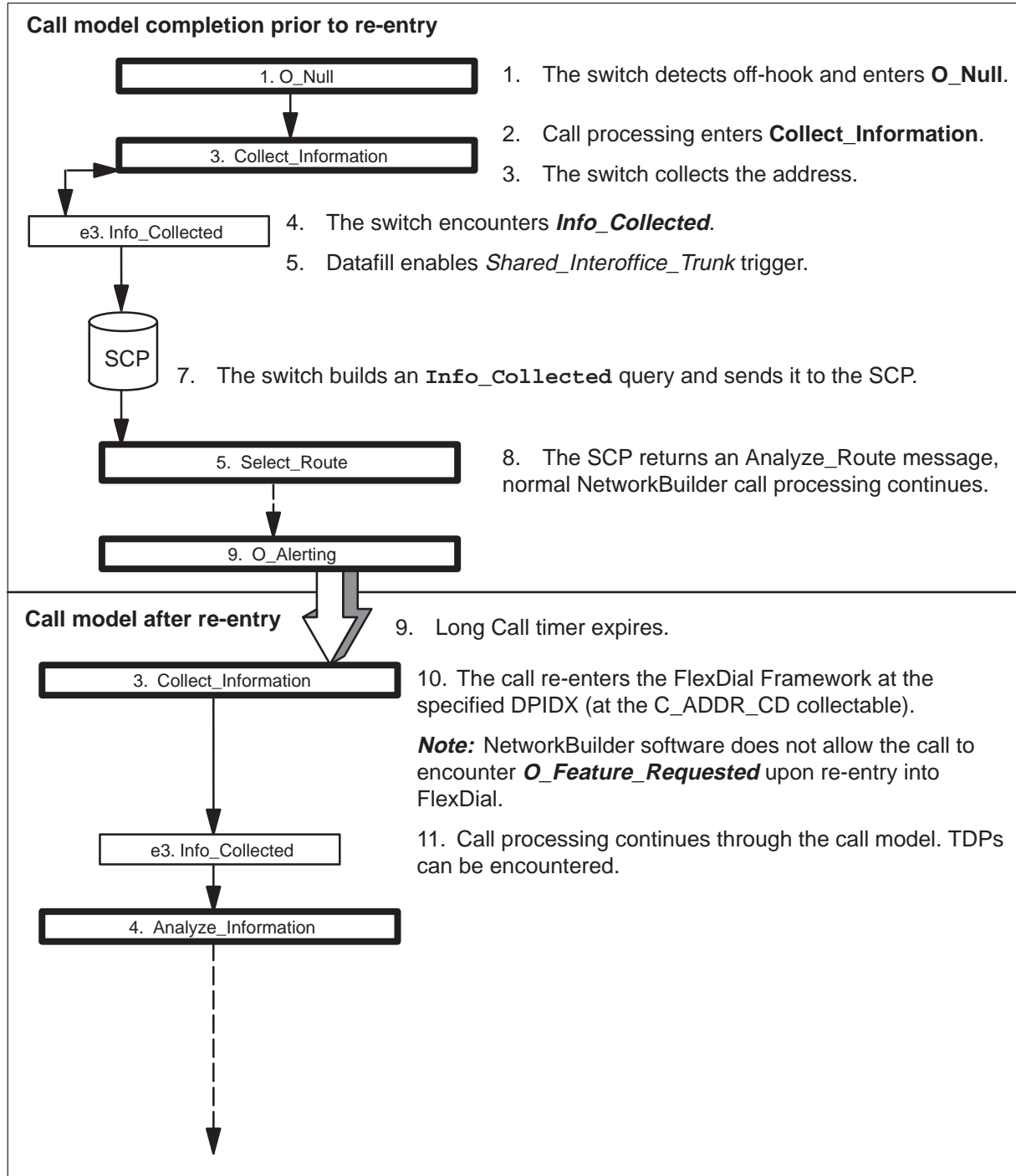
AXDAL273TWDTLS AXCESS 30 NPDGP MIDL FXS LS DT LSDT_0004
TRKFEAT_1503 (SNPA_SNXX_ANI) (L_AU_SD_AD) (TERM_STOP) \$ DAL

Table TRKFEAT

TRKFEAT_1503 \$ (NOANSDUR 5 DPIDX C_ADDR_CD) \$



DPIDX re-entry based on long call duration (cont)



Implied RECOL re-entry based on TERMINATE with CONTINUE

FlexDial Framework	CAIN Framework
EXIT CONDITION: TERMINATE CONTINUE collectable executed.	HISTORY: Trigger criteria met Info_Collected
	Note: Triggering at O_Feature_Requested (with an action of FEAT) prevents re-entry into the FlexDial Framework.

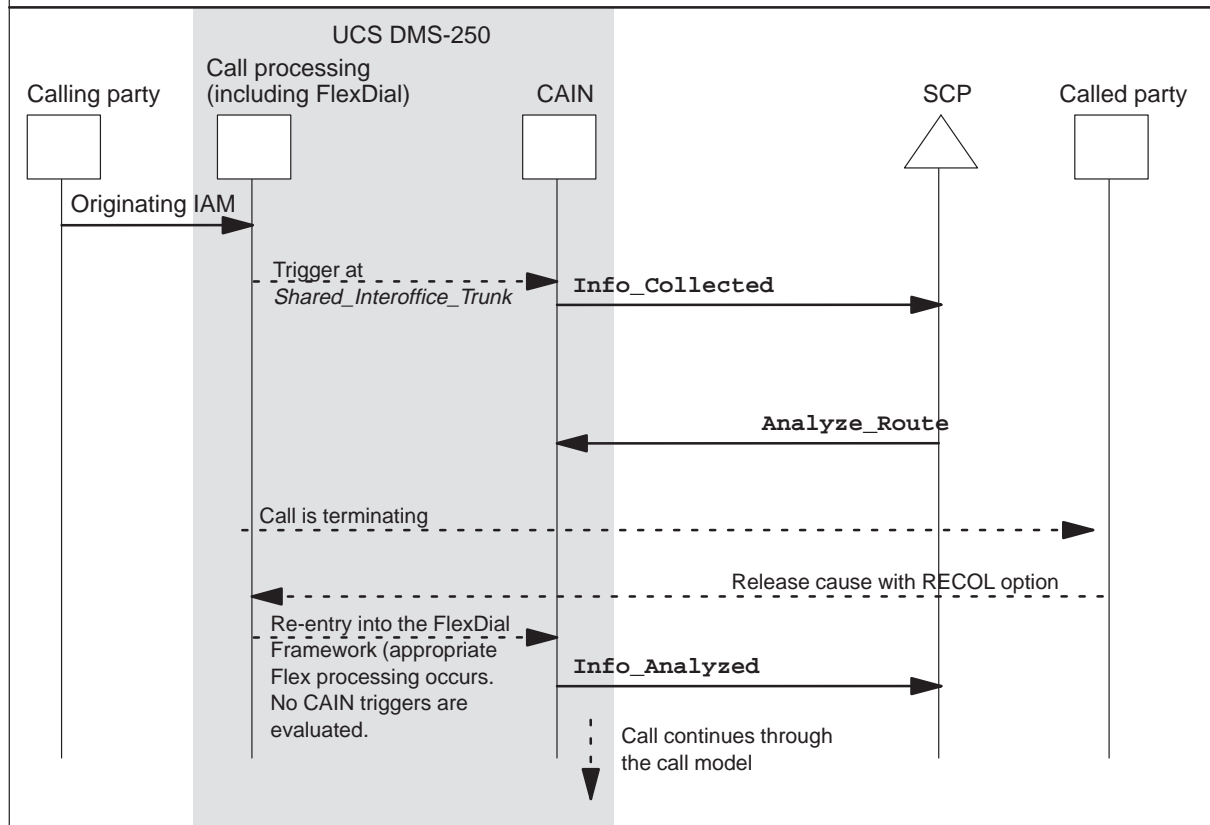
Re-entry scenario: Implied RECOL re-entry based on TERMINATE with CONTINUE
 Related datafill:

Table TRKGRP

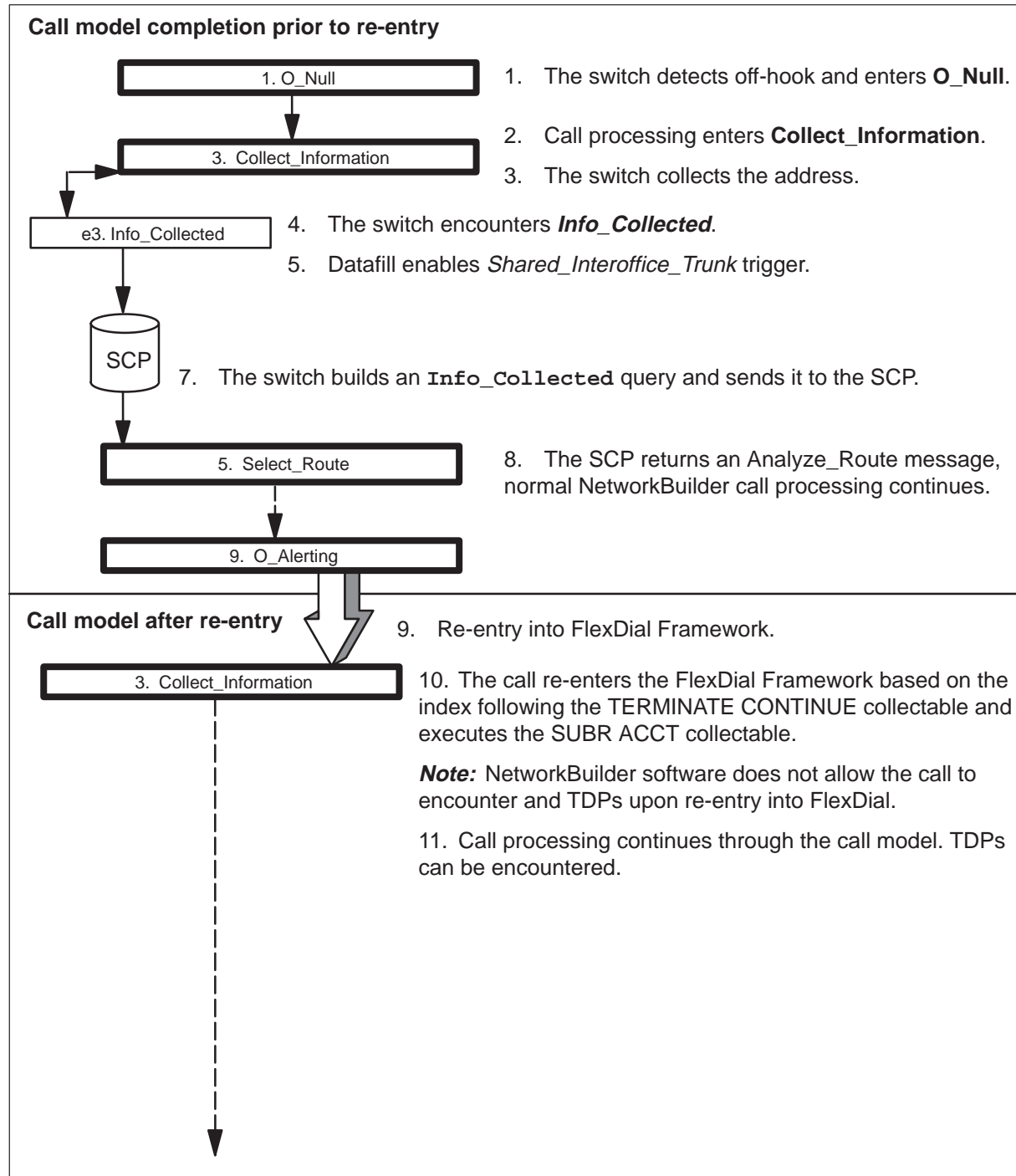
AXDAL271TWDTLS 30 NPDGP NCIT MIDL FXS LSDT_0004
 TRKFEAT_0099 (SNPA_SNXX_ANI) (I_AU_SD_AD) (TERM_CONT) \$ DAL

Table FLEXDIAL

KEY	DATA	NEXT_INSTR
TERM_CONT	(TERMINATE CONTINUE \$) (SUBR 5 5 ACCT N N \$ (FILED PREFIX 12345) (VALIDATE INSWITCH FLEXVAL (1 5) \$ IGNORE) \$) \$	N



Implied RECOL re-entry based on TERMINATE with CONTINUE (cont)



TCAP messaging

For information on TCAP messaging, refer to the *UCS DMS-250 NetworkBuilder Application Guide*.

The following limitation on FlexDial PANI construction applies to CAIN/FlexDial TCAP messaging:

- The SCP must be able to process this PANI (delivered in the *CallingPartyID* parameter)
- If the PANI is not suitable for global title translations, provision CAIN_ADDR_GT, CAIN_FEAT_GT or CAIN_OFCD_GT for global title translations.

Note: The FlexDial Framework does not differentiate between ANIs and PANIs. Therefore, if the CAIN_CLID_GT option is used for global title translations with an invalid national number, global title translations will fail and the query is not delivered to the SCP.

Outgoing message parameters

ATTENTION

Outgoing messages require the CAIN0100 SOC option; all extension parameters (outgoing and incoming messages) require the CAIN0200 SOC option. Refer to the *UCS DMS-250 Software Optionality Control User's Manual* for more information about SOCs.

Once trigger criteria is met and an extension block is allocated, the appropriate TCAP outgoing message is built and sent to the SCP.

For more detailed information on the outgoing messages and their valid responses, refer to the *UCS DMS-250 NetworkBuilder Application Guide*.

CAIN nature of address determination

The following table describes the determination of the CAIN nature of address (NOA) used in building queries to the SCP.

Table 5-1
NOA determination

Parameter	NOA
<i>AccessCode</i>	Always set to ACCT. Any NOA received from an IAM is ignored. These digits and NOA are determined through the use of the ACCT CAIN flag.
<i>CalledPartyID</i>	Based on pretranslation results of ADDR and ADDRPARM collectables. Valid NOAs: INTL, HOTL, NATL, N00, or VPN
—continued—	

5-2 Outgoing message parameters

Table 5-1
NOA determination

Parameter	NOA
<i>CallingPartyID</i>	For inband (PTS) calls: The NOA is always NATL. These digits are collected by the SUBR collectable. The last digits collected by the SUBRPARAM collectable where the corresponding FLEXTYPE tuple has the CALLING or CAINFLG CLGPTYADD option; the NOA is taken from the IAM parameter provisioned in the SUBRPARAM collectable.
<i>ChargeNumber</i>	See table 5-6
<i>CollectedDigits</i>	Always set to PIN. Any NOA received from an IAM is ignored. These digits and NOA are determined through the use of the PIN CAIN flag.
<i>ExtensionParameter</i> universalAccess	Always set to UNK.
<i>VerticalServiceCode</i>	Always set to HOTL.
—end—	

Table 5-2
ChargeNumber NOA determination

Scenario	Datafill	CAIN flag	NOA
Calling party billed	table FLEXDIAL: SUBR or SUBRPARAM collectable; table FLEXTYPE: BILLFLD BILLNUM	ACCT	ACCT
Calling party billed	table FLEXDIAL: SUBR or SUBRPARAM collectable; table FLEXTYPE: BILLFLD BILLNUM	AUTH	AUTH
Calling party billed	table FLEXDIAL: SUBR or SUBRPARAM; table FLEXTYPE: BILLFLD BILLNUM	CLGPTYADD	See table 5-5 (Calling PartyID)
Calling party billed	table FLEXDIAL: SUBR or SUBRPARAM; table FLEXTYPE: BILLFLD BILLNUM	MCCS	MCCS
Calling party billed	table FLEXDIAL: SUBR or SUBRPARAM; table FLEXTYPE: BILLFLD BILLNUM	PIN	PIN
—end—			

Table 5-2
ChargeNumber NOA determination (continued)

Scenario	Datafill	CAIN flag	NOA
Calling party billed	table FLEXDIAL: SUBR or SUBRPARM; table FLEXTYPE: BILLFLD BILLNUM	none	UNK
Calling party billed	table FLEXDIAL: SUBR or SUBRPARM; table FLEXTYPE: CALLING option	na	I2ANI
Called party billed	table FLEXDIAL: ADDR or ADDRPARM (translated using the ES selector)	na	N00
Called party billed	table FLEXFEAT: CLDPBILL option	na	N00
—end—			

CalledPartyID

Parameter definition

The *CalledPartyID* is the known address or directory number of the called party. This number is used to route the call. When a call is initiated, *CalledPartyID* holds the last digits collected by the ADDR or ADDRPARM collectables prior to querying the SCP. However, this number may change during standard pretranslation or N00 translation (in-switch or off-board).

If the LNP feature is active for the call, the ported number is used to populate this parameter.

AXCESS requirements

When the FlexDial Framework does not execute an ADDR or ADDRPARM collectable, the *CalledPartyID* is not sent to the SCP.

Note: If the switch does not send a *CalledPartyID* to the SCP, then the SCP must return a *CalledPartyID*, unless terminating to a PTS DAL. When call processing enters **Select_Route** PIC without previously querying (or when the SCP returns a Continue), the switch uses the route specified by the ROUTE collectable (if executed) to route the call. However, the switch can only route to an agent that does not require address outpulsing.

Population precedence rules

When populating the *CalledPartyID* parameter of outgoing SCP messages, CAIN assigns its own nature of address, independent of the NOA written into the CDR by the FlexDial Framework. Because of this, you may need to reconcile NOAs recorded by the switch and the SCP during downstream processing of CDRs. Refer to the *UCS DMS-250 NetworkBuilder Application Guide* for the population precedence rules for the *CalledPartyID* parameter.

CalledPartyID (continued)**Nature of address**

The FlexDial Framework assigns its own nature of address (NOA) when collecting address digits. This NOA is written to CDR field CLDNOA. NOAs for CAIN and FlexDial (differences between the switch and SCP) may need to be reconciled at the downstream processor. The following table shows how FlexDial determines the NOA from the collected digits:

CalledPartyID NOA

Signaling method	Digit string	NOA
PTS	0 digits	NO_NUM_OP
	First 3 digits are 011	INTL
	First 3 digits are 01N	INTL_OP
	First 2 digits are 0N; total digit count is 8	SUBR_OP
	First 2 digits are 0N; total digit count is 11	NATL_OP
	First digit is not 0; more than 11 digits	INTL
	First digit is not 0 or 1; more than 10 digits	INTL
	First digit is not 0; total digit count is 10	NATL
	First digits is 1; total digit count is 11	NATL
	First digit is not 0; total digit count is 7	SUBR
Other	UNK	
SS7	na	The NOA contained in the Address parameter of the originating IAM. If the IAM does not contain an NOA, CAIN sets the NOA to UNK.
—end—		

Restrictions

- The SCP should supply an address in the *CalledPartyID* parameter in all cases where the switch's query lacked the *CalledPartyID* or *CollectedAddressInfo* parameters.

CalledPartyID (end)

- Natures of address for CAIN and FlexDial (differences between the switch and SCP) may need to be reconciled at the downstream processor.

CallingPartyID

Parameter definition

The *CallingPartyID* contains the address used for caller identification purposes and is always transmitted.

AXXESS requirements

If the Calling Party Address parameter for the incoming IAM is required, datafill the following:

- table FLEXDIAL – SUBRPARM collectable
- table FLEXTYPE – CAINFLG option as CLGPTYADD

If the Calling Party Address parameter for the incoming IAM is not required, then the Charge Number parameter is required; datafill the following:

- table FLEXDIAL – SUBRPARM collectable
- table FLEXTYPE – CALLING option

Population precedence rules

The following table shows the population precedence rules for the *CallingPartyID* parameter on AXXESS originations (mimicking FGD agents).

CallingPartyID parameter population rules

	Rule	NOA	Numbering Plan
1	Digits received in the <i>CallingPartyID</i> parameter of an <i>Analyze_Route</i> message.	same as received NOA	same as received Numbering Plan
2	Digits collected against the CLGPTYADD CAIN flag through table FLEXTYPE	see Note 2	ISDN
2	Digits collected against the CALLING option through table FLEXTYPE	see Note 2	ISDN
Note 1: Refer to the <i>UCS DMS-250 NetworkBuilder AIN 0.2 TCAP Protocol Definition</i> for encoding information.			
Note 2: Refer to table 5-1 for NOA determination information.			
—continued—			

CallingPartyID (continued)

CallingPartyID parameter population rules

	Rule	NOA	Numbering Plan
3	SNPA datafilled in table TRKFEAT	NATL	ISDN
4	Default SNPA	NATL	ISDN
<p>Note 1: Refer to the <i>UCS DMS-250 NetworkBuilder AIN 0.2 TCAP Protocol Definition</i> for encoding information.</p> <p>Note 2: Refer to table 5-1 for NOA determination information.</p>			
—end—			

Nature of address

The FlexDial Framework assigns its own nature of address (NOA) when collecting calling party digits. This NOA is written to CDR field CLGNOA. The following table shows how FlexDial determines the NOA from the collected digits:

CallingPartyID NOA

Signaling method	Digit string	NOA
PTS	More than 10 digits	INTL
	10 digits	NATL
	7 digits	SUBR
	Other digit count	SUBR
SS7	na	The NOA contained in the Calling Party Address of the originating IAM. If the IAM does not contain an NOA, CAIN sets the NOA to UNK.
—end—		

Restrictions

- Natures of address for CAIN and FlexDial (differences between the switch and SCP) may need to be reconciled at the downstream processor.
- The content of this parameter is used for global title with CAIN_CLID_GT global title translations.

CallingPartyID (end)

Note: If the NOA of the *callingPartyID* is not NATL, the SNPA is used for global title translations.

- The FlexDial Framework does not differentiate between ANIs and PANIs. Therefore, if the CAIN_CLID_GT option is used for global title translations with an invalid national number, global title translations will fail and the query is not delivered to the SCP.
- This parameter is always sent with presentation allowed.
- The presentation restriction indicator from an incoming SS7 IAM or PRI SETUP message is ignored.

ChargeNumber

Parameter definition

This parameter holds the billing number that would be used to populate the CDR at this point in the call. If the switch hasn't determined a charge number, this parameter is not transmitted to the SCP.

AXXESS requirements for calling party billed calls

The BILLFLD BILLNUM option in table FLEXTYPE identifies that the collected digits for the subscriber number type are used to update the BILLNUM field of the CDR. This option also indicates, to CAIN, the billing number for the call.

Note: The BILLCAPT field (SUBR provisioning), in table FLEXDIAL, must be set to Y for the BILLFLD BILLNUM option to be recognized by CAIN.

CAIN uses the digits collected by a SUBR or SUBRPARM collectable with the CALLING option as the ANI for the call. For calling party billed calls, if no billing was identified through the BILLFLD BILLNUM option, CAIN considers the call ANI billed, as identified by a SUBR or SUBRPARM collectable, which processes a FLEXTYPE tuple with the CALLING option provisioned.

AXXESS requirements for called party billed calls

The call is identified as a called party billed when:

- translations through table STDPRTCT identify the call as called party billed (using the ES selector with INWATS, SACREMOT, INWFEAT)
- processing the CLDPBILL option in table FLEXFEAT for a SUBR, SUBRPARM, or CALLTYPE collectable

If the CLDPBILL option is processed after translations are performed, the BILLNUM is changed to the translated address, instead of the dialed N00 number. an N00 address is collected, resulting in a called party billed.

Note: When the CLDPBILL option is set, the address digits collected by the ADDR or ADDRPARAM collectables are written to the BILLNUM field of the CDR.

Population rules

The following table shows the population precedence rules for the *ChargeNumber* parameter.

ChargeNumber (end)**ChargeNumber parameter population rules**

	Rule	NOA	Numbering Plan
1	MCCS or authorization number determined through the <i>O_Feature_Requested</i> feature processor	MCCS	PVT
2	N00 billed	N00	PVT
3	Authcode billed	AUTH	PVT
4	ANI billed (Note 3 and 4)	I2ANI NATL	PVT ISDN
<p>Note 1: An <i>Origination_Attempt</i> message sends the filed authorization code, when present.</p> <p>Note 2: Refer to the <i>UCS DMS-250 NetworkBuilder AIN 0.2 TCAP Protocol Definition</i> for encoding information.</p> <p>Note 3: If the <i>CAIN_PROTOCOL_VERSION</i> parameter in table <i>CAINPARM</i> is set V2 or higher, the info digits are not included in the <i>ChargeNumber</i> parameter and the NOA is set to NATL.</p> <p>Note 4: If an ANI is used for billing, the number is prefixed by information digits, when available. If the information digits are unavailable, the ANI is prefixed by 00.</p>			
—end—			

Restrictions

The SCP should provide a billing number in the *ChargeNumber* parameter in all cases where the query did not include the *ChargeNumber* parameter.

ATTENTION

Failure to provide the *ChargeNumber* parameter in this case will be treated as a fatal application error.

Incoming messages and parameters

ATTENTION

Extension parameters require the CAIN0200 SOC option. Refer to the *UCS DMS-250 Software Optionality Control User's Manual* for more information about SOCs.

SCP responses

Once the SCP has determined how the call is handled, the appropriate TCAP response message is sent to the switch. CAIN software interprets, validates, and performs the instructions contained within the response message.

CAIN/FlexDial interaction does not affect the processing of SCP-returned data. Refer to the *UCS DMS-250 NetworkBuilder Application Guide* for more information.

CAIN and FlexDial tools

CAIN and FlexDial offer several tools for troubleshooting and tracking purposes.

The following tools are useful for CAIN/FlexDial interactions:

- CAINTEST
- SCP simulator
- FLEXSIM
- VPTRACE

CAINTEST

CAINTEST is the Carrier Advanced Intelligent Network (CAIN) message query test tool. You can create and send test TCAP queries to the SCP. The SCP responds as if an actual call were being made and does not recognize any difference between a CAINTEST-generated message and a call-generated messages.

For more information on CAINTEST, refer to the *UCS DMS-250 NetworkBuilder Application Guide* and the *UCS DMS-250 Commands Reference Manual*.

SCP simulator

NetworkBuilder's release of CAIN includes an SCP simulator used to test SCP responses to CAIN framework queries. Simulator logic is provided through user-defined datafill.

For more information on SCP simulator, refer to the *UCS DMS-250 NetworkBuilder Application Guide* and the *UCS DMS-250 Commands Reference Manual*.

FLEXSIM

By using the FLEXSIM tool, you can execute FlexDial collectables and simulate FlexDial call processing.

FLEXSIM provides the following data:

- names and order of the collectables executed during the simulation
- events display
- messages received through table MSGCTR
- features activated, reset or cleared
- translation results
- interaction with CAIN up to and including **Analyze_Information**, including TDPs encountered, subscription changes (including resets and feature clearing), and trigger matches

FLEXSIM interaction with CAIN is discussed further in this chapter.

VPTRACE

VPTRACE allows you to enable or disable the tracing of VAMP messages (logged in VAMP 90x logs). If called with no parameter, the command displays the current status of tracing (enabled or disabled).

For more information on VPTRACE, refer to the *UCS DMS-250 Commands Reference Manual*.

FLEXSIM

FLEXSIM is the FlexDial Framework simulator tool. You can simulate FlexDial call processing translations to understand the dialplan executed by the FlexDial Framework and the resulting translations.

FLEXSIM provides the following data:

- names and order of the collectables executed during the simulation
- events display
- messages received through table MSGCTR
- features activated, reset or cleared
- translation results
- interaction with CAIN up to and including **Analyze_Information**, including TDPs encountered, subscription changes (including resets and feature clearing), and trigger matches

Commands

When the FLEXSIM tool is activated, the user is able to select different commands to perform different tasks. The following is a summary of these commands and their definitions:

- **HELP**—displays a list of subcommands with a brief description
- **SELECT (SEL)**—selects the trunk used for call simulation
- **INPUT (IN)**—selects inputs from user, file, or preset buffers
- **DIGITS (DIGS)**—defines digits for call simulation
- **ISUPNDX (ISUP)**—defines an ISUPTST index for call simulation
- **LEVEL (LEV)**—defines the level of detail for the report
- **FLXREAD**—executes FLEXSIM subcommands from a predefined file
- **STATUS (STAT)**—displays current status of FLEXSIM values
- **START (GO)**—starts call simulation
- **DISPLAY**—displays the integrated services user part (ISUP) IAM in table FLEXSIMT
- **ABORT**—cancels the call simulation in progress
- **QUIT**—exits the FLEXSIM command

Note: Refer to the UCS DMS-250 FlexDial Framework Application Guide for more information.

FLEXSIM (continued)

FLEXSIM and CAIN

Beginning with the UCS07 software release, the FLEXSIM tool is updated to show CAIN interactions. FLEXSIM call simulation terminates on successful trigger criteria matches where the trigger action is QUERY, BLOCK, or FEAT.

FLEXSIM supports three levels of output, each providing different levels of detail. Since the main interaction between CAIN and the FlexDial framework occurs during **Collect_Information**, the output from running a FlexDial simulation shows this interaction. Also, translation results from **Analyze_Information** are included. The three levels of output are defined as follows:

- Level 1 provides a high-level flow through table FLEXDIAL and includes the names and order of the executed collectables, user interaction with digit buffer display, and any resulting translations from **Analyze_Information**.
- Level 2 includes Level 1 output, plus a display of the events from the FlexDial event model which drives FLEXDIAL processing.
- Level 3 includes Levels 1 and 2 output, plus details on messages received (from table MSGCTR) by each collectable, and features activated, reset, or cleared through digit validation or otherwise (for example, CALLTYPE and CLRFRS collectables).

The following table shows the CAIN elements that are added to the output of the FLEXSIM tool, the type of information is displayed, and the FLEXSIM levels where they are displayed.

FLEXSIM (continued)

CAIN elements mapped to FLEXSIM output

FLEXSIM/CAIN display elements/Information displayed	Level 1	Level 2	Level 3
<p>Encountering of the CAIN TDP. Simple strings labeling the CAIN TDP are displayed. Example of FLEXSIM output:</p> <pre> +++ CAIN +++ TDP Encountered: ORIGATT +++ CAIN +++ TDP Encountered: O_FTRREQ +++ CAIN +++ TDP Encountered: INFOCOLL +++ CAIN +++ TDP Encountered: INFOANLZ </pre>	x	x	x
<p>Subscription information during CAIN trigger checking. Includes subscription methods visited at each method/trigger combination as well as tuple information, as appropriate, from tables FLEXTYPE, FLEXVAL, FLEXFEAT, CAINPARAM, STDPRT:STDPRTCT, and CAINGRP. Example of FLEXSIM output:</p> <pre> ***** +++ CAIN +++ Subscription found for Method FLEX_SUBR (precedence level 3) FLEXTYPE: AUTH (BILLFLD BILLNUM REPLACE) (FLEXLOG) (OPERDISP SPL REPLACE) (CAINFLG AUTH) \$ FLEXVAL: AUTH 1 6119999 110999 FLEXFEAT: 110999 (REORGACT I_REO_AD_SD 0 15) (TRANSTS PARTITION 111 0) (PVSPDIDX 51) (CITYCODE 111) (CAINGRP AUTHSTGRP) (REORGTYP ONKEY STR P 15 5) \$ CAINGRP: AUTHSTGRP 555 CAIN02 TCAP_SCCP (COLLINFO INFOCOLL SIOTRK) \$ \$ \$ ***** </pre>			x
<p>Trigger matches during CAIN trigger checking. Includes display of tuples from trigger tables, as well as display of query message contents on QUERY trigger action. Example of FLEXSIM output:</p> <pre> +++ CAIN +++ Trigger Match TABLE: SIOTRK AGTTSTGRP ADDR 2211234 2211234 QUERY SIO_ADDR \$ UserID: AXDAL274TWDTLS 11 BearerCapability: SPEECH CalledPartyID: 2211234 TriggerCriteriaType: SIO_ADDR CallingPartyID: 02214000 </pre>	x	x	x
—continued—			

FLEXSIM (continued)**CAIN elements mapped to FLEXSIM output** (continued)

FLEXSIM/CAIN display elements/Information displayed	Level 1	Level 2	Level 3
Restoration of previous value of a collectable-based subscription method as a result of the processing of a "reset to" reset action. A simple string indicating the CAINGRP restored is displayed. Example of FLEXSIM output <pre>+++ CAIN +++ Subscription reset for FLEX_SUBR: VPNGRP (precedence level 3) +++ CAIN +++ Subscription reset for ADDR: INTLGRP (precedence level 2)</pre>		x	x
Clearing to NIL of a collectable-based subscription method as performed by the CLRFTRS collectable. A simple string indicating which method is cleared is displayed. Example of FLEXSIM output: <pre>+++ CAIN +++ FLEX_SUBR subscription cleared to NIL (precedence level 3) +++ CAIN +++ ADDR subscription cleared to NIL (precedence level 2)</pre>		x	x
—end—			

Using FLEXSIM

The following examples show the use of FLEXSIM on a CAIN call.

Datafill setup

- 1 Table TRKFEAT datafill.

```
>TRKFEAT_0274 (CAIN ) (CAINGRP AGTTSTGRP) (CITYCODE 111) (OHQ )
(SNPA 214) (TRANSTS PARTITION 111 0) (MSGCTR 2003) $ (NOANS DUR 3
NOANSTRMT NTRS) (OHQTERM ) (ONNETTRK ) (SNPA 214) $
```
- 2 Table FLEXTYPE datafill.

```
>AUTH (BILLFLD BILLNUM REPLACE) (FLEXLOG )
(OPERDISP SPL REPLACE) (CAINFLG AUTH) $
```
- 3 Table FLEXVAL datafill.

```
>AUTH 1 6119999 110999
```
- 4 Table FLEXFEAT datafill.

```
>110999 (REORGACT I_REO_AD_SD 0 15) (TRANSTS PARTITION 111 0)
(PVSPDIDX 51) (CITYCODE 111) (CAINGRP AUTHSTGRP)
(REORGTYP ONKEY STR P 15 5) $
```

FLEXSIM (continued)

5 Table CAINGRP datafill.

```
>AUTHSTGRP 555 CAIN02 TCAP_SCCP (COLLINFO INFOCOLL SIOTRK)
$$$
```

```
>AGTTSTGRP 777 CAIN02 TCAP_SCCP (COLLINFO INFOCOLL SIOTRK) $
$$
```

6 Table SIOTRK datafill.

```
>AUTHSTGRP ADDR 221 221 IGNORE $
```

```
>AGTTSTGRP ADDR 221 221 QUERY SIO_ADDR $
```

FLEXSIM Level 1 output with no triggering

1 Set up the FlexDial simulation:

```
>FLEXSIM
```

```
>SELECT axdal274twdtls
```

```
>LEVEL 1
```

```
>GO
```

Example of a MAP response:

```
Executing FLEXDIAL Call Processing Simulation for: AXDAL274TWDTLS
Initial FLEXDIAL Index(s): (SNPA_SNXX_ANI) (I_AU_SD_AD)
*****
+++ CAIN +++ TDP Encountered: ORIGATT
*****
Digits Buffer Digits :
Reset : NO Term : NO Timeout : NO CSP: NO
NOA : UNKNOWN_UNIQUENESS
CarrSel: NO_INDICATION

Flexdial Index: SNPA_SNXX_ANI
Collectable List: (AGNTDATA) >>AGNTDATA<< (SUBR) (OLI) (MODNOA)
(INCLUDE) (ADDR)

Flexdial Index: SNPA_SNXX_ANI
Collectable List: (AGNTDATA) (AGNTDATA) >>SUBR<< (OLI) (MODNOA)
(INCLUDE) (ADDR)

Flexdial Index: SNPA_SNXX_ANI
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) >>OLI<< (MODNOA)
(INCLUDE) (ADDR)

Flexdial Index: I_AU_SD_AD
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) >>MODNOA<<
(INCLUDE) (ADDR)

Flexdial Index: I_AU_SD_AD
```

FLEXSIM (continued)

```
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
>>INCLUDE<< (ADDR)

Flexdial Index: I_AUTH_SD
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
(INCLUDE) >>SUBR<< (ADDR)

Digits Buffer  Digits :
Reset       : NO   Term : NO   Timeout : NO   CSP: NO
NOA        : UNKNOWN_UNIQUENESS
CarrSel    : NO_INDICATION
*****
Peripheral Interaction
*****
Prompt      : Tone - DIAL_TONE_350_440HZ
Digits Required : 5 (min), 7 (max)
Digit Set   : NONE
Signalling  : DTMF
(Use $ to terminate input without entering digits.)
Next par is: <Digits> STRING
Enter: <Digits>
>6119999

Received Digit Set : 6119999
Reporting: 61199
Remaining: 99
*****
End Peripheral Interaction
*****
Digits Buffer  Digits :
Reset       : NO   Term : NO   Timeout : NO   CSP: NO
NOA        : UNKNOWN_UNIQUENESS
CarrSel    : NO_INDICATION
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
(INCLUDE) >>SUBR<< (ADDR)

*****
Peripheral Interaction
*****
Digits Required : 2 (min), 2 (max)
Digit Set      : 99
Signalling     : DTMF
Reporting      : 99
Remaining      : NONE
*****
End Peripheral Interaction
*****
Digits Buffer  Digits : 61199
Reset       : NO   Term : NO   Timeout : NO   CSP: NO
NOA        : UNKNOWN_UNIQUENESS
CarrSel    : NO_INDICATION
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
(INCLUDE) >>SUBR<< (ADDR)

***** Validate *****
Subscriber Type : AUTH
Collected Digits : 6119999
Validation Type : FLEXVAL
Validation Status: No Failure
```


FLEXSIM (continued)

```

FLEXVAL Index      :      1,   FLEXFEAT Index:   110999
Flexdial Index: C_ADDR
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
                  (INCLUDE) (SUBR) >>ADDR<<

```

```

*****
Peripheral Interaction
*****
Digits Required   : 1 (min), 6 (max)
Digit Set        : NONE
Signalling       : DTMF
(Use $ to terminate input without entering digits.)
Next par is: <Digits> STRING
Enter: <Digits>
>2201234

```

```

Received Digit Set : 2201234
Reporting: 2
Remaining: 201234
*****
End Peripheral Interaction
*****
Digits Buffer  Digits :
Reset       : NO   Term : NO   Timeout : NO   CSP: NO
NOA        : UNKNOWN_UNIQUENESS
CarrSel: NO_INDICATION
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
                  (INCLUDE) (SUBR) >>ADDR<<

```

```

*****
Peripheral Interaction
*****
Digits Required   : 5 (min), 5 (max)
Digit Set        : 201234
Signalling       : DTMF
Reporting: 20123
Remaining: 4
*****
End Peripheral Interaction
*****
Digits Buffer  Digits : 2
Reset       : NO   Term : NO   Timeout : NO   CSP: NO
NOA        : UNKNOWN_UNIQUENESS
CarrSel: NO_INDICATION
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
                  (INCLUDE) (SUBR) >>ADDR<<

```

```

*****Validate state data*****
pretranslator name:      DAL
pretranslation digits:  220123
pretranslator route selector: CT
call_feature:          ONNET
*****
Peripheral Interaction
*****
Digits Required   : 0 (min), 1 (max)
Digit Set        : 4
Signalling       : DTMF
Reporting: 4

```

FLEXSIM (continued)

```
Remaining: NONE
*****
End Peripheral Interaction
*****
Digits Buffer Digits :
Reset : NO Term : NO Timeout : NO CSP: NO
NOA : UNKNOWN_UNIQUENESS
CarrSel: NO_INDICATION
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
(INCLUDE) (SUBR) >>ADDR<<

*****Validate state data*****
pretranslator name: DAL
pretranslation digits: 2201234
pretranslator route selector: CT
call_feature: ONNET
*****
+++ CAIN +++ TDP Encountered: O_FTRREQ
*****
+++ CAIN +++ TDP Encountered: O_FTRREQ
*****
called_noa: SUBR
Flexdial Index: C_ADDR
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
(INCLUDE) (SUBR) >>ADDR<<

called_noa: SUBR
*****
+++ CAIN +++ TDP Encountered: INFOCOLL
*****
+++ CAIN +++ TDP Encountered: INFOCOLL
*****
+++ CAIN +++ TDP Encountered: INFOANLZ
*****
+++ CAIN +++ TDP Encountered: INFOANLZ
*****
TRANSLATIONS
*****
TRANSLATE
HNPA_RTER 1A00DC
Dialed : 2201234
STS : 611
TRANS SYS : NA
CALL TYPE : NP
ROUTE : CKT DAL220TWDTGS 1
*****
Flexsim Execution Complete for Trunk: AXDAL274TWDTLS
*****
```

FLEXSIM (continued)**2** During simulation execution, the following logs are generated:

```

250Z      CAIN902 JUN10 13:31:28 6948 INFO CAIN SUBSCRIPTION METHOD
          PIC = COLLINFO
          TDP = NIL
          CAINGRP = AGTTSTGRP
          OTG = AXDAL274TWDTLS
          METHOD = AGENT (PRECEDENCE LEVEL 5)

250Z      CAIN902 JUN10 13:31:31 7049 INFO CAIN SUBSCRIPTION METHOD
          PIC = COLLINFO
          TDP = NIL
          CAINGRP = AUTHSTGRP
          OTG = AXDAL274TWDTLS
          CLGNO = 02214000
          FLEX_ACCESS = AUTH      1      6119999      110999
          SUBR QUALIFIER = CAINFLG AUTH; 6119999
          METHOD = FLEX_SUBR (PRECEDENCE LEVEL 3)

250Z      CDR273 JUN10 13:31:39 7150 INFO CDR_CALL_ENTRY

          RECCD      F0                      TEMPLID      003
          ACTIDX      0                      TOOLGEN      Y
          USEEDIT      N                      VARLNTH      N
          SEQNUM      00213                  CIC
          ORIGTIME    05488                  ORIGDATE     161
          ANSTYPE     00                      TIMECHNG     N
          ORIGAMP     1                      INCINTL
          DISCTIME    05497                  DISCDATE     161
          ANISUFF
          INCBILL
          PINDIGS
          INFODIG
          ACCTCD
          RLTCDR
          UNIVACC
          CALLEDNO    2201234                OUTPULNO
          ORIGOPRT
          CNPREDIG    0                      OPART        111
          TPART       00                      QUEUED       N
          RTELIST
          NETSEC      N                      OPCHOICE
          COLLTIME    009                    ADIN
          LNPCHK
          WBCKTS
          BILLTYPE
          DISCTYPE    3                      ORIGGRP      0274
          PASSTHRU    0                      ORIGMEM      0001
          PRESIND
          OSRASSOC    N                      TERMGRP
          FINTKGRP
          COSOVE      N                      TERMMEM
          SCPBILL
          PIC
          TRMTCD      000                    DIGDATA     N
          CRID
          ORIGPVN
          TRMPVN

```

FLEXSIM (continued)

ORIGLRN	PORTEDNO
CAINCT	ACG N
DNISNOA	DCR N
MLTCOSID	COSINDEX
CARRSEL	

FLEXSIM (continued)**FLEXSIM Level 1 output; triggering at Info_Collected**

1 Set up the FlexDial simulation:

>FLEXSIM

>SELECT axdal274twdtls

>LEVEL 1

>GO

Example of a MAP response:

```

Executing FLEXDIAL Call Processing Simulation for: AXDAL274TWDTLS
Initial FLEXDIAL Index(s): (SNPA_SNXX_ANI) (I_AU_SD_AD)
*****
+++ CAIN +++ TDP Encountered: ORIGATT
*****
Digits Buffer  Digits :
                Reset : NO   Term : NO   Timeout : NO   CSP: NO
                NOA   : UNKNOWN_UNIQUENESS
                CarrSel: NO_INDICATION

Flexdial Index: SNPA_SNXX_ANI
Collectable List: (AGNTDATA) >>AGNTDATA<< (SUBR) (OLI) (MODNOA)
                  (INCLUDE) (ADDR)

Flexdial Index: SNPA_SNXX_ANI
Collectable List: (AGNTDATA) (AGNTDATA) >>SUBR<< (OLI) (MODNOA)
                  (INCLUDE) (ADDR)

Flexdial Index: SNPA_SNXX_ANI
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) >>OLI<< (MODNOA)
                  (INCLUDE) (ADDR)

Flexdial Index: I_AU_SD_AD
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) >>MODNOA<<
                  (INCLUDE) (ADDR)

Flexdial Index: I_AU_SD_AD
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
                  >>INCLUDE<< (ADDR)

Flexdial Index: I_AUTH_SD
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
                  (INCLUDE) >>SUBR<< (ADDR)

Digits Buffer  Digits :
                Reset : NO   Term : NO   Timeout : NO   CSP: NO
                NOA   : UNKNOWN_UNIQUENESS
                CarrSel: NO_INDICATION
*****
Peripheral Interaction

```

FLEXSIM (continued)

```

*****
Prompt : Tone - DIAL_TONE_350_440HZ
Digits Required : 5 (min), 7 (max)
Digit Set : NONE
Signalling : DTMF
(Use $ to terminate input without entering digits.)
Next par is: <Digits> STRING
Enter: <Digits>
>6119999

Received Digit Set : 6119999
Reporting: 61199
Remaining: 99
*****
End Peripheral Interaction
*****
Digits Buffer Digits :
Reset : NO Term : NO Timeout : NO CSP: NO
NOA : UNKNOWN_UNIQUENESS
CarrSel: NO_INDICATION
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
(INCLUDE) >>SUBR<< (ADDR)

*****
Peripheral Interaction
*****
Digits Required : 2 (min), 2 (max)
Digit Set : 99
Signalling : DTMF
Reporting: 99
Remaining: NONE
*****
End Peripheral Interaction
*****
Digits Buffer Digits : 61199
Reset : NO Term : NO Timeout : NO CSP: NO
NOA : UNKNOWN_UNIQUENESS
CarrSel: NO_INDICATION
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
(INCLUDE) >>SUBR<< (ADDR)

***** Validate *****
Subscriber Type : AUTH
Collected Digits : 6119999
Validation Type : FLEXVAL
Validation Status: No Failure
FLEXVAL Index : 1, FLEXFEAT Index: 110999
Flexdial Index: C_ADDR
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
(INCLUDE) (SUBR) >>ADDR<<

*****
Peripheral Interaction
*****
Digits Required : 1 (min), 6 (max)
Digit Set : NONE
Signalling : DTMF
(Use $ to terminate input without entering digits.)
Next par is: <Digits> STRING

```

FLEXSIM (continued)

```

Enter: <Digits>
>2211234

Received Digit Set : 2211234
Reporting: 2
Remaining: 211234
*****
End Peripheral Interaction
*****
Digits Buffer Digits :
Reset : NO Term : NO Timeout : NO CSP: NO
NOA : UNKNOWN_UNIQUENESS
CarrSel: NO_INDICATION
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
(INCLUDE) (SUBR) >>ADDR<<

*****
Peripheral Interaction
*****
Digits Required : 5 (min), 5 (max)
Digit Set : 211234
Signalling : DTMF
Reporting: 21123
Remaining: 4
*****
End Peripheral Interaction
*****
Digits Buffer Digits : 2
Reset : NO Term : NO Timeout : NO CSP: NO
NOA : UNKNOWN_UNIQUENESS
CarrSel: NO_INDICATION
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
(INCLUDE) (SUBR) >>ADDR<<

*****Validate state data*****
pretranslator name: DAL
pretranslation digits: 221123
pretranslator route selector: CT
call_feature: ONNET
*****
Peripheral Interaction
*****
Digits Required : 0 (min), 1 (max)
Digit Set : 4
Signalling : DTMF
Reporting: 4
Remaining: NONE
*****
End Peripheral Interaction
*****
Digits Buffer Digits :
Reset : NO Term : NO Timeout : NO CSP: NO
NOA : UNKNOWN_UNIQUENESS
CarrSel: NO_INDICATION
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
(INCLUDE) (SUBR) >>ADDR<<

*****Validate state data*****
pretranslator name: DAL

```

FLEXSIM (continued)

```

pretranslation digits:          2211234
pretranslator route selector:   CT
call_feature:                   ONNET
*****
+++ CAIN +++ TDP Encountered: O_FTRREQ
*****
+++ CAIN +++ TDP Encountered: O_FTRREQ
*****
called_noa:                      SUBR
Flexdial Index: C_ADDR
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
                  (INCLUDE) (SUBR) >>ADDR<<

called_noa:                      SUBR
*****
+++ CAIN +++ TDP Encountered: INFOCOLL
*****
+++ CAIN +++ Trigger Match
TABLE: SIOTRK
      AUTHSTGRP ADDR 2211234 2211234 IGNORE $
      UserID: AXDAL274TWDTLS 11
      BearerCapability: SPEECH
      CalledPartyID: 2211234
      TriggerCriteriaType: SIO_ADDR
      CallingPartyID: 02214000
*****
+++ CAIN +++ TDP Encountered: INFOCOLL
*****
+++ CAIN +++ Trigger Match
TABLE: SIOTRK
      AGTSTGRP ADDR 2211234 2211234 QUERY SIO_ADDR $
      UserID: AXDAL274TWDTLS 11
      BearerCapability: SPEECH
      CalledPartyID: 2211234
      TriggerCriteriaType: SIO_ADDR
      CallingPartyID: 02214000
*****
flag timeout
*****
Flexsim Execution Complete for Trunk: AXDAL274TWDTLS
*****
>quit

```


FLEXSIM (continued)**2** During simulation execution, the following logs are generated:

```

250Z      CAIN902 JUN10 13:32:14 7958 INFO CAIN SUBSCRIPTION METHOD
          PIC = COLLINFO
          TDP = NIL
          CAINGRP = AGTTSTGRP
          OTG = AXDAL274TWDTLS
          METHOD = AGENT (PRECEDENCE LEVEL 5)

250Z      CAIN902 JUN10 13:32:17 8059 INFO CAIN SUBSCRIPTION METHOD
          PIC = COLLINFO
          TDP = NIL
          CAINGRP = AUTHSTGRP
          OTG = AXDAL274TWDTLS
          CLGNO = 02214000
          FLEX_ACCESS = AUTH      1      6119999      110999
          SUBR_QUALIFIER = CAINFLG AUTH; 6119999
          METHOD = FLEX_SUBR (PRECEDENCE LEVEL 3)

250Z      CAIN903 JUN10 13:32:21 8260 INFO CAINGRP TO TRIGGER
          PIC = COLLINFO
          TDP = INFOCOLL
          TRIGGER = SIOTRK
          TRIGVIA = ADDR 2211234
          CAINGRP = AUTHSTGRP
          ACTION = IGNORE
          OTG = AXDAL274TWDTLS

250Z      CAIN903 JUN10 13:32:21 8361 INFO CAINGRP TO TRIGGER
          PIC = COLLINFO
          TDP = INFOCOLL
          TRIGGER = SIOTRK
          TRIGVIA = ADDR 2211234
          CAINGRP = AGTTSTGRP
          ACTION = QUERY
          OTG = AXDAL274TWDTLS

250Z      VAMP902 JUN10 13:32:21 8462 INFO VAMP OUTBOUND MESSAGE
KEY:      MSGSET=CAIN02, PROTO=TCAP_SCCP
CONXTX:  DEFAULT
TRID:     00 01 00 25
ROUTE:    GTTYPE=CAIN_CLID_GT, GTDIGS=214000
PACKAGE   : QWP
*** COMPONENT ***
COMP      : INVL
COMP_ID   : 01
OPCODE    : INFO_COLLECTED
*** PARAMETERS ***
USERID    : CAIN SWID = 011 TRKGRP = 00274
BEARCAP   : 3_1KHZ
CHGNO     : NOA = AUTH      NUMPLAN = PRIV      DIGITS = 6119999
TRIGCRIT  : SIO_ADDR
CLGNO     : NOA = NATL      NUMPLAN = ISDN      DIGITS = 214000
COLLADDR  : NOA = UNKNOWN  NUMPLAN = UNKNOWN  DIGITS = 2211234
----- hex dump of message -----
E2 3B C7 04 01 00 25 00 E8 33 E9 31 CF 01 01 D1
02 64 02 30 28 BF 35 07 81 05 00 10 01 20 47 8D

```

FLEXSIM (continued)

```

01 01 93 06 E4 50 16 91 99 09 9F 34 01 6B 92 05
03 10 12 04 00 96 06 80 00 22 11 32 04

250Z   CAIN900 JUN10 13:32:21 8563 INFO SCP SIMULATOR INDICES
      TRANSID = 0001 0025
      CAINKEY = INFO_COLLECTED
      CAIN DALLAS AXDAL274
      SIO_ADDR
      CAINMTCH = 1497

250Z   VAMP901 JUN10 13:32:24 8664 INFO VAMP INBOUND MESSAGE
KEY:    MSGSET=CAIN02, PROTO=TCAP_SCCP
CONXTX: DEFAULT
TRID:   00 01 00 25
ROUTE:  PC=01-08-01, SSN=33
PACKAGE : RESP
*** COMPONENT ***
COMP    : INVL
COMP_ID : 5E 01
OPCODE  : ANALYZE_ROUTE
*** PARAMETERS ***
PRITRK  : OPULSNO = N      SWID = 011      TRKGRP = 00220
----- hex dump of message -----
E4 1C C7 04 01 00 25 00 E8 14 E9 12 CF 02 5E 01
D1 02 65 01 30 08 9F 2A 05 80 10 01 20 02

250Z   CDR273 JUN10 13:32:25 8765 INFO CDR_CALL_ENTRY

RECCD      F0                      TEMPLID    003
ACTIDX      0                      TOOLGEN     Y
USEEDIT     N                      VARLNGTH   N
SEQNUM      00214                  CIC
ORIGTIME    05534                  ORIGDATE   161
ANSTYPE     00                    TIMECHNG   N
ORIGAMPM    1                      INCINTL
DISCTIME    05544                  DISCDATE   161
ANISUFF
INCBILL
PINDIGS
INFODIG
ACCTCD
RLTCDR
UNIVACC
CALLEDNO    2211234                OUTPULNO
ORIGOPRT
CNPREDIG    0                      OPART
TPART
RTELIST
NETSEC      N                      OPCHOICE
COLLTIME    007                    ADIN
LNPCHECK
WBCKTS
BILLTYPE
DISCTYPE    3                      ORIGGRP    0274
PASSTHRU    0                      ORIGMEM    0001
PRESIND
OSRASSOC    N                      TERMMEM
FINTKGRP
COSOVE      N                      TRAP       N
FINTKMEM

```

FLEXSIM (continued)

SCPBILL		TDP	03
PIC	03	TRIGGER	05
TRMTCD	000	COMPCODE	0
CRID		DNIS	
ORIGPVN		TERMPVN	
ORIGLRN		PORTEDNO	
CAINCT		ACG	N
DNISNOA		DCR	N
MLTCOSID		COSINDEX	
CARRSEL			

FLEXSIM (continued)

FLEXSIM Level 3 output with no triggering

1 Set up the FlexDial simulation:

>FLEXSIM

>SELECT axdal274twdtls

>LEVEL 3

>GO

Example of a MAP response:

```
Executing FLEXDIAL Call Processing Simulation for: AXDAL274TWDTLS
Initial FLEXDIAL Index(s): (SNPA_SNXX_ANI) (I_AU_SD_AD)
*****
+++ CAIN +++ Subscription found for Method AGENT (precedence level 5)
*****
+++ CAIN +++ TDP Encountered: ORIGATT
*****
Digits Buffer  Digits :
                Reset  : NO   Term  : NO   Timeout : NO   CSP: NO
                NOA    : UNKNOWN_UNIQUENESS
                CarrSel: NO_INDICATION

-----
MsgAnalysis Event
-----
Next State:Setup State
-----
Setup Event
-----
Processing Collectable:AGNTDATA
-----
Start Event
-----
Processing Collectable:AGNTDATA
***** Collectable: AgntDataCol *****
Loc: PREFIX
digType: SNXX
agntDgt: 214
***** Digit Buffer *****
Digits Buffer  Digits :
                Reset  : NO   Term  : NO   Timeout : NO   CSP: NO
                NOA    : UNKNOWN_UNIQUENESS
                CarrSel: NO_INDICATION
Current State : 13
-----
ApplyFtrs Event
-----
***** Collectable: AgntDataCol *****
Loc: PREFIX
```

FLEXSIM (continued)

```

digType: SNXX
agntDgt: 000
***** Digit Buffer *****
Digits Buffer  Digits : 000
               Reset  : NO   Term  : NO   Timeout : NO   CSP: NO
               NOA    : UNKNOWN_UNIQUENESS
               CarrSel: NO_INDICATION
Current State : 13
-----
Collect Event
-----
Flexdial Index: SNPA_SNXX_ANI
Collectable List: (AGNTDATA) >>AGNTDATA<< (SUBR) (OLI) (MODNOA)
                  (INCLUDE) (ADDR)
-----
Start Event
-----
Processing Collectable:AGNTDATA
***** Collectable: AgntDataCol *****
Loc: PREFIX
digType: SNPA
agntDgt: 000
***** Digit Buffer *****
Digits Buffer  Digits : 000
               Reset  : NO   Term  : NO   Timeout : NO   CSP: NO
               NOA    : UNKNOWN_UNIQUENESS
               CarrSel: NO_INDICATION
Current State : 13
-----
ApplyFtrs Event
-----
***** Collectable: AgntDataCol *****
Loc: PREFIX
digType: SNPA
agntDgt: 214
***** Digit Buffer *****
Digits Buffer  Digits : 214000
               Reset  : NO   Term  : NO   Timeout : NO   CSP: NO
               NOA    : UNKNOWN_UNIQUENESS
               CarrSel: NO_INDICATION
Current State : 13
-----
Collect Event
-----
Flexdial Index: SNPA_SNXX_ANI
Collectable List: (AGNTDATA) (AGNTDATA) >>SUBR<< (OLI) (MODNOA)
                  (INCLUDE) (ADDR)
-----
Start Event
-----
Processing Collectable:SUBR
-----
Parse Event
-----
ApplyFtrs Event
-----

```

FLEXSIM (continued)

```
-----  
Collect Event  
-----  
Flexdial Index: SNPA_SNXX_ANI  
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) >>OLI<< (MODNOA)  
                  (INCLUDE) (ADDR)  
  
Name: OLI  
Print Data:  
UNKNOWN event  
-----  
Start Event  
-----  
Processing Collectable:OLI  
Name: OLI  
Print Data:  
Filed: POS PREFIX DIGITS:           02  
State: start  
-----  
Parse Event  
-----  
Name: OLI  
Print Data:  
State: parse  
-----  
ApplyFtrs Event  
-----  
Name: OLI  
Print Data:  
Captured INFODIGS in CDR:  
State: applyftrs  
-----  
Collect Event  
-----  
Flexdial Index: I_AU_SD_AD  
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) >>MODNOA<<  
                  (INCLUDE) (ADDR)  
  
***** Collectable: ModNoaCol *****  
Party: CALLED  
NATOFADD: UNKNOWN  
IF Collectable: MODNOA  
State..... Y  
Success..... Y  
Compmeth.... NOT  
Action.....  
True index...  
False index..  
-----  
Start Event  
-----  
Processing Collectable:MODNOA  
***** Collectable: ModNoaCol *****  
Party: CALLED  
NATOFADD: UNKNOWN  
IF Collectable: MODNOA  
State..... Y  
Success..... Y  
Compmeth.... NOT
```

FLEXSIM (continued)

```

Action.....
True index...
False index..
-----
ApplyFtrs Event
-----
***** Collectable: ModNoaCol *****
Party: CALLED
NATOFADD: UNKNOWN
IF Collectable: MODNOA
State..... Y
Success..... Y
Compmeth.... NOT
Action.....
True index...
False index..
-----
Collect Event
-----
Flexdial Index: I_AU_SD_AD
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
                  >>INCLUDE<< (ADDR)

***** Collectable: INCLUDE *****
Current State : 13
-----
Start Event
-----
Processing Collectable:INCLUDE
***** Collectable: INCLUDE *****
INCLUDE Flexdial index: I_AUTH_SD
Current State : 13
-----
ApplyFtrs Event
-----
***** Collectable: INCLUDE *****
INCLUDE Flexdial index: I_AUTH_SD
Current State : 13
-----
Collect Event
-----
Flexdial Index: I_AUTH_SD
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
                  (INCLUDE) >>SUBR<< (ADDR)

-----
Start Event
-----
Processing Collectable:SUBR
-----
Parse Event
-----
Prompt Event
-----
Digits Buffer  Digits :
Reset : NO    Term : NO    Timeout : NO    CSP: NO
NOA       : UNKNOWN_UNIQUENESS
CarrSel: NO_INDICATION

```

FLEXSIM (continued)

```
*****
Peripheral Interaction
*****
Prompt : Tone - DIAL_TONE_350_440HZ
Digits Required : 5 (min), 7 (max)
Digit Set : NONE
Signalling : DTMF
(Use $ to terminate input without entering digits.)
Next par is: <Digits> STRING
Enter: <Digits>
>6119999

Received Digit Set : 6119999
Reporting: 61199
Remaining: 99
*****
End Peripheral Interaction
*****
Digits Buffer Digits :
Reset : NO Term : NO Timeout : NO CSP: NO
NOA : UNKNOWN_UNIQUENESS
CarrSel: NO_INDICATION
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
(INCLUDE) >>SUBR<< (ADDR)

-----
MsgAnalysis Event
-----
Next State:Load State
-----

Load Event
-----
*****
Peripheral Interaction
*****
Digits Required : 2 (min), 2 (max)
Digit Set : 99
Signalling : DTMF
Reporting: 99
Remaining: NONE
*****
End Peripheral Interaction
*****
Digits Buffer Digits : 61199
Reset : NO Term : NO Timeout : NO CSP: NO
NOA : UNKNOWN_UNIQUENESS
CarrSel: NO_INDICATION
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
(INCLUDE) >>SUBR<< (ADDR)

-----
MsgAnalysis Event
-----
Next State:Load State
-----

Load Event
-----
```


FLEXSIM (continued)

```

Digits Buffer   Digits : 6119999
                Reset  : NO   Term  : NO   Timeout : NO   CSP: NO
                NOA    : UNKNOWN_UNIQUENESS
                CarrSel: NO_INDICATION
-----
Parse Event
-----
Validate Event
-----
***** Validate *****
Subscriber Type : AUTH
Collected Digits : 6119999
Validation Type : FLEXVAL
Validation Status: No Failure
FLEXVAL Index   :           1,   FLEXFEAT Index:   110999
*****
+++ CAIN +++ Subscription found for Method FLEX_SUBR (precedence level
3)
FLEXTYPE:
  AUTH (BILLFLD BILLNUM REPLACE) (FLEXLOG ) (OPERDISP SPL REPLACE)
  (CAINFLG AUTH) $
FLEXVAL:
  AUTH 1 6119999 110999
FLEXFEAT:
  110999 (REORGACT I_REO_AD_SD 0 15) (TRANSTS PARTITION 111 0)
  (PVSPDIDX 51) (CITYCODE 111) (CAINGRP AUTHSTGRP)
  (REORGTYP ONKEY STR P 15 5) $
CAINGRP:
  AUTHSTGRP 555 CAIN02 TCAP_SCCP (COLLINFO INFOCOLL SIOTRK) $ $ $
*****
-----
ApplyFtrs Event
-----
Collect Event
-----
Flexdial Index: C_ADDR
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
                  (INCLUDE) (SUBR) >>ADDR<<
-----
Start Event
-----
Processing Collectable:ADDR
-----
Parse Event
-----
Request Event
-----
*****
Peripheral Interaction
*****
Digits Required : 1 (min), 6 (max)
Digit Set      : NONE
Signalling     : DTMF
(Use $ to terminate input without entering digits.)
Next par is: <Digits> STRING

```

FLEXSIM (continued)

```
Enter: <Digits>
>2201234

Received Digit Set : 2201234
Reporting: 2
Remaining: 201234
*****
End Peripheral Interaction
*****
Digits Buffer Digits :
Reset : NO Term : NO Timeout : NO CSP: NO
NOA : UNKNOWN_UNIQUENESS
CarrSel: NO_INDICATION
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
(INCLUDE) (SUBR) >>ADDR<<

-----
MsgAnalysis Event
-----
Next State:Load State
-----

Load Event
-----
*****
Peripheral Interaction
*****
Digits Required : 5 (min), 5 (max)
Digit Set : 201234
Signalling : DTMF
Reporting: 20123
Remaining: 4
*****
End Peripheral Interaction
*****
Digits Buffer Digits : 2
Reset : NO Term : NO Timeout : NO CSP: NO
NOA : UNKNOWN_UNIQUENESS
CarrSel: NO_INDICATION
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
(INCLUDE) (SUBR) >>ADDR<<

-----
MsgAnalysis Event
-----
Next State:Load State
-----

Load Event
-----
Digits Buffer Digits : 220123
Reset : NO Term : NO Timeout : NO CSP: NO
NOA : UNKNOWN_UNIQUENESS
CarrSel: NO_INDICATION
-----
Parse Event
-----

Validate Event
```

FLEXSIM (continued)

```

-----
*****Validate state data*****
pretranslator name:           DAL
pretranslation digits:       220123
pretranslator route selector: CT
call_feature:                ONNET
-----
Request Event
-----
*****
Peripheral Interaction
*****
Digits Required : 0 (min), 1 (max)
Digit Set : 4
Signalling : DTMF
Reporting: 4
Remaining: NONE
*****
End Peripheral Interaction
*****
Digits Buffer Digits :
Reset : NO Term : NO Timeout : NO CSP: NO
NOA : UNKNOWN_UNIQUENESS
CarrSel: NO_INDICATION
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
                  (INCLUDE) (SUBR) >>ADDR<<
-----
MsgAnalysis Event
-----
Next State:Load State
-----
Load Event
-----
Digits Buffer Digits : 4
Reset : NO Term : NO Timeout : NO CSP: NO
NOA : UNKNOWN_UNIQUENESS
CarrSel: NO_INDICATION
-----
Parse Event
-----
Validate Event
-----
*****Validate state data*****
pretranslator name:           DAL
pretranslation digits:       2201234
pretranslator route selector: CT
call_feature:                ONNET
*****
+++ CAIN +++ TDP Encountered: O_FTRREQ
*****
+++ CAIN +++ TDP Encountered: O_FTRREQ
*****
-----
ApplyFtrs Event
-----

```

FLEXSIM (continued)

```
*****ApplyFtrs state data*****
called_noa:                               SUBR
-----
Collect Event
-----
Flexdial Index: C_ADDR
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
                  (INCLUDE) (SUBR) >>ADDR<<

*****ApplyFtrs state data*****
called_noa:                               SUBR
*****
+++ CAIN +++ TDP Encountered: INFOCOLL
*****
+++ CAIN +++ TDP Encountered: INFOCOLL
*****
+++ CAIN +++ TDP Encountered: INFOANLZ
*****
+++ CAIN +++ TDP Encountered: INFOANLZ
*****
TRANSLATIONS
*****
TRANSLATE
  HNP_A_RTER 1A00DC
    Dialed   : 2201234
    STS      : 611
    TRANS SYS : NA
    CALL TYPE : NP
    ROUTE    : CKT   DAL220TWDTGS   1
*****
Flexsim Execution Complete for Trunk: AXDAL274TWDTLS
*****
```

FLEXSIM (continued)**2** During simulation execution, the following logs are generated:

```

250Z      CAIN902 JUN10 13:33:35 0679 INFO CAIN SUBSCRIPTION METHOD
          PIC = COLLINFO
          TDP = NIL
          CAINGRP = AGTTSTGRP
          OTG = AXDAL274TWDTLS
          METHOD = AGENT (PRECEDENCE LEVEL 5)

250Z      CAIN902 JUN10 13:33:40 0780 INFO CAIN SUBSCRIPTION METHOD
          PIC = COLLINFO
          TDP = NIL
          CAINGRP = AUTHSTGRP
          OTG = AXDAL274TWDTLS
          CLGNO = 02214000
          FLEX_ACCESS = AUTH      1      6119999      110999
          SUBR QUALIFIER = CAINFLG AUTH; 6119999
          METHOD = FLEX_SUBR (PRECEDENCE LEVEL 3)

250Z      CDR273 JUN10 13:33:45 1083 INFO CDR_CALL_ENTRY

          RECCD      F0                      TEMPLID    003
          ACTIDX     0                      TOOLGEN    Y
          USEEDIT    N                      VARLNTH    N
          SEQNUM     00220                  CIC
          ORIGTIME   05615                  ORIGDATE   161
          ANSTYPE    00                      TIMECHNG   N
          ORIGAMPM   1                      INCINTL
          DISCTIME   05625                  DISCDATE   161
          ANISUFF
          INCBILL
          PINDIGS
          INFODIG
          ACCTCD
          RLTCDR
          UNIVACC
          CALLEDNO   2201234                OUTPULNO
          ORIGOPRT
          CNPREDIG   0                      OPART      111
          TPART      00                      QUEUED     N
          RTELIST
          NETSEC     N                      OPCHOICE
          COLLTIME   010                  ADIN
          LNPCHECK
          WBCKTS
          BILLYPE
          DISCTYPE   3                      ORIGGRP    0274
          PASSTHRU   0                      ORIGMEM    0001
          PRESIND
          OSRASSOC   N                      TERMGRP
          FINTKGRP
          COSOVE     N                      TERMMEM
          SCPBILL
          PIC
          TRMTCD     000                    DIGDATA    N
          CRID
          TRIGGER
          COMPCODE   0                      TRAP       N
          DNIS
          FINTKMEM
          TDP
          TRIGGER
          COMPCODE   0                      TRAP       N
          DNIS

```

FLEXSIM (continued)

ORIGPVN	TERMPVN
ORIGLRN	PORTEDNO
CAINCT	ACG N
DNISNOA	DCR N
MLTCOSID	COSINDEX
CARRSEL	

FLEXSIM (continued)**FLEXSIM Level 3 output; triggering at Info_Collected**

1 Set up the FlexDial simulation:

>FLEXSIM

>SELECT axdal274twdtls

>LEVEL 3

>GO

Example of a MAP response:

```

Executing FLEXDIAL Call Processing Simulation for: AXDAL274TWDTLS
Initial FLEXDIAL Index(s): (SNPA_SNXX_ANI) (I_AU_SD_AD)
*****
+++ CAIN +++ Subscription found for Method AGENT (precedence level 5)
*****
+++ CAIN +++ TDP Encountered: ORIGATT
*****
Digits Buffer  Digits :
                Reset : NO   Term : NO   Timeout : NO   CSP: NO
                NOA   : UNKNOWN_UNIQUENESS
                CarrSel: NO_INDICATION

-----
MsgAnalysis Event
-----
Next State:Setup State
-----

Setup Event
-----

Processing Collectable:AGNTDATA
-----

Start Event
-----

Processing Collectable:AGNTDATA
***** Collectable: AgntDataCol *****
Loc: PREFIX
digType: SNXX
agntDgt: 214
***** Digit Buffer *****
Digits Buffer  Digits :
                Reset : NO   Term : NO   Timeout : NO   CSP: NO
                NOA   : UNKNOWN_UNIQUENESS
                CarrSel: NO_INDICATION

Current State : 13
-----

ApplyFtrs Event
-----

***** Collectable: AgntDataCol *****
Loc: PREFIX

```

FLEXSIM (continued)

```

digType: SNXX
agntDgt: 000
***** Digit Buffer *****
Digits Buffer  Digits : 000
               Reset  : NO   Term  : NO   Timeout : NO   CSP: NO
               NOA    : UNKNOWN_UNIQUENESS
               CarrSel: NO_INDICATION
Current State : 13
-----
Collect Event
-----
Flexdial Index: SNPA_SNXX_ANI
Collectable List: (AGNTDATA) >>AGNTDATA<< (SUBR) (OLI) (MODNOA)
                  (INCLUDE) (ADDR)
-----
Start Event
-----
Processing Collectable:AGNTDATA
***** Collectable: AgntDataCol *****
Loc: PREFIX
digType: SNPA
agntDgt: 000
***** Digit Buffer *****
Digits Buffer  Digits : 000
               Reset  : NO   Term  : NO   Timeout : NO   CSP: NO
               NOA    : UNKNOWN_UNIQUENESS
               CarrSel: NO_INDICATION
Current State : 13
-----
ApplyFtrs Event
-----
***** Collectable: AgntDataCol *****
Loc: PREFIX
digType: SNPA
agntDgt: 214
***** Digit Buffer *****
Digits Buffer  Digits : 214000
               Reset  : NO   Term  : NO   Timeout : NO   CSP: NO
               NOA    : UNKNOWN_UNIQUENESS
               CarrSel: NO_INDICATION
Current State : 13
-----
Collect Event
-----
Flexdial Index: SNPA_SNXX_ANI
Collectable List: (AGNTDATA) (AGNTDATA) >>SUBR<< (OLI) (MODNOA)
                  (INCLUDE) (ADDR)
-----
Start Event
-----
Processing Collectable:SUBR
-----
Parse Event
-----
ApplyFtrs Event
-----

```


FLEXSIM (continued)

```

-----
Collect Event
-----
Flexdial Index: SNPA_SNXX_ANI
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) >>OLI<< (MODNOA)
                  (INCLUDE) (ADDR)

Name: OLI
Print Data:
UNKNOWN event
-----
Start Event
-----
Processing Collectable:OLI
Name: OLI
Print Data:
Filed: POS PREFIX DIGITS:          02
State: start
-----
Parse Event
-----
Name: OLI
Print Data:
State: parse
-----
ApplyFtrs Event
-----
Name: OLI
Print Data:
Captured INFODIGS in CDR:
State: applyftrs
-----
Collect Event
-----
Flexdial Index: I_AU_SD_AD
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) >>MODNOA<<
                  (INCLUDE) (ADDR)

***** Collectable: ModNoaCol *****
Party: CALLED
NATOFADD: UNKNOWN
IF Collectable: MODNOA
State..... Y
Success..... Y
Compmeth.... NOT
Action.....
True index...
False index..
-----
Start Event
-----
Processing Collectable:MODNOA
***** Collectable: ModNoaCol *****
Party: CALLED
NATOFADD: UNKNOWN
IF Collectable: MODNOA
State..... Y
Success..... Y
Compmeth.... NOT

```

FLEXSIM (continued)

```

Action.....
True index...
False index..
-----
ApplyFtrs Event
-----
***** Collectable: ModNoaCol *****
Party: CALLED
NATOFADD: UNKNOWN
IF Collectable: MODNOA
State..... Y
Success..... Y
Compmeth.... NOT
Action.....
True index...
False index..
-----
Collect Event
-----
Flexdial Index: I_AU_SD_AD
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
                  >>INCLUDE<< (ADDR)

***** Collectable: INCLUDE *****
Current State : 13
-----
Start Event
-----
Processing Collectable:INCLUDE
***** Collectable: INCLUDE *****
INCLUDE Flexdial index: I_AUTH_SD
Current State : 13
-----
ApplyFtrs Event
-----
***** Collectable: INCLUDE *****
INCLUDE Flexdial index: I_AUTH_SD
Current State : 13
-----
Collect Event
-----
Flexdial Index: I_AUTH_SD
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
                  (INCLUDE) >>SUBR<< (ADDR)

-----
Start Event
-----
Processing Collectable:SUBR
-----
Parse Event
-----
Prompt Event
-----
Digits Buffer  Digits :
Reset       : NO   Term : NO   Timeout : NO   CSP: NO
NOA         : UNKNOWN_UNIQUENESS
CarrSel     : NO_INDICATION

```

FLEXSIM (continued)

```

*****
Peripheral Interaction
*****
Prompt : Tone - DIAL_TONE_350_440HZ
Digits Required : 5 (min), 7 (max)
Digit Set : NONE
Signalling : DTMF
(Use $ to terminate input without entering digits.)
Next par is: <Digits> STRING
Enter: <Digits>
>6119999

Received Digit Set : 6119999
Reporting: 61199
Remaining: 99
*****
End Peripheral Interaction
*****
Digits Buffer Digits :
Reset : NO Term : NO Timeout : NO CSP: NO
NOA : UNKNOWN_UNIQUENESS
CarrSel: NO_INDICATION
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
(INCLUDE) >>SUBR<< (ADDR)

-----
MsgAnalysis Event
-----
Next State:Load State
-----

Load Event
-----
*****
Peripheral Interaction
*****
Digits Required : 2 (min), 2 (max)
Digit Set : 99
Signalling : DTMF
Reporting: 99
Remaining: NONE
*****
End Peripheral Interaction
*****
Digits Buffer Digits : 61199
Reset : NO Term : NO Timeout : NO CSP: NO
NOA : UNKNOWN_UNIQUENESS
CarrSel: NO_INDICATION
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
(INCLUDE) >>SUBR<< (ADDR)

-----
MsgAnalysis Event
-----
Next State:Load State
-----

Load Event
-----

```

FLEXSIM (continued)

```

Digits Buffer      Digits : 6119999
                  Reset  : NO   Term : NO   Timeout : NO   CSP: NO
                  NOA    : UNKNOWN_UNIQUENESS
                  CarrSel: NO_INDICATION
-----
Parse Event
-----
Validate Event
-----
***** Validate *****
Subscriber Type   : AUTH
Collected Digits : 6119999
Validation Type   : FLEXVAL
Validation Status : No Failure
FLEXVAL Index    :          1,   FLEXFEAT Index:   110999
*****
+++ CAIN +++ Subscription found for Method FLEX_SUBR (precedence level
3)
FLEXTYPE:
  AUTH (BILLFLD BILLNUM REPLACE) (FLEXLOG ) (OPERDISP SPL REPLACE)
  (CAINFLG AUTH) $
FLEXVAL:
  AUTH 1 6119999 110999
FLEXFEAT:
  110999 (REORGACT I_REO_AD_SD 0 15) (TRANSTS PARTITION 111 0)
  (PVSPDIDX 51) (CITYCODE 111) (CAINGRP AUTHSTGRP)
  (REORGTYP ONKEY STR P 15 5) $
CAINGRP:
  AUTHSTGRP 555 CAIN02 TCAP_SCCP (COLLINFO INFOCOLL SIOTRK) $ $ $
*****
-----
ApplyFtrs Event
-----
Collect Event
-----
Flexdial Index: C_ADDR
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
                  (INCLUDE) (SUBR) >>ADDR<<
-----
Start Event
-----
Processing Collectable:ADDR
-----
Parse Event
-----
Request Event
-----
*****
Peripheral Interaction
*****
Digits Required  : 1 (min), 6 (max)
Digit Set       : NONE
Signalling      : DTMF
(Use $ to terminate input without entering digits.)
Next par is: <Digits> STRING

```

FLEXSIM (continued)

```

Enter: <Digits>
>2211234

Received Digit Set : 2211234
Reporting: 2
Remaining: 211234
*****
End Peripheral Interaction
*****
Digits Buffer Digits :
Reset : NO Term : NO Timeout : NO CSP: NO
NOA : UNKNOWN_UNIQUENESS
CarrSel: NO_INDICATION
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
(INCLUDE) (SUBR) >>ADDR<<

-----
MsgAnalysis Event
-----
Next State:Load State
-----

Load Event
-----
*****
Peripheral Interaction
*****
Digits Required : 5 (min), 5 (max)
Digit Set : 211234
Signalling : DTMF
Reporting: 21123
Remaining: 4
*****
End Peripheral Interaction
*****
Digits Buffer Digits : 2
Reset : NO Term : NO Timeout : NO CSP: NO
NOA : UNKNOWN_UNIQUENESS
CarrSel: NO_INDICATION
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
(INCLUDE) (SUBR) >>ADDR<<

-----
MsgAnalysis Event
-----
Next State:Load State
-----

Load Event
-----
Digits Buffer Digits : 221123
Reset : NO Term : NO Timeout : NO CSP: NO
NOA : UNKNOWN_UNIQUENESS
CarrSel: NO_INDICATION
-----
Parse Event
-----

Validate Event

```

FLEXSIM (continued)

```

-----
*****Validate state data*****
pretranslator name:           DAL
pretranslation digits:       221123
pretranslator route selector: CT
call_feature:                ONNET
-----
Request Event
-----
*****
Peripheral Interaction
*****
Digits Required : 0 (min), 1 (max)
Digit Set : 4
Signalling : DTMF
Reporting: 4
Remaining: NONE
*****
End Peripheral Interaction
*****
Digits Buffer  Digits :
Reset : NO    Term : NO    Timeout : NO    CSP: NO
NOA : UNKNOWN_UNIQUENESS
CarrSel: NO_INDICATION
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
                  (INCLUDE) (SUBR) >>ADDR<<
-----
MsgAnalysis Event
-----
Next State:Load State
-----
Load Event
-----
Digits Buffer  Digits : 4
Reset : NO    Term : NO    Timeout : NO    CSP: NO
NOA : UNKNOWN_UNIQUENESS
CarrSel: NO_INDICATION
-----
Parse Event
-----
Validate Event
-----
*****Validate state data*****
pretranslator name:           DAL
pretranslation digits:       2211234
pretranslator route selector: CT
call_feature:                ONNET
*****
+++ CAIN +++ TDP Encountered: O_FTRREQ
*****
+++ CAIN +++ TDP Encountered: O_FTRREQ
*****
-----
ApplyFtrs Event
-----

```

FLEXSIM (continued)

```

*****ApplyFtrs state data*****
called_noa:                               SUBR
-----
Collect Event
-----
Flexdial Index: C_ADDR
Collectable List: (AGNTDATA) (AGNTDATA) (SUBR) (OLI) (MODNOA)
                  (INCLUDE) (SUBR) >>ADDR<<

*****ApplyFtrs state data*****
called_noa:                               SUBR
*****
+++ CAIN +++ TDP Encountered: INFOCOLL
*****
+++ CAIN +++ Trigger Match
TABLE: SIOTRK
      AUTHSTGRP ADDR 2211234 2211234 IGNORE $
      UserID: AXDAL274TWDTLS 11
      BearerCapability: SPEECH
      CalledPartyID: 2211234
TriggerCriteriaType: SIO_ADDR
      CallingPartyID: 02214000
*****
+++ CAIN +++ TDP Encountered: INFOCOLL
*****
+++ CAIN +++ Trigger Match
TABLE: SIOTRK
      AGTTSTGRP ADDR 2211234 2211234 QUERY SIO_ADDR $
      UserID: AXDAL274TWDTLS 11
      BearerCapability: SPEECH
      CalledPartyID: 2211234
TriggerCriteriaType: SIO_ADDR
      CallingPartyID: 02214000
*****
flag timeout
*****
Flexsim Execution Complete for Trunk: AXDAL274TWDTLS
*****

```

FLEXSIM (continued)**2** During simulation execution, the following logs are generated:

```

250Z      CAIN902 JUN10 13:34:15 3096 INFO CAIN SUBSCRIPTION METHOD
          PIC = COLLINFO
          TDP = NIL
          CAINGRP = AGTTSTGRP
          OTG = AXDAL274TWDTLS
          METHOD = AGENT (PRECEDENCE LEVEL 5)

250Z      CAIN902 JUN10 13:34:20 3197 INFO CAIN SUBSCRIPTION METHOD
          PIC = COLLINFO
          TDP = NIL
          CAINGRP = AUTHSTGRP
          OTG = AXDAL274TWDTLS
          CLGNO = 02214000
          FLEX_ACCESS = AUTH      1      6119999      110999
          SUBR_QUALIFIER = CAINFLG AUTH; 6119999
          METHOD = FLEX_SUBR (PRECEDENCE LEVEL 3)

250Z      CAIN903 JUN10 13:34:25 3298 INFO CAINGRP TO TRIGGER
          PIC = COLLINFO
          TDP = INFOCOLL
          TRIGGER = SIOTRK
          TRIGVIA = ADDR 2211234
          CAINGRP = AUTHSTGRP
          ACTION = IGNORE
          OTG = AXDAL274TWDTLS

250Z      CAIN903 JUN10 13:34:25 3399 INFO CAINGRP TO TRIGGER
          PIC = COLLINFO
          TDP = INFOCOLL
          TRIGGER = SIOTRK
          TRIGVIA = ADDR 2211234
          CAINGRP = AGTTSTGRP
          ACTION = QUERY
          OTG = AXDAL274TWDTLS

250Z      VAMP902 JUN10 13:34:25 3400 INFO VAMP OUTBOUND MESSAGE
KEY:      MSGSET=CAIN02, PROTO=TCAP_SCCP
CONXTX:  DEFAULT
TRID:     00 01 00 28
ROUTE:    GTTYPE=CAIN_CLID_GT, GTDIGS=214000
PACKAGE :  QWP
*** COMPONENT ***
COMP      :  INVL
COMP_ID   :  01
OPCODE    :  INFO_COLLECTED
*** PARAMETERS ***
USERID    :  CAIN SWID = 011 TRKGRP = 00274
BEARCAP   :  3_1KHZ
CHGNO     :  NOA = AUTH      NUMPLAN = PRIV      DIGITS = 6119999
TRIGCRIT  :  SIO_ADDR
CLGNO     :  NOA = NATL      NUMPLAN = ISDN      DIGITS = 214000
COLLADDR  :  NOA = UNKNOWN  NUMPLAN = UNKNOWN  DIGITS = 2211234
----- hex dump of message -----
E2 3B C7 04 01 00 28 00 E8 33 E9 31 CF 01 01 D1
02 64 02 30 28 BF 35 07 81 05 00 10 01 20 47 8D

```


FLEXSIM (continued)

01 01 93 06 E4 50 16 91 99 09 9F 34 01 6B 92 05
 03 10 12 04 00 96 06 80 00 22 11 32 04

250Z CAIN900 JUN10 13:34:25 3501 INFO SCP SIMULATOR INDICES
 TRANSID = 0001 0028
 CAINKEY = INFO_COLLECTED
 CAIN DALLAS AXDAL274
 SIO_ADDR
 CAINMTCH = 1497

250Z VAMP901 JUN10 13:34:27 3602 INFO VAMP INBOUND MESSAGE
 KEY: MSGSET=CAIN02, PROTO=TCAP_SCCP
 CONTXT: DEFAULT
 TRID: 00 01 00 28
 ROUTE: PC=01-08-01, SSN=33
 PACKAGE : RESP
 *** COMPONENT ***
 COMP : INVL
 COMP_ID : 5F 01
 OPCODE : ANALYZE_ROUTE
 *** PARAMETERS ***
 PRITRK : OPULSNO = N SWID = 011 TRKGRP = 00220

----- hex dump of message -----
 E4 1C C7 04 01 00 28 00 E8 14 E9 12 CF 02 5F 01
 D1 02 65 01 30 08 9F 2A 05 80 10 01 20 02

250Z CDR273 JUN10 13:34:28 3703 INFO CDR_CALL_ENTRY

RECCD	F0	TEMPLID	003
ACTIDX	0	TOOLGEN	Y
USEEDIT	N	VARLNGTH	N
SEQNUM	00224	CIC	
ORIGTIME	05655	ORIGDATE	161
ANSTYPE	00	TIMECHNG	N
ORIGAMPM	1	INCINTL	
DISCTIME	05667	DISCDATE	161
ANISUFF		DISCAMPM	1
INCBILL		CALLDUR	
PINDIGS		ANISP	
INFODIG		BILLNUM	6119999
ACCTCD		CLGPTYNO	
RLTCDR		ACCTV	0
UNIVACC		DIALEDNO	2211234
CALLEDNO	2211234	OUTPULNO	
ORIGOPRT		PREDIG	0
CNPREDIG	0	OPART	
TPART		QUEUED	N
RTELIST		RTENO	
NETSEC	N	OPCHOICE	
COLLTIME	010	ADIN	
LNPCHK		NUMWBCKT	
WBCKTS		FINSID	
BILLTYPE		ORIGGRP	0274
DISCTYPE	3	ORIGMEM	0001
PASSTHRU	0	TERMGRP	
PRESIND		TERMMEM	
OSRASSOC	N	DIGDATA	N
FINTKGRP		TRAP	N
COSOVE	N	FINTKMEM	

FLEXSIM (end)

SCPBILL		TDP	03
PIC	03	TRIGGER	05
TRMTCD	000	COMPCODE	0
CRID		DNIS	
ORIGPVN		TERMPVN	
ORIGLRN		PORTEDNO	
CAINCT		ACG	N
DNISNOA		DCR	N
MLTCOSID		COSINDEX	
CARRSEL			

Appendix A

Engineering guidelines

This appendix contains the engineering guidelines you should consider when provisioning your switch for CAIN/FlexDial interaction.

General engineering rules for CAIN

Consider the following when provisioning your switch:

- CAIN groups should be used judiciously. Where possible multiple services/triggers should be combined into a single group.
- Engineer CAIN extension block parameters when introducing NetworkBuilder for the first time. Both CAIN and TCAIN services need to be considered.
- High traffic services such as N00 should use global titles to route to a dedicated SCP via a dedicated linkset. This will help avoid SS7/SCP congestion.
- When converting a subscriber address for use by CAIN, it may be advisable to datafill the number as CT OFFNET in table STDPRTCT to minimize feature interactions.
- For standardized digit collection, the *O_Feature_Requested* trigger is much more efficient than conversation.
- Using CAIN on SS7 originators will increase the holding time for ISUP extension blocks. These blocks are engineered in table OFCENG (NUM_ISUP_EXT_BLKs).
- The switch allows a maximum of 32,768 permutations of CAINGRP/DIGTYPE combinations in the trigger tables. For example, each of the following tuples (from table SPECDIG) count as one permutation:
 - > CAINGRP2 ADDR 214 214 QUERY SDS_ADDR TREAT \$
 - > CAINGRP2 INFO 11 11 QUERY SDS_INFO ROUTE \$
 - > CAINGRP2 ADDR 800 800 QUERY SDS_N00 ROUTE \$
 - > CAINGRP2 ADDR 011 011 IGNORE \$

- > CAINGRP2 ANI 684 684 BLOCK \$
- > CAINGRP2 CIC 333 333 QUERY SDS_CIC TREAT

Refer to the *UCS DMS-250 NetworkBuilder Application Guide* for additional engineering guidelines.

General engineering rules for FlexDial

Engineering your switch for FlexDial requires thorough knowledge of the FlexDial Framework and the following tables:

- FLEXDIAL
- FLEXTYPE
- MSGCTR
- FLEXFEAT
- FLEXVAL
- TRKFEAT

Refer to the *UCS DMS-250 FlexDial Framework Application Guide* for more details.

General engineering rules for CAIN/FlexDial Interactions

Remember, by using the full interaction capabilities of CAIN and FlexDial, your switch will require more resources than only one of the applications.

Limitations and restrictions

CAIN/FlexDial interaction limitations and restrictions include, but are not limited to the following:

- The Programmable Service Node (PSN) is not supported.
- In order for the BILLNUM field to be populated in the CDR and for CAIN to recognize a number as the billing number, you must provision the BILLFLD BILLNUM option (table FLEXTYPE) for non-ANI billing numbers. For example, when billing to an authorization code or travel card. The same is true for the following CDR fields:
 - ANISP
 - PINDIGS
 - ACCTCD
 - SUBRNUM1
 - SUBRNUM2
- Delivery of the `adin` extension parameter to the SCP is not supported.
- FlexDial PANI construction:

- The SCP must be able to process this PANI (delivered in the *CallingPartyID* parameter)
- If the PANI is not suitable for global title translations, provision CAIN_ADDR_GT, CAIN_FEAT_GT for global title translations.

Note: The FlexDial Framework does not differentiate between ANIs and PANIs. Therefore, if the CAIN_CLID_GT option is used for global title translations with an invalid national number, global title translations will fail and the query is not delivered to the SCP.

- Precedence Level 3 and 4 subscription is only available for inswitch validation through table FLEXVAL. Subscription is not available for MATCH validations.
- The *Origination_Attempt* message will never contain the *Carrier* or *ChargeNumber* parameters. FlexDial does not perform any digit collection prior to **Collect_Information**. Therefore, the *Carrier* and *ChargeNumber* are not available at *Off_Hook_Immediate*.
- When a call queries the SCP from *Off_Hook_Immediate* and the SCP responds with an **Analyze_Route**, the *ChargeNumber* parameter must be included in the **Analyze_Route**.
- When the FlexDial Framework collects the equivalent of an authorization code or travel card number, the BILLFLD option must be provisioned in table FLEXTYPE.
- When the switch generates a FLEX302 log (similar to TRK logs showing valid treatment) associated with a call originating on a CAIN-capable AXXESS agent, CAIN call processing suppresses the log until determining that the call will not trigger.
- If any subscriber number fails validation, re-validation is unavailable on all reoriginations.
- If a FlexDial deferred reset is active when a FlexDial ADDR collectable is executed, call processing does not trigger at *O_Feature_Requested*.
- For AXXESS agents mimicking DALs, typical provisioning constructs a default PANI that can be overwritten by a collected PANI. This default PANI satisfies outpulsing requirements to DS1 multi-stage outpulsers when a PANI is not collected for a call. Therefore, in this scenario, the *CalledPartyID* would contain the default PANI.
- Full implementation of the UCS11 CAIN software requires the CAIN_PROTOCOL_STREAM parameter to be set to UCS11 and the CAIN_PROTOCOL_VERSION parameter to be set to V0 or higher. Both parameters are provisioned in table CAINPARAM.

Appendix B

List of terms

advanced intelligent network

A network that allows the switch to off-load some of the call processing functions to an intelligent service control point (SCP).

AIN

advanced intelligent network

AIND

AIN disconnect treatment

AINF

AIN final treatment

ANI

automatic number identification

ANISCUSP

Automatic Number Identification Screening Customer Profile table. May be used to assign a CAIN group to a particular ANI.

AUTHCDU2

Authcode Database 2 table. May be used to assign a CAIN group to a particular authorization code.

AUTHCDU3

Authcode Database 3 table. May be used to assign a CAIN group to a particular authorization code.

AUTHCDU4

Authcode Database 4 table. May be used to assign a CAIN group to a particular authorization code.

AUTHCDU5

Authcode Database 5 table. May be used to assign a CAIN group to a particular authorization code.

AUTHCODU

Authcode Database table. May be used to assign a CAIN group to a particular authorization code.

authorization code

A unique multidigit code that identifies an authorized subscriber. Authorization codes are usually 5-7 digits (due to UCS DMS-250 limitations) and identify a subscriber, bill a call, prevent unauthorized network use, determine the originating caller's class of service, and control access to special features. For example, a caller can be required to enter an authorization code to retrieve voice mail messages.

automatic number identification

billing number for the calling party provided to the IEC from the LEC

BC

bearer capability

bearer capability

A characteristic associated with a directory number to indicate the type of call (voice or data) and the rate of transmission allowed.

C7GTT

CCS7 Global Title Translation table. Maps a translation type (defined in table C7GTTTYPE) to a CCS7 network address.

C7GTTTYPE

CCS7 Global Title Type table. Maps a CCS7-defined translation to a network-defined global title translation type.

C7LINK

CCS7 Link table. Makes the association between the physical equipment of the link and the logical view of the link as a member of a linkset.

C7LKSET

CCS7 Link Set table. Defines the characteristics of a linkset. A linkset is a set of links used as a group. Each link carries traffic between the origination point code and a destination point code. The table also defines attributes that are common to all links in the link set. The links are defined in table C7LINK.

C7LOCSSN

CCS7 Location Subsystem Number table. Defines the subsystems located on the switch.

C7NETSSN

CCS7 Network Subsystem Number table. Provides the set of remote point codes (PC) and subsystems, at the remote PCs, where messages are routed by the SCCP. A PC is a node in the CCS7 network that may be an SSP, an STP, or an SCP.

C7NETWRK

CCS7 Network table. Describes the signaling networks in use in a switching office.

C7RPLSSN

CCS7 Replicate Subsystem table. Provides the set of remote subsystem replicate pairs. It has a one part key, the subsystem name. For each subsystem a list of PC pairs at which the replicated subsystems reside must be given.

C7RSSCRN

CCS7 Remote Subsystem Concerned Node table. Provides a list of concerned nodes for a remote subsystem point code combination. The table has a two part key. The first part is the PC and the second part is the subsystem name. The PC and subsystem combination must be datafilled in table C7NETSSN.

C7RTESET

CCS7 Route Set table. Associates linksets used as possible routes for each signaling point in the network. An office point code identifies a signaling point within any network. Each office point code must have a routeset. The information in this table records which routes and linksets can carry the signaling information to the destination signaling point. This table is also used for alternate routing decisions.

CAIN routing parameters

Parameters provided in an **Analyze_Route** response. The parameters are: *PrimaryTrunkGroup*, *AlternateTrunkGroup*, *SecondAlternateTrunkGroup*, *CalledPartyID*, and the *GenericAddressList*'s *OverflowRoutingNo*.

CAINCONV

CAIN Conversation table. Controls SCP simulator interaction during conversation with the switch.

CAINGRP

CAIN Group table. Identifies CAIN groups and trigger sets used for CAIN subscription.

CAINKEY

CAIN Key table. Determines a range of possible responses for a defined three-part key into the SCP simulator. The range of possible responses is represented by an option vector of indexes into table CAINMTCH.

CAINMTCH

CAIN Matching table. Screens and selects possible responses for the SCP simulator

CAINPARAM

CAIN Parameters table. Assigns CAIN office parameter values.

CAINRESP

CAIN Response table. Contains response data to return to the switch. The SCP simulator's encoder takes this data and builds a Transaction Capabilities Application Part (TCAP) message.

CAINRSRC

CAIN Resource table. Maps the data returned from the SCP to a resource available on the switch.

CAINUID

CAIN User Identification table. Provides symbolic names for trunk groups and switches used in the simulator. It is similar to table CLLI in function. The use of symbolic names rather than numbers provides enhanced clarity when datafilling the simulator tables.

CAINXDFT

CAIN Extension Parameter Defaults table. Defines default values for five extension parameters: `servTranslationScheme`, `callType`, `satRestriction`, `classofSvc`, and `callBranding`.

call branding

Tones or announcements (returned from the SCP or provisioned in table CAINXDFT) played by the switch as directed by CAIN call processing before routing is attempted.

call detail record

Formatted billing data used to generate subscriber billing.

call model

generic representation of a basic call in terms of the processing activities required to establish, maintain, and clear a call.

call processing

The function of call processing software is to establish connections among telephony agents. A number of functional steps are required to process a call, such as detecting the incoming call, receiving digits, analyzing (translating) digits to determine call destination, selecting terminating agent, establishing connection, signaling to and detecting an answer from the terminating agent, and detecting disconnect.

CALLATTR

Call Attributes table. Used to provision a PRI call attribute.

called number

The number of the party receiving the call. Also known as called party ID.

called party

The end user that receives a call.

calling line identification

In data transmission, a feature provided by the network that allows a called terminal to be notified by the network of the address from which the call was originated.

calling number

The number of the party initiating the call. This number can identify the origin of a call to the called party. Also known as calling party number or calling party ID.

calling party

The end user that originates a call.

carrier identification code

Three- to four-digit number that identifies which interexchange carrier a call will use. Subscribers can dial these digits with each long distance call, or can pre-subscribe to a particular carrier and let the digital switch software add the CIC.

CCS7

Common Channel Signaling No. 7

CDR

call detail record

CIC

carrier identification code

CLI

calling line identification

CLLI

common language location identifier

CM

computing module

common channel signaling No. 7

A digital, message-based network signaling standard defined by the CCITT which separates call signaling information from voice channels so that interoffice signaling is exchanged over a separate signaling link.

common language location identifier

A standard identification method for trunk groups, tones, and announcements.

computing module

The processor and memory of the dual-plane combined core used by the DMS SuperNode. Each CM consists of a pair of central processing units (CPU) with associated memory that operate in a synchronous matched mode on two separate planes. Only one plane is active; it maintains overall control of the system while the other plane is on standby.

conversation package

A transaction capabilities application part (TCAP) package expecting a reply. It can be sent by the service control point (SCP), or adjunct, to the service switching point (SSP) that instructs the SSP to perform a user interaction, collect digits, and send a reply to the SCP. The SSP can also send a conversation package to the SCP.

COS

class of service

CPID

calling party identification

CUSTDP

Customized Dialing Plan table. Defines trigger criteria for *Customized_Dialing_Plan*.

DAL

dedicated access line

dedicated access line

Network connection, often leased from a local exchange carrier or competitive access provider, that provides a direct link from a customer to the long distance network. Typical DALs include outbound WATS lines, PBX tie trunks, and foreign exchange lines.

digital recorded announcement machine (DRAM)

A peripheral module (PM), developed for the DMS switch, in which voice messages are stored in digital form, providing access to up to 30 different service voice announcements.

dual-tone multifrequency (DTMF) signaling

A signaling method employing set combinations of two specific voice-band frequencies, one of which is selected from a group of four low frequencies, and the other from a group of three or four relatively high frequencies.

EANT

equal access network trunking

fast interdigit (FIDT) timer

time allowed for the subscriber to dial digits between the minimum and maximum required

EDP

event detection point

event detection point (EDP)

A point in basic call processing, as modeled by the basic call model (BCM), which can be armed by the receipt of a **Request_Report_BCM_Event** message from the SCP/Adjunct. EDPs are located at transitions between Points In Call (PICs). EDPs can be armed as Requests (EDP-Rs) or Notifications (EDP-Ns). When the request event occurs, the switch reports it to the SCP/Adjunct.

event

Events are armed by the **Request_Report_BCM_Event** non call-related component when received in a response from the SCP along with an **Analyze_Route, Continue, or Collect_Information** message. During a call, the switch may encounter requested (armed) events at EDPs. An event is used to determine the action to take for the call.

FCC

Federal Communications Commission

Federal Communications Commission

An agency of the U. S. Government that regulates standards and companies within all aspects of the communications industry (radio, television, telephony).

FGD

feature group D

FIDT

fast interdigit timer

FlexDial

FlexDial Framework is a UCS DMS-250 feature that allows you to program the call origination side of a call to implement customized functionality. UCS06 software does not support CAIN/FlexDial interaction.

GDP

generic digits parameter

generic digits parameter

ISUP parameter used to transport generic digits with a specified identifier tag

global title

An application address. The SCCP global title translation (GTT) function is required to translate a GT into a valid network address.

global title translation

The process that translates an application-specific address (such as a dialed 800 number) into the Common Channel Signaling No. 7 (CCS7) PC subsystem address, usually that of the appropriate service control point (SCP).

GT

global title

GTT

global title translation

hotline

A connection that has another address mapped to it. This second address can be filed in a table, and therefore not changeable by the end user, or it can be operated by an authcode for an effect similar to speed dialing.

IAM

initial address message

IN

intelligent network

IN/1

UCS proprietary intelligent network protocol for offboard databases with limited call control

IEC

interexchange carrier

IMT

intermachine trunk

information digit

Digit received by the switch either in the incoming digit stream or with the incoming signaling information. An information digit carries additional information about the call that cannot be determined by the address digits. The type of information digit received and the method it is received depends on the incoming signaling system.

Integrated Services Digital Network

A network that provides end-to-end digital connectivity using CCS7 to support a wide range of voice and data services to the end-user.

intelligent peripheral

Contains functionality and resources for exchanging information (such as voice announcements and dual-tone multifrequency digit collection) with a subscriber.

intermachine trunk

A trunk that connects the UCS DMS-250 IEC networks.

IP

intelligent peripheral

ISDN

Integrated Services Digital Network

ISDN User Part

SS7 protocol that defines the messages, parameters, and procedures to set up and tear down all circuit switched calls, both ISDN and non-ISDN, in U.S. SS7 networks. It includes support for ISDN Supplementary Voice services and interworks with Q.931/932 to provide end-to-end ISDN.

ISUP

ISDN User Part

LEC

local exchange carrier

MCCS

mechanized calling card service

mechanized calling card service

The service that allows a call to be billed to a calling card number.

message parameters

Parameters within a message. Each message type has its own set of mandatory and optional parameters.

message set

the set of standard messages required by an advanced intelligent network (AIN) specification, such as Bellcore's Release 0.2, for communication among AIN network elements.

message switch and buffer

A peripheral module used by the switch, along with a signaling terminal, to act as an interface to and operate within a common channel signaling environment. The message switch and buffer supports the signaling terminal and routes the messages received by the signaling terminal through the network module to the digital trunk controller. The message switch and buffer also receives messages sent from central control and routes them to the signaling link through the signaling terminal. A different configuration of the message switch and buffer exists for each of the two protocols used to implement common channel signaling.

message switch and buffer 7

The message switch and buffer (MSB) for Common Channel Signaling 7 (CCS7) protocol. *See also* message switch and buffer (MSB)

MSB

message switch and buffer

MSB7

message switch and buffer 7

NETBUSY

Network Busy table. Defines trigger criteria for *Network_Busy*.

numbering plan area

Any of the designated geographical divisions of the United States, Canada, Bermuda, and Northwestern Mexico within which no two telephones have the same seven-digit number. Each NPA is assigned a unique three digit area code for World Zone 1 dialing.

OCLDBUSY

O Called Party Busy table. Defines trigger criteria for *O_Called_Party_Busy*.

OFFCCODE

Office Code table. Defines trigger criteria for *Office_Code*.

OFFHKIMM

Off Hook Immediate table. Defines trigger criteria for *Off_Hook_Immediate*.

off-hook

The condition existing in telephone operations when the receiver or handset is removed from its hookswitch.

OFTRREQ

O Feature Requested table. Defines trigger criteria for *O_Feature_Requested*.

OIECREO

Origination Inter-Exchange Carrier Reorigination table. Defines trigger criteria for *O_IEC_Reorigination*.

OM

operational measurements

ONOANSWR

O No Answer table. Defines trigger criteria for *O_No_Answer*.

operational measurements

The software resources of the switch that control the collection and display of measurements taken on an operating system. The OM subsystem organizes the measurements data and manages its transfer to displays and records. The OM data is used for maintenance, traffic, accounting, and provisioning decisions.

PANI

pseudo-automatic number identification

parameter

Data contained within a message.

partial dial (PDIL) timer

time allowed between each subscriber dialed digit (before minimum number of digits are dialed)

PBX

private branch exchange

PC

point code

PCL

product computing module loads

PDIL

partial dial timer

per-trunk signalling

A conventional telephony method of signaling that multiplexes the control signal of a call with voice or data over the same trunk (in-band signaling).

permanent signal timer

time allowed before the subscriber enters the first digit

personal identification number

Authorization number (usually composed of the caller's telephone number plus a four-digit code) that allows subscribers to access their long distance carriers when away from home.

PIN

personal identification number

PIC

point in call

point in call

A generic representation of a sequence of switch based call processing actions considered essential to establish, maintain, or clear a two-party call. PICs are separated by trigger detection points (TDP).

PRI

primary rate interface

PRIBCHNL

PRI B-CHANNEL table. Defines trigger criteria for *PRI_B-Channel*.

primary rate interface

An interface that carries nB+D channels over a digital DS-1 facility (23B+D in North America and 30B+D in Europe). PRI is used to link private networking facilities, such as private branch exchanges (PBX), local area networks (LAN), and host computers with a standardized architecture acting as the bridge between private switching equipment and the public network. Formerly known as primary rate access.

private branch exchange

A private telephone exchange, either automatic or attendant operated, serving extensions in an organization and providing access to the public network.

pseudo automatic number identification

A 10-digit translations code derived from a combination of the authorization code, personal identification number, and serving number plan area number.

PSIG

permanent signal timer

PSN

programmable service node

PTS

per trunk signaling

query

A type of communication message sent by the service switching point (SSP) to the service control point (SCP), or adjunct, requesting call processing instructions. In AIN Release 0.2, the message is contained in a transaction capabilities application part (TCAP) query package.

REL

CCS7 release message

response package

A transaction capabilities application part (TCAP) package containing one or more messages sent in response to another TCAP package. The service control point (SCP), or adjunct, can use a response package to instruct the service switching point (SSP) to perform an activity. The SSP can also send response packages.

response processing

The service switching point (SSP) receives and processes response messages from the service control point (SCP), or adjunct. There are several different types of response messages, each of which is handled differently according to the information it contains.

SCCP

signaling connection control part

SCE

service creation environment

SCP

service control point

service control point

A node in a Common Channel Signaling No 7 (CCS7) signaling network that supports application databases. The function of an SCP is to accept a query for information, retrieve the requested information from one of its application databases, and to send a response message to the originator of the request.

service switching point

A switch that is capable of interacting with the Common Channel Signaling No. 7 (CCS7) network databases. It contains hardware to support CCS7 signaling, software to control call processing and also create network database query messages, and software to interpret network database response messages.

serving translation scheme

The scheme the UCS DMS-250 switch uses to translate and route a call. STS codes are three digits codes (000-999). The switch uses the three digits serving translation scheme codes (000-999) to derive routing information.

signal transfer point

A switch that is used to provide signaling link connections between switches. That is, it is a tandem node for Common Channel Signaling No. 7 (CCS7) signaling links and contains hardware to support CCS7 hardware and software to route CCS7 messages. It does not contain any software to create or interpret CCS7 messages. STPs are deployed in pairs. If one STP fails, the mate takes over, ensuring that service continues without interruption. One of the STPs primary functions is performing global title translations.

signaling

Communication between switches, or switches and end points, to set-up, manage, and tear-down calls. Signaling methods include dial pulse (rotary dial), dual-tone multifrequency (DTMF) (touch-tone), and digital "packet" technology (ISDN, SS7)

Signaling Connection Control Part

A level of common Channel Signaling No. 7 (CCS7) layered protocol. The main functions of the SCCP include the transfer of signaling units with or without the use of a logical signaling connection at the provisioning of flexible translations (GTT) for different applications.

SIOTRK

Shared Interoffice Trunk table. Defines trigger criteria for *Shared_Interoffice_Trunk*.

SMS

service management system

SOC	software optionality control
SPECDIG	Specific Digit String table. Defines trigger criteria for <i>Specific_Digit_String</i> .
SPECFEAT	Specific Feature Code table. Defines trigger criteria for <i>Specific_Feature_Code</i> .
SS7	signaling system number 7
SSP	service switching point
STP	signal transfer point
STR-Connection	A connection that is made from an SSP to an IP over an ISDN IP interface in response to a Send_To_Resource message.
SUS	CCS7 suspend message
TANDMRTE	Tandem Routing table. Provisions routing through tandem switches within the IEC network to reach the required terminating switch.
TCAP	transaction capabilities application part
TDP	trigger detection point
TERMRTE	Termination Routing table. Provisions routing to a terminating network directly connected to the current network.
Transaction Capabilities Application Part	A service that provides a common protocol for remote operations across the Common Channel Signaling No. 7 (CCS7) network. The protocol consists of message formatting, content rules, and exchange procedures.

translation verification

A diagnostic tool that allows the operating company to access and simulate a telephone call in software and display the tables and tuples used to establish the lines, trunks, or positions to which a call is routed.

TRAVER

translation verification

TRID

transaction identifiers

trigger

A trigger defines the actions taken once trigger criteria is met at a TDP on the SSP.

trigger criteria

Trigger criteria defines the conditions used to determine whether a particular call will trigger. A trigger may contain criteria of one or more trigger criteria types. In order for a call to trigger, all trigger criteria must be met.

trigger detection point

A point in basic call processing, as modeled by the basic call model (BCM), which identifies when a service control point (SCP) can receive notification of a given event and influence subsequent call processing. TDPs are located at transitions between points in call (PIC).

trigger tables

Trigger tables store information about the advanced intelligent network (AIN) application, triggers, trigger detection points, trigger criteria, transport protocol.

triggering

The process where a call indicates that it requires advanced intelligent network (AIN) service(s).

TRKGRP

Trunk Group table. Assigns a CAIN group to a particular agent.

VAMP

variable AIN messaging platform

VAMPTRID

VAMP Transaction Identifiers table. Provisions the key resources used in Carrier AIN messaging, including transaction and component identifiers and message buffers.

VPN

virtual private network

Ordering information

Use the following table for ordering Nortel NTPs (Northern Telecom Publications) and Product Computing-Module Loads (PCLs):

Type of product	Source	Phone	Cost
Technical documents (paper or CD-ROM)	Nortel Product Documentation	1-877-662-5669, Option 4 + 1	Yes
Individual NTPs (paper)	Merchandising Order Service	1-800-347-4850	Yes
Marketing documents	Sales and Marketing Information Center (SMIC)	1-800-4NORTEL (1-800-466-7835) * ESN 444-5930	No
PCL software	Nortel	Consult your Nortel sales representative * Employee	Yes

When ordering publications on CD

Please have the CD number and software version available, for example, **HLM-2631-001 02.02**.

When ordering individual paper documents

Please have the document number and name available, for example, **297-2631-001, UCS DMS-250 Master Index of Publications**.

When ordering software

Please have the eight-digit ordering code, for example, **UCSE0009**, as well as the ordering codes for the features you wish to purchase. Contact your Nortel representative for assistance.

Digital Switching Systems
UCS DMS-250
CAIN/FlexDial Interactions

Product Documentation—Dept 3423
Northern Telecom
P.O. Box 13010
RTP, NC 27709-3010
1-877-662-5669, Option 4 + 1

© 1997-1999 Northern Telecom
All rights reserved

NORTHERN TELECOM CONFIDENTIAL: The information contained in this document is the property of Northern Telecom. Except as specifically authorized in writing by Northern Telecom, the holder of this document shall keep the information contained herein confidential and shall protect same in whole or in part from disclosure and dissemination to third parties and use same for evaluation, operation, and maintenance purposes only.

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

DMS, MAP, NORTEL, NORTEL NETWORKS, NORTHERN TELECOM, NT, and SUPERNODE are trademarks of Northern Telecom.

Publication number: 297-2621-372
Product release: UCS11
Document release: Standard 04.02
Date: May 1999
Printed in the United States of America

NORTEL
NORTHERN TELECOM