AM-TR-PSS-000032

Ameritech ACD/MIS Service Network Interface Specifications

To:	N/A
Priority:	N/A
Effective Date:	February 1990
Issue Date:	Issue 1, February 1990
Expires On:	N/A
Training Time:	N/A
Related Documents:	N/A
Canceled Documents:	N/A
Issuing Department:	N/A
Distribution:	NA
Business Unit:	N/A

Points of Contact:
N/A

Author(s):
N/A

Table of Contents

TECHNICAL REFERENCE NOTICE

This Technical Reference is published by Ameritech to provide a technical description of the Ameritech ACD Management Information System.

Ameritech reserves the right to revise this document for any reason, including but not limited to conformity with standards promulgated by various agencies, utilization of advances in the state of the technical arts, or the reflection of changes in the design of any equipment, techniques or procedures described or referred to herein.

AMERITECH MAKES NO REPRESENTATION OR WARRANTY, EXPRESS OR IMPLIED, WITH RESPECT TO THE SUFFICIENCY, ACCURACY, OR UTILITY OF ANY INFORMATION OR OPINION CONTAINED HEREIN. AMERITECH EXPRESSLY ADVISES THAT ANY USE OF OR RELIANCE UPON THIS TECHNICAL REFERENCE IS AT THE RISK OF THE USER AND THAT AMERITECH SHALL NOT BE LIABLE FOR ANY DAMAGE OR INJURY INCURRED BY ANY PERSON ARISING OUT OF THE SUFFICIENCY, ACCURACY, OR UTILITY OF ANY INFORMATION OR OPINION CONTAINED HEREIN.

This document is not to be construed as a suggestion to any manufacturer to modify or change any of its products, nor does this document represent any commitment by Ameritech or any Ameritech operating company (AOC) to purchase any product, whether or not it provides the described characteristics.

Ameritech does not recommend products, and nothing contained herein is intended as a recommendation of any product to anyone.

Nothing contained herein shall be construed as conferring by implication, estoppel, or otherwise any license or right under any patent, whether or not the use of any information herein necessarily employs an invention of any existing or later issued patent.

Ameritech reserves the right not to offer any or all of these services and to withdraw any or all of them at any future time.

Document may be ordered from Ameritech by contacting the Document Order Center at (847) 248-4324.

1. Introduction

1.1. General

This document provides a description of the application message protocol to be used for the ACD Management Information System Data Stream when used with a Northern Telecom DMS-100 switch. It also provides a description of the messages (and their contents) used to implement the protocol.

The protocol described in this document may change during the evolution of the ACD Management Information System Data Stream protocol.

1.2. Scope

The protocol specified in this document enables a down stream processor (DSP) to collect ACD information from a Northern Telecom DMS-100 switch. The information can be used to provide real time statistics and historical reports.

1.3. Document Overview

- Sections 2, 3, and 4 respectively provide information on the standards used in this document, abbreviations used, and the protocol architecture.
- Section 5 outlines the requirements of the ACD MIS Data Stream protocol.
- Section 6 gives a brief overview of X.410 Operation Protocol Data Units (OPDUs).
- Section 7 outlines the Remote Operations that are part of the protocol defined in this specification.
- Section 8 contains state transition diagrams that specify when ROs can successfully execute.
- Section 9 contains time sequence diagrams that illustrate the flow of control of RO OPDUs between the DSP and the switch.
- Section 10 describes the meaning and usage of the AcdGroupData, ACDSuppleDnData, AgentPositionData, and AcdEvent arguments in the switch originated ROs.
- Section 11 presents the formal RO and type definitions for the ACD Management Information System Data Stream protocol.

- Section 12 outlines implementation restrictions placed on the ACD MIS application protocol.

- Section 13 contains some examples of ACD MIS ROs.

- Section 14 contains examples of ACD event scenarios and the ACD MIS ROs that will be generated as a result.

2. References

2.1. International Standards

OSI - CCITT Recommendation X.200; Reference Model of Open Systems Interconnection of CCITT applications

X.25 - CCITT Recommendation X.25

2.2. Telcordia (formerly Bellcore) Documents

X.25 Telcordia (formerly Bellcore) Pub. 54001, Bx25 Issue 3, Addendum A

2.3. Northern Telecom Documents

NOP Protocol - NIS-D301-1, Issue 1.1; Network Operations Protocol Interface Specification; October 1986

X.410 - NIS-D301-1; Remote Operations Protocol; DEC. 1985

X.409 - NIS-D301-1; OSI Abstract Syntax Notation and Basic Encoding Rules; Dec. 1985

NTP 297-1001-114 - Northern Telecom Practice; Operational Measurements; July 1984

3. Symbols and Abbreviations

ACD - Automatic Call Distribution

ACD MIS - ACD Management Information Systems

ACD-DN - ACD Directory Number

CCITT - International Telegraph and Telephone Consultative Committee

DSP - Down Steam Processor

OPDU - Operation Protocol Data Unit

OSI - Open Systems Interconnection

RO - Remote Operation

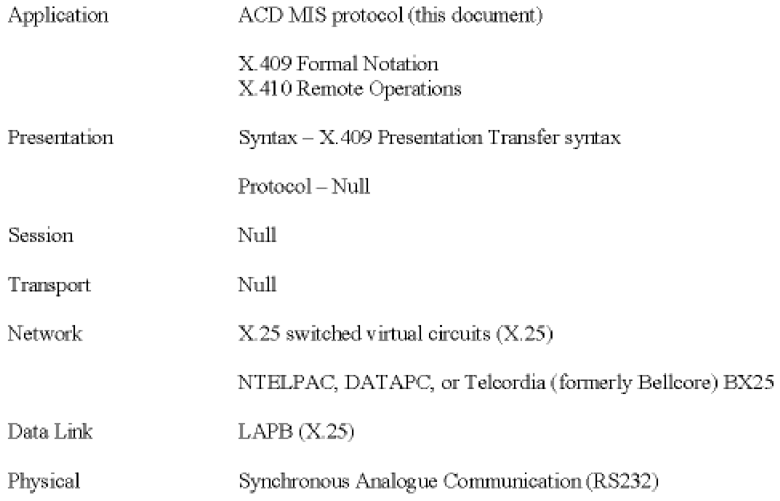
SVC - Switched Virtual Circuit

Switch - DMS-100/MSL-100

RTS - Reliable Transfer Server

4. Architecture

The ACD Management Information System (ACD MIS) Application Protocol is a sub-layer of the application layer of the OSI model. It runs on top of the Remote Operations application sub-layer. There are no dependencies on Presentation, Session, or Transport layer functions. For this reason, this protocol uses null instances of these layer protocols, although the X.409 Presentation Transfer syntax is used. The protocols used to support the ACD MIS Application Protocol are summarized below:



Application	ACD MIS protocol (this document)
	X.409 Formal Notation X.410 Remote Operations
Presentation	Syntax – X.409 Presentation Transfer syntax
	Protocol – Null
Session	Null
Transport	Null
Network	X.25 switched virtual circuits (X.25)
	NTELPAC, DATAPC, or Telcordia (formerly Bellcore) BX25
Data Link	LAPB (X.25)
Physical	Synchronous Analogue Communication (RS232)

5. ACD MIS Data Stream Requirements

The ACD MIS Data Stream protocol is required to support the following configuration:

1. Multiple ACD MIS Data Streams per switch; each data stream on its own SVC.

2. One ACD customer per data stream; hence, one ACD customer per SVC.

3. Multiple ACD customers per data stream; hence, multiple ACD customers per SVC.

The AC MIS Data Stream protocol is not required to support multiple data streams per SVC.

Before multiple ACD MIS Data Streams can be set up in the switch, the ACD configuration must be partitioned between the data streams so that the switch knows which data stream the data for a particular ACD group is to be sent to. The ACD group partitioning between data streams is accomplished by defining “pools” of ACD groups within the switch. The rules for setting up and using these pools of ACD groups are:

1. An ACD group can only belong to one pool at any one point in time.

2. A pool can only be associated with one data stream at any one point in time.

3. Only one pool of ACD groups can be associated with each data stream at any one point in time.

4. When setting up an ACD MIS Data Stream session, the DSP must specify the pool of ACD groups (with password security) that is to be associated with the data stream (SVC).

By enforcing these rules, the switch will know which data stream (SVC) the data for a particular ACD group is to be sent to.

Switch security for the ACD MIS Data Stream is required regardless of whether there is one ACD customer or multiple ACD customers per data stream. Also, because the switch is required to support multiple ACD customers per data stream, the switch must also strictly enforce boundaries between the customers to ensure customer security.

The switch security and ACD customer security is accomplished by defining “subpools” of ACD groups within a pool of ACD groups. Typically, all ACD groups belonging to an ACD customer will be grouped within one subpool. If there is only one ACD customer on a data stream there will probably only be one subpool defined within the pool of ACD groups associated with the data stream; however, there can be more than one. In the situation where there are multiple ACD customers on a signal data stream, multiple subpools will be defined.

The rules for setting up and using subpools of ACD groups are:

1. The ACD groups in subpool must belong to the same pool.

2. An ACD group can only belong to one subpool at any one point in time.

On the following pages, some examples of Switch/DSP configurations are presented to help clarify the rules outlined in the paragraphs above.

Figure 1. Single Customer Per DSP: This example shows two ACD customers; each on their own ACD MIS Data Stream, and hence their own SVC. There is a single subpool defined within each pool.

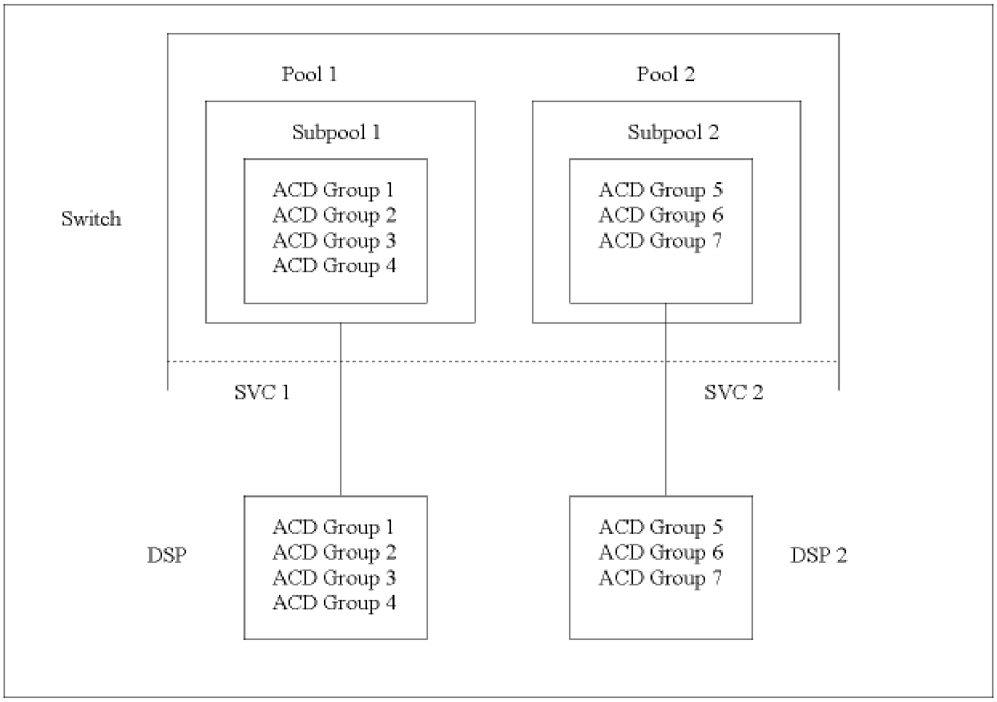
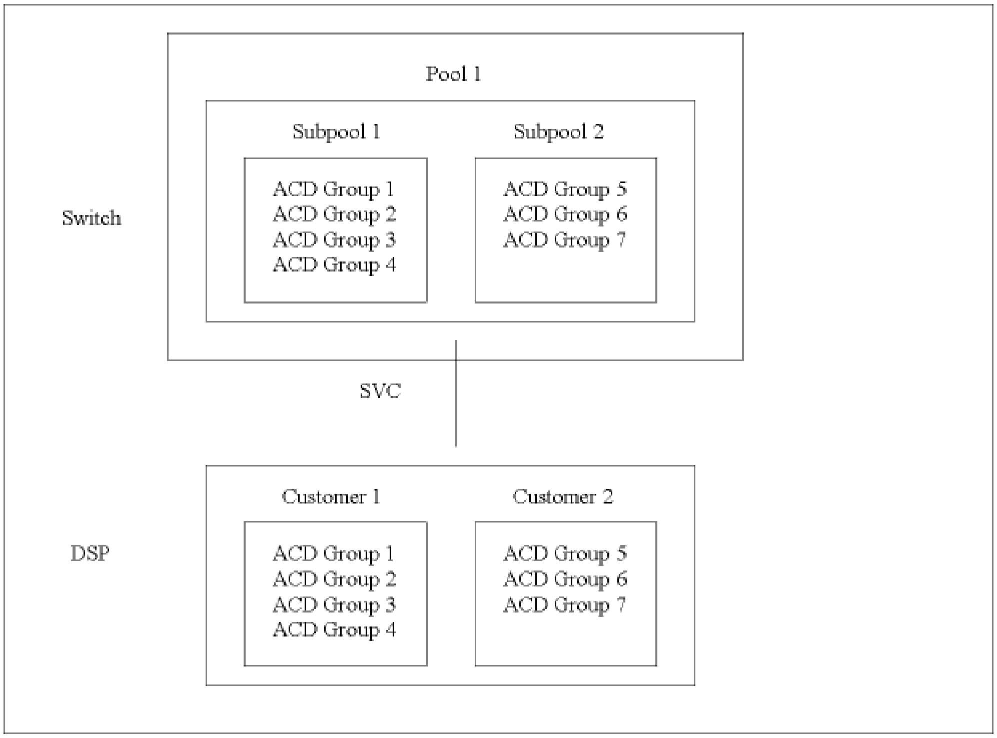


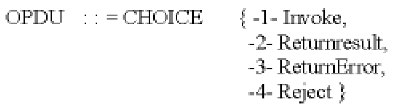
Figure 2. Multiple Customers Per DSP: This example shows two customers on the same DSP using one ACD MIS Data Stream. Two subpools are defined so that the customers' ACD configuration can be kept secure in the switch.



6. X.410 Operation Protocol Data Unit Overview

Before presenting the details of the ACD MIS Data Stream Protocol, a brief overview of the X.410 OPDUs will be presented.

Four types of Operation Protocol Data Units (OPDUs) are defined; an Invoke, a ReturnResult, a ReturnError, and a Reject. The X.410 formal definition of an OPDU is as follows:



```
OPDU  : : = CHOICE     { -1- Invoke,
                         -2- Returnresult,
                         -3- ReturnError,
                         -4- Reject }
```

6.1. Invoke OPDU

The Invoke OPDU requests that an operation must be performed. It is sent whenever one application desires assistance from another. An application need not wait for one operation to complete before invoking another. At any point in time an application may have any number of operations in progress at a particular remote application.

The X.410 formal definition of the Invoke OPDU is as follows:

Invoke : : = SEQUENCE { invokeid INTEGER, OPERATION, argument ANY }

The “invokeid” element specifies the invoke identifier assigned the operation (invokation). It distinguishes the present operation from others the invoking application may have in progress at the invoked application.

The “OPERATION” element specifies the operation to be performed distinguishing it from other operations of which the invoked application is capable. This value must agree with that specified in the application’s defining recommendation.

The “argument” element specifies the operation’s argument. Its type must be that specified in the application’s defining recommendation.

6.2. ReturnResult OPDU

The ReturnResult OPDU reports the successful completion of an operation. It is sent in eventual response to an Invoke OPDU if the latter is well-formed, the operation is one that reports success only or both success and failure, and the operation succeeds.

The X.410 formal definition of the ReturnResult OPDU is as follows:

ReturnResult : : = SEQUENCE { invokeid INTEGER, result ANY }

the “invokeid” element specifies the operation (invokation) whose success is being reported.

The “result” element specifies the operation’s result. Its type must be that specified in the application’s defining recommendation.

6.3. ReturnError OPDU

The ReturnError OPDU reports the unsuccessful completion of an operation. It is sent in eventual response to an Invoke OPDU if the latter is well-formed, the operation is one that reports failure only or both success and failure, and the operation fails.

The X.40 formal definition of the ReturnError OPDU is as follows:

ReturnError : : = SEQUENCE { invokeid INTEGER, ERROR, parameter ANY }

The “invokeid” element identifies the operation (invokation) whose failure is being reported.

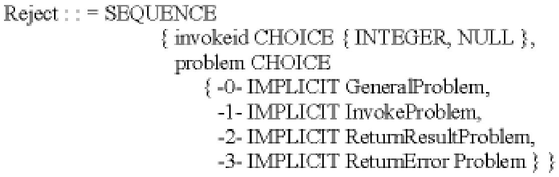
The “ERROR” element specifies the error being reported, distinguishing it from other errors the invoked application reports. Its value must agree with that specified i the application’s defining recommendation, and must be one of those specified as reportable by the operation that was invoked.

The “parameter” element specifies the error’s parameter. Its type must be that specified in the application’s defining recommendation.

6.4. Reject OPDU

The Reject OPDU reports the receipt and rejection of a malformed OPDU. It is sent in eventual response to a malformed OPDU whose type is other than a Reject.

The X.410 formal definition of the Reject OPDU is as follows:

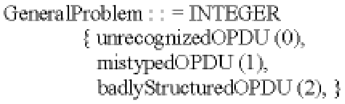


```
Reject : : = SEQUENCE
                  { invokeid CHOICE { INTEGER, NULL },
                    problem CHOICE
                       { -0- IMPLICIT GeneralProblem,
                         -1- IMPLICIT InvokeProblem,
                         -2- IMPLICIT ReturnResultProblem,
                         -3- IMPLICIT ReturnError Problem } }
```

The “invokeid” element specifies the invoke identifier carried by the rejected OPDU. If none was present, the element is of type Null and has a value of NULL.

The “problem” element specifies why the OPDU was rejected. The problems are categorized by OPDU type.

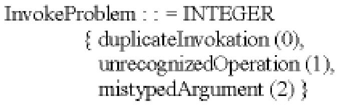
The following general problems are reported:



```
GeneralProblem : : = INTEGER
          { unrecognizedOPDU (0),
            mistypedOPDU (1),
            badlyStructuredOPDU (2), }
```

The problem “unrecognizedOPDU” signifies that the type of the OPDU, as evidenced by its Identifier, is not one of the four defined in the X.410 recommendation. The problem “mistypedOPDU” signifies that the structure of the OPDU does not conform to the X.410 recommendation. The problem “badlyStructureOPDU” signifies that the structure of the OPDU does not conform to the X.409 recommendation.

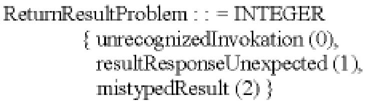
The following Invoke OPDU-specific problems are reported:



```
InvokeProblem : : = INTEGER
            { duplicateInvokation (0),
              unrecognizedOperation (1),
              mistypedArgument (2) }
```

The problem “duplicateInvokation” signifies that the invoke identifier violates the assignment rule of Section 6.1. The problem “unrecognizedOperation” signifies that the operation is not one of those specified in the application’s defining recommendation. The problem “mistypedArgument” signifies that the type of the operation argument supplied is not that specified in the application’s defining recommendation.

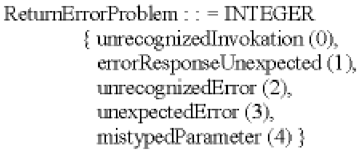
The following ReturnResultProblem OPDU-specific problems are reported:



```
ReturnResultProblem : : = INTEGER
            { unrecognizedInvokation (0),
              resultResponseUnexpected (1),
              mistypedResult (2) }
```

The problem “unrecognizedInvokation” signifies that no operation with the specified invoke identifier is in progress. The problem “resultResponseUnexpected” signifies that the invoked operation does not report success. The problem “mistypedResult” signifies that the type of the operation result supplied is not that specified in the application’s defining recommendation.

The following ReturnError OPDU-specific problems are reported:



```
ReturnErrorProblem : : = INTEGER
            { unrecognizedInvokation (0),
              errorResponseUnexpected (1),
              unrecognizedError (2),
              unexpectedError (3),
              mistypedParameter (4) }
```

The problem “unrecogniedInvokation” signifies that no operation with the specified invoke identifier is in progress. The problem “errorResponseUnexpeced” signifies that the invoked operation does not report failure. The problem “unrecognizedError” signifies that the reported error is not of those specified in the application’s defining recommendation. The problem “unexpectedError” signifies that the reported error is not one that the invoked operation may report. The problem “mistypedParameter” signifies that the type of the error parameter supplied is not that specified in the application’s defining recommendation.

7. ACD MIS Data Stream Protocol Definition

The services of the protocol are defined by informally summarizing the features supported by each of the remote operation used, when this remote operations can be used, and the parameters each requires.

This document defines version of zero of the ACD MIS data stream protocol. Further revisions of this document will define new versions of the protocol.

The ACD MIS ROs are structured in the following format:

- Session Control ROs - sent from the DSP to the switch. These ROs are prefixed with “nos”.
- ROs for functionality the switch provides to the DSP - sent from the DSP to the switch. These ROs are prefixed with “dsp”.
- ROs for functionality the DSP provides to the switch - sent from the switch to the DSP. These ROs are prefixed with “switch”.

NOTE: 1. No more than 10 ROs generated by the DSP and requiring a replay can be outstanding and not responded to at any one point in time. If 10 ROs requiring a reply are outstanding, and the switch receives another DSP generated RO that requires a replay, the switch will take down the X.25 session.

NOTE: 2. A Reject OPDU can be sent in response to any of the ROs outlined in this protocol.

7.1. Session Control ROs

noslogon

- The DSP must execute a nosLogon O on each X.25 SVC set up to the switch - the DSP is responsible for setting up the X.25 SVC.
- The logon operation is used for security control and must be executed before any other ROs can be executed by the switch.
- After the switch executes a nosLogon for ACD MIS, the SVC will only be used for transfer of ACD MIS ROs until a nosLogout is received.
- A nosLogon RO must be executed on a X.25 SVC before any other RO can be successfully executed by the switch.

- Arguments required to Invoke the nosLogon operation:
  - ProtocolVersion – the NOP protocol version in the DSP.
  - Userid – DSP userid.
  - Password – the combination of the Userid and Password is used by the switch to enforce access security - i.e., to prevent an unauthorized customer from gaining access to the switch.
  - Profile – this parameter identifies the “activity” to follow - i.e., ROs related to file transfer, ROs related to alarms and reports, ROs related to the ACD Management Information System Data Stream, etc.
- ReturnResult expected from the nosLogon operation:
  - ProtocolVersion – the NOP protocol version the switch is running. The DSP will always use the NOP protocol version that is in the switch.
  - NetworkEquipmentSoftwareVersion – exact structure yet to be defined.
- ReturnErrors for a nosLogon operation (choice of):
  - invalidArgument – e.g., unrecognized Userid or incorrect Password.
  - operationSequenceProblem – e.g., a valid nosLogon operation was previously executed on this “connection” and has not been terminated with a nosLogout operation.
  - systemProblem – switch system problem.
  - ApplicationResourceShortage – e.g., the logon operation cannot be completed because some system limit on the maximum number of nosLogon operations has been exceeded.

NOTE: It is assumed that a nosLogon operation in the switch is effectively terminated (i.e., partially completed ROs are assumed to be implicitly aborted) after an RTS exception.

nosLogout

- The logout operation is invoked by the DSP to terminate a previous logon operation.

- After a nosLogout RO is executed, no further ROS can be successfully executed on the X.25 SVC until after a nosLogon RO is executed by the switch.

- The nosLogout Ro will execute successfully anytime after the nosLogon RO has successfully executed on the switch. Successful execution of the nosLogout RO will terminate the ACD MIS Data Stream application on the SVC - the DSP is responsible for terminating the SVC after the nosLogout RO has successfully executed in the switch.

- The nosLogout operation does not require an argument.

- ReturnResult expected from a nosLogut Operation – Explicit parameters are not required since the ReturnResult OPDU conveys sufficient information (i.e., the invokeid) to indicate completion of the nosLogout Operation

- ReturnErrors for a nosLogout operation (choice of):

  – operationSequenceProblem – e.g., a nosLogon operation was not previously executed on this connection.

NOTE: 1. Exception to condition handling – The nosLogout operation is assumed to implicitly about any partially completed ROs.

NOTE: 2. Once initiated, a nosLogout is irreversible – i.e., it cannot be refused.

7.2. ROs For Functionality Switch Provides to DSP

dspAssociatePool

- The dspAssociatePool RO is invoked by the DSP to indicate to the switch a pool of ACD groups that the DSP wants associated with the ACD MIS Data Stream application on the SVC.

- The pool of ACD groups specified must have been previously set up in the switch.

- The pool is identified by a “pool name” which is an IA5STRONG string of one to 16 characters.

- The DSP must know this pool name, and the password that has been associated with the pool, before the dspAssociatePool RO can be executed successfully by the switch.

- After the switch has successfully executed a nosLogon RO, the dspAssociatePool RO must be executed before any other ROs (besides nosLogout and dspQuery-DateAndTod) from the DSP to the switch can be successfully executed by the switch on the ACD groups within the pool.

- A pool of ACD groups can only be associated with one SVC at any one point in time.

- Arguments required to Invoke the dspAssociatePool operation:

  – PoolName - a one to 16 character IA5STRING string that identifies a collection (pool) of ACD groups within the switch that the DSP wants associated with this SVC. Valid characters that the switch will accept for a pool name are capital A to Z, and digits 0 to 9.

  – Password – the combination of the PoolName and Password is used by the switch to enforce security for ACD customers - i.e., only the DSP with the correct Password for the PoolName will be able to associate the pool with the DSP’s SVC. Valid characters that the switch will accept for a password are capital A to Z, and digits 0 to 9.

  – Throttle - this field is reserved for future use in throttling the switch originated ROs. For version 0 of this protocol, the throttle value should always be zero.

- ReturnResult expected from the dspAssociatePool operation – Explicit parameters are not required since the ReturnResult OPDU conveys sufficient information (i.e., the invokeid) to indicate completion of the dspAssociationPool operation.

- ReturnErrors for a dspAssociatePool operation (choice of):

  – invalidArgument – e.g., unrecognized PoolName or invalid Password.

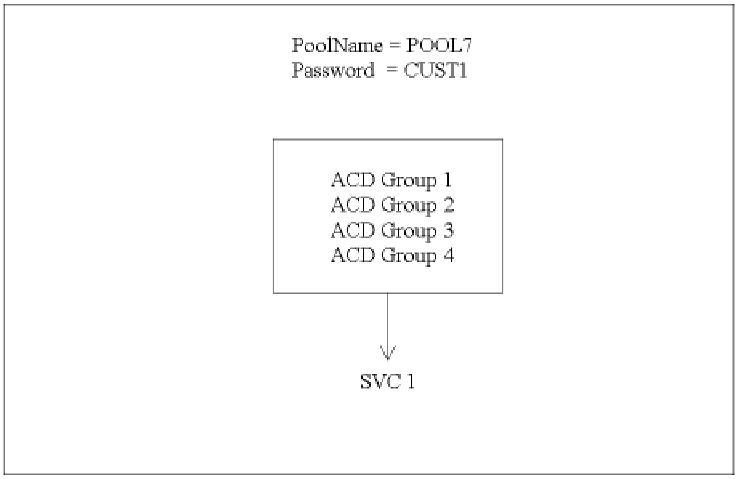
  – operationSequenceProblem – e.g., a nosLogon operation was not previously executed on this SVC.

  – systemProblem

- applicationResourceProblem

In the following example, a DSP connected up to the switch on “SVC 1” would have to send a dspAssociatedPool RO to the switch with “POOL7” specified as the PoolName, and “CUST1” specified as the password in order to associate ACD groups ACD GROUP 1, ACD GROUP 2, ACD GROUP 3, and ACD GROUP4 with the SVC.

Figure 3. Example of Pool of ACD Groups



dspRequestInit

- The dspRequestInit Ro is invoked by the DSP to request that the ACD configuration of the pool associated with the SVC be transmitted to the DSP so that the DSP can initialize its database. The configuration transmitted to the DSP is a snapshot of the ACD configuration in the switch at the time of the initialization. If ACD groups are added to and/or deleted from the switch, or agent positions are added to and/or deleted from the switch, and so on, then the DSP has to be reinitialized before it has knowledge of the new ACD configuration in the switch.

- The ACD configuration of the pool associated with the SVC is transmitted to the DSP via the switchSendSubPoolData, switchSendAcdGroupData, switchSendSupplDnData, and switchSendAgentPosData ROs. The switchSendSubPoolData ROs (one for each subpool within the pool) will be transmitted first. Next, for each ACD group, a switchSendAcdGroupData RO followed by switchSendSupplDnData ROs (one for each supplementary ACD-DN associated with the ACD group), will be transmitted. Finally, switchSendAgentPosData ROs will be transmitted for each agent position belonging to an ACD group within the associated pool.

- Once the transmission of an ACD configuration has started, it will continue until either the entire configuration of the ACD groups associated with the SVC has been transmitted to the DSP, or the switch receives a dspStopInit RO or nosLogout RO.

- The dspRequestInit RO will successfully execute in the switch only after the switch has successfully executed a dspAssociatePool Ro and before the nosLogout RO successfully executes on the switch.

- The dspRequestInit RO will NOT successfully execute while transmission of an ACD configuration is in progress.

- The dspRequestInit operation does not require an argument.

- ReturnResult expected from the dspRequestInit operation – Explicit parameters are not required since the ReturnResult OPDU conveys sufficient information (i.e., the invokeid) to indicate completion of the dspRequestInit operation.

- ReturnErrors for a dspRequestInit operation (choice of):

  – operationSequenceProblem – e.g., a dspAssociatePool operation was not previously executed on this SVC.

  – systemProblem

  – applicationResourceProblem

dspStopInit

- The dspStopInit RO is invoked by the DSP to request that the transmission of the configuration of the pool of ACD groups associated with the SVC be stopped immediately before the transmission is complete.

- Once a ACD configuration transmission has been stopped by a dspStopInit RO, it will remain stopped until the switch receives another dspRequestInit RO. The transmission of an ACD configuration will NOT start from where it was stopped by the dspStopInit RO; it will start over again, and transmit the entire configuration of the pool of ACD groups associated with the SVC.

- The dspStopInit RO will successfully execute on the switch only while transmission of an ACD configuration is in progress.

- The dspStopInit operation does not require an argument.

- ReturnResult expected from the dspStopInit operation – Explicit parameters are not required Since the ReturnResult OPDU conveys sufficient information (i.e., the invokeid) to indicate completion of the dspStopInit operation.

- ReturnErrors for a dspStopInit operation (choice of):

  - operationSequenceProblem – e.g., a dspRequestInit operation was not previously executed on this SVC.

  - systemProblem

  - applicationResourceProblem

dspQueryDateAndTod

- The dspQueryDateAndTod RO is invoked by the DSP to request the current date and time of day from the switch.

- The dspQueryDateAndTod RO will successfully execute in the switch any time after the switch has successfully executed the nosLogon RO and anytime before the nosLogout RO successfully executes in the switch.

- The dspueryDateAndTod operation does not require an argument.

- ReturnResult expected from the dspQueryDateAndTod operation:

  - DateAndTime – the current date and time registered in the switch.

- ReturnErrors for a dspQueryDateAndTod operation (choice of):

  - operationSequenceProblem – e.g., a nosLogon operation was not previously executed on this SVC.

  - systemProblem

  - applicationResourceProblem

dspStartTransfer

- The dspStartTransfer RO in invoked by the DSP to inform the switch that the DSP is ready to accept transmission of switchSendEvent ROs.

- Once the transmission of the switchSendEvent ROs has started, it will continue until either a dspStopTransfer RO, or a nosLogout RO is received from the DSP and the RO successfully executes in the switch.

- The dspStartTransfer RO will successfully execute in the switch only after the switch has successfully executed the dspAssociatePool RO and only before the nosLogout RO successfully executes in the switch.

- The dspStartTransfer RO will NOT successfully execute in the switch while switchSendEvent ROs are already being transmitted to the DSP, i.e., a dspStartTransfer RO has already successfully executed in the switch.

- The dspStartTransfer operation does not require an argument.

- ReturnResult expected from the dspStartTransfer operation – Explicit parameters are not required since the ReturnResult OPDU conveys sufficient information (i.e., the invokeid) to indicate completion of the dspStartTransfer operation.

- ReturnErrors for a dspStartTransfer operation (choice of):

    - operationSequenceProblem – e.g., a dspAssociatePool operation was not previously executed on this SVC.

    - systemProblem

    - applicationResourceProblem

dspStopTransfer

- The dspStopTransfer RO is invoked by the DSP to request that the transmission of switchSendEvent ROs be stopped immediately.

- Once the transmission of switchSendEvent ROs has been stopped by a dspStopTransfer RO, it will remain stopped until the switch receives another dsp-StartTansfer RO.

- The dspStopTransfer RO will only execute successfully in the switch while switchSendEvent ROs are being transmitted to the DSP - i.e., after the dspStartTransfer RO has successfully execute in the switch.

- The dspStopTransfer operation does not require an argument.

- ReturnResult expected from the dspStopTransfer operation – Explicit parameters are not required since the ReturnResult OPDU conveys sufficient information (i.e., the invokeid) to indicate completion of the dspStopTransfer operation.

- ReturnErrors for a dspStopTransfer operation (choice of):

  - operationSequenceProblem –e.g., a dspStartTransfer operation was not previously executed on this SVC.

  - systemProblem

  - applicationResourceProblem

7.3. ROs For Functionality DSP Provides To Switch

switchSendSubPoolData

- The switchSendSubPoolData RO is invoked by the switch to inform the DSP of a subpool name to subpool number association.

- One switchSendPoolData RO is transmitted to the DSP at initialization time for each subpool defined within the pool that has been associated with the SVC.

- The switchSendSubPoolData ROs are transmitted before any other initialization ROs in the initialization sequence.

- The switchSendSubPoolData RO will execute successfully in the DSP only after the dspRequestInit RO has successfully executed in the switch, and only before either the dspStopInit or nosLogout ROs have successfully executed in the switch, or the switchEndOfInit Ro has successfully executed in the DSP.

- Arguments required to Invoke the switchSendSubPoolData operation:

  - AcdSubPoolData – the information about a subpool that is defined within the pool of ACD groups associated with the SVC.

- A ReturnResult is not expected from the DSP for this operation.

- A ReturnError is not expected from the DSP for this operation.

switchSendAcdGroupData

- The switchSendAcdGroupData RO is invoked by the switch to transmit information about an ACD group that is part of the pool of ACD groups associated with the SVC.

- One switchSendAcdGroupData RO is transmitted to the DSP at initialization time for each ACD group in the pool associated with the SVC.

- The switchSendAcdGroupData ROs are transmitted after the switchSendSubPool-Data ROs, and before the switchSendAgentPosData ROs in the initialition sequence.

- The switchSendAcdGroupData RO will execute successfully in the DSP only after the dspRequestInit RO has successfully executed in the switch, and only before either the dspStopInit or nosLogout ROs have successfully executed in the switch, or the switchEndOfInit RO has successfully executed in the DSP.

- Arguments required to Invoke the switchSendAcdGroupData operation:

  – AcdGroupData – the information about an ACD group that belongs to the ACD configuration that has been associated with the SVC.

- A ReturnResult is not expected from the DSP for this operation.

- A ReturnError is not expected from the DSP for this operation.

switchSendSupplDnData

- The switchSendSupplAcDn RO invoked by the switch to transmit information about a supplementary ACD-DN that is associated with an ACD group that is part of the pool associated with the SVC.

- The switchSendSupplAcdDn Ros for supplementary ACD-DNs associated with a particular ACD group are transmitted directly after the switchSendAcdGroupData RO for the ACD group - before the switchSendAcdGroupData RO for the next ACD group is transmitted.

- The switchSendSupplAcdDn RO will execute successfully in the DSP only after the dspRequestInit RO has successfully executed in the switch, and only before either the dspStopInit or nosLogout Ros have successfully executed in the switch, or the switchEndOfInit RO has successfully executed in the DSP.

- Arguments required to Invoke the switchSendSupplDnData operation:

  - AcdSupplDnData – the information about an ACD supplementary DN that belongs to an ACD group within the ACD configuration that has been associated with the SVC.

- A ReturnResult is not expected from the DSP for this operation.
- A ReturnError is not expected from the DSP for this operation.

switchSendAgentPosData

- The switchSendAgentPosData RO is invoked by the switch to transmit information about an ACD agent position that belongs to an ACD group that is part of the pool associated with the SVC.
- The switchSendAgentPosData ROs are transmitted to the DSP only after all of the switchSendAcdSupplDn ROs have been transmitted.
- The switchSendAgentPosData ROs will NOT be grouped together by ACD group when they are transmitted to the DSP.
- The switchSendAgentPosData RO will execute successfully in the DSP only after the dspRequestInit Ro has successfully executed in the switch, and only before either the dspStopInit or nosLogout ROs have successfully executed in the switch, or the switchEndOfInit RO has successfully executed in the DSPs.
- Arguments required to Invoke the switchSendAgentPosData operation:
  - AgentPositionData – the information about an ACD agent position within the ACD configuration that has been associated with the SVC.
- A ReturnResult is not expected from the DSP for this operation.
- A ReturnError is not expected from the DSP for this operation.

switchEndOfInit

- The switchEndOfInit RO is invoked by the switch to inform to the DSP that the transmission of the ACD configuration is complete.
- The switchEndOfInit RO will execute successfully in the DSP only after the dspRequestInit RO has successfully executed in the switch, and only after the ACD configuration ROs have been transmitted to the DSP.

- The switchEndOfInit operation does not require an argument.
- A ReturnResult is not expected from the DSP for this operation.
- A ReturnError is not expected from the DSP for this operation.

switchSendEvent

- The switchSendEvent RO is invoked by the switch to inform the DSP of an ACD call event or ACD agent position event that has occurred within the pool of ACD groups associated with the SVC.
- The switchSendEvent RO will execute successfully in the DSP only after the dspStartTransfer RO has successfully executed in the switch, and only before either the dspStopTransfer RO, or nosLogout RO successfully executes in the switch.
- Events occurring on agent positions within the groups belonging to the pool associated with the SVC will only be transmitted to the DSP when agents are logged into the agent positions.
- Arguments required to Invoke the switchSendEvent operation:
  - AcdEvent – the information about an ACD call or agent position event that has occurred in the switch.
- A ReturnResult is not expected from the DSP for this operation.
- A ReturnError is not expected from the DSP for this operation.

8. State Transition Diagrams

Seven states exist in the ACD Management Information System Data Stream protocol:

1. Idle

No SVC has been established.

2. Logged Off

The SVC has been connected. The switch is waiting for a nosLogon RO.

3. Logged On

The logon has been verified. The ACD MIS application is running.

4. ACD MIS Wait

The ACD MIS application has received and verified a dspAssociatePool RO. The association of a ACD group pool has been made to the SVC.

5. Initializing

The ACD MIS application has received a dspRequestInit RO and is in the process of transmitting the ACD configuration to the DSP.

6. Transferring

The ACD MIS application has received a dspStartTransfer RO and is transmitting ACD event and ACD load management event ROs to the DSP.

7. Initializing and Transferring

Either the ACD MIS application has received a dspRequestInit RO after transmission of ACD event ROs has already been started to the DSP, or the ACD MIS application has received a dspStartTransfer RO after transmission of the ACD configuration ROs has already been started to the DSP. In either case, ACD configuration data is transmitted to the DSP interspersed with ACD event ROs.

The following state tables represent the state transitions that can occur in the ACD Management Information System protocol.

Remote Operation	State 1 Idle	State 2 Logged Off	State 3 Logged On
nosLogon	n/a	New State is State 3	Operation Sequence Problem
nosLogout	n/a	Operation Sequence Problem	New State is State 2
dspAssociatePool	n/a	Operation Sequence Problem	New State is State 4
dspRequestInit	n/a	Operation Sequence Problem	Operation Sequence Problem
dspStopInit	n/a	Operation Sequence Problem	Operation Sequence Problem

dspQueryDateAndTod	n/a	Operation Sequence Problem	New State is State 3
dspStartTransfer	n/a	Operation Sequence Problem	Operation Sequence Problem
dspStopTransfer	n/a	Operation Sequence Problem	Operation Sequence Problem
switchSendSubPoolData	n/a	Operation Sequence Problem	Operation Sequence Problem
switchSEndACDGroupData	n/a	Operation Sequence Problem	Operation Sequence Problem
switchSendSupplAcdDn	n/a	Operation Sequence Problem	Operation Sequence Problem
switchSendAgentPosData	n/a	Operation Sequence Problem	Operation Sequence Problem
switchEndOfInit	n/a	Operation Sequence Problem	Operation Sequence Problem
switchSendEvent	n/a	Operation Sequence Problem	Operation Sequence Problem

Remote Operation	State 4 ACD MIS Wait	State 5 Initializing	State 6 Transferring	State 7 Init and Transfer
nosLogon	Operation Sequence Problem	Operation Sequence Problem	Operation Sequence Problem	Operation Sequence Problem
nosLogout	New State is State 2	New State is State 2	New State is State 2	New State is State 2
dspAssociatePool	Operation Sequence Problem	Operation Sequence Problem	Operation Sequence Problem	Operation Sequence Problem
dspRequestInit	New State is State 5	Operation Sequence Problem	New State is State 7	Operation Sequence Problem
dspStopInit	Operation Sequence Problem	New State is State 4	Operation Sequence Problem	New State is State 6

dspQuery-DateAndTod	New State is State 4	New State is State 5	New State is State 6	New State is State 7
dspStart-Transfer	New State is State 6	New State is State 7	Operation Sequence Problem	Operation Sequence Problem
dspStop-Transfer	Operation Sequence Problem	Operation Sequence Problem	New State is State 4	New State is State 5
switchSend-SubPoolData	Operation Sequence Problem	New State is State 5	Operation Sequence Problem	New State is State 7
switch-SendACD GroupData	Operation Sequence Problem	New State is State 5	Operation Sequence Problem	New State is State 7
switchSend-SupplAcdDn	Operation Sequence Problem	New State is State 5	Operation Sequence Problem	New State is State 7
switchSendAgentPosData	Operation Sequence Problem	New State is State 5	Operation Sequence Problem	New State is State 7
switchEnd-OfInit	Operation Sequence Problem	New State is State 4	Operation Sequence Problem	New State is State 6
switchSendEvent	Operation Sequence Problem	Operation Sequence Problem	New State is State 6	New State is State 7

9. Time Sequence Diagrams

The following diagrams are intended to illustrate the flow of control and RO OPDU’s between the DSP and the switch. Examples of individual operations, sequences of operations, and exception and error situations are included. Only the type of OPDU (i.e., Invoke, ReturnResult, ReturnError, and Reject), the invokeid (an integer), and the RO name (for Invoke OPDUs only) are given in the diagrams.

NOTE: The invokeIDs used in these examples were arbitrarily chosen within the rules for invokeId assignment (i.e., an invokeId is an integer and cannot be reused while there is an outstanding ReturnResult or ReturnError expected).

Figure 4. nosLogon Operation

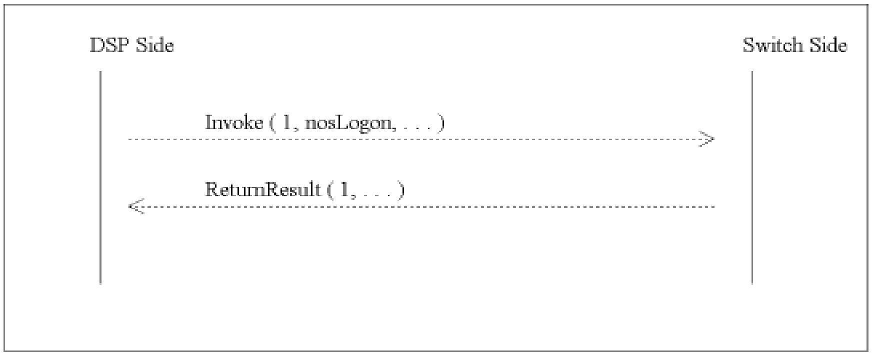


Figure 5. dspQueryDateAndTod Operation

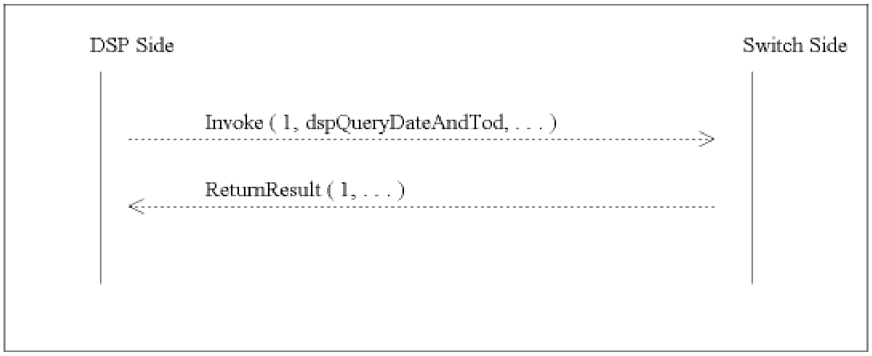

DSP Side
Switch Side
Invoke (1, dspQueryDateAndTod, . . .)
ReturnResult (1, . . .)

Figure 6. Logon, Initialization, and Transfer

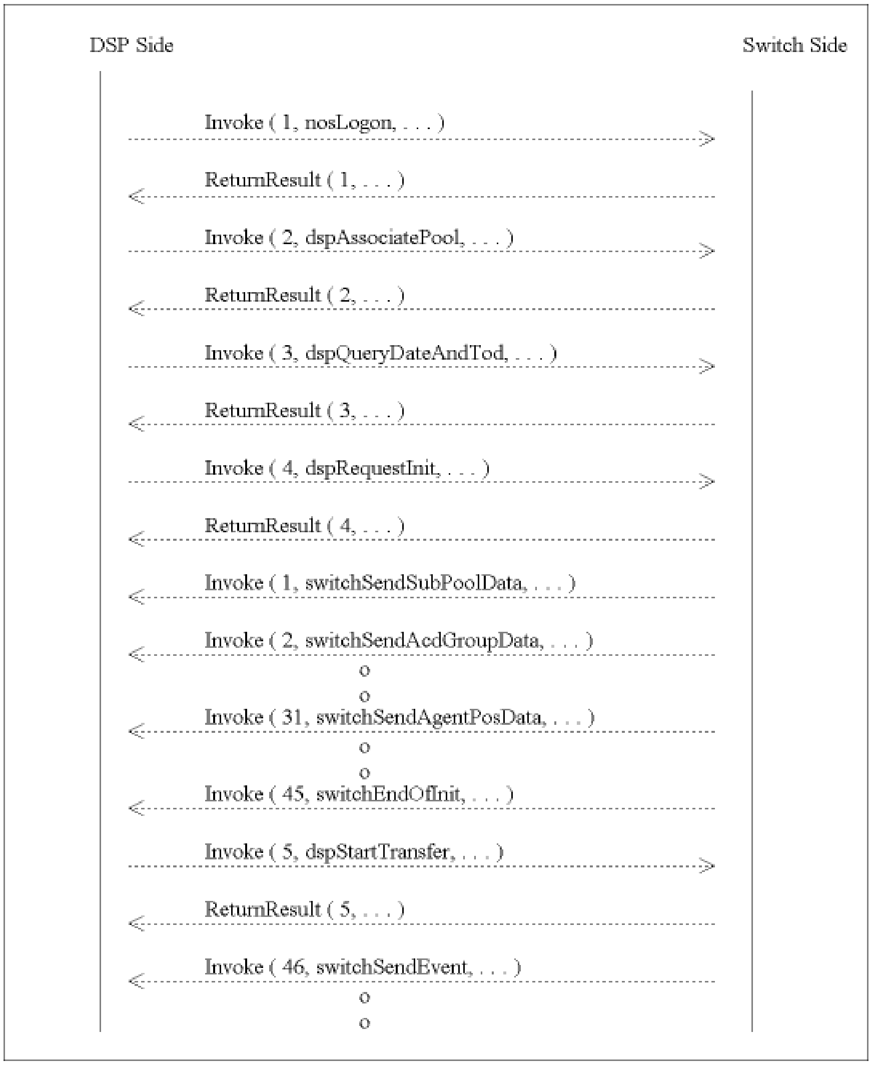

DSP Side
Switch Side
Invoke (1, nosLogon, . . .)
ReturnResult (1, . . .)
Invoke (2, dspAssociatePool, . . .)
ReturnResult (2, . . .)
Invoke (3, dspQueryDateAndTod, . . .)
ReturnResult (3, . . .)
Invoke (4, dspRequestInit, . . .)
ReturnResult (4, . . .)
Invoke (1, switchSendSubPoolData, . . .)
Invoke (2, switchSendAcdGroupData, . . .)
o
o
Invoke (31, switchSendAgentPosData, . . .)
o
o
Invoke (45, switchEndOfInit, . . .)
Invoke (5, dspStartTransfer, . . .)
ReturnResult (5, . . .)
Invoke (46, switchSendEvent, . . .)
o
o

Figure 7. Logon and Initialization Halted

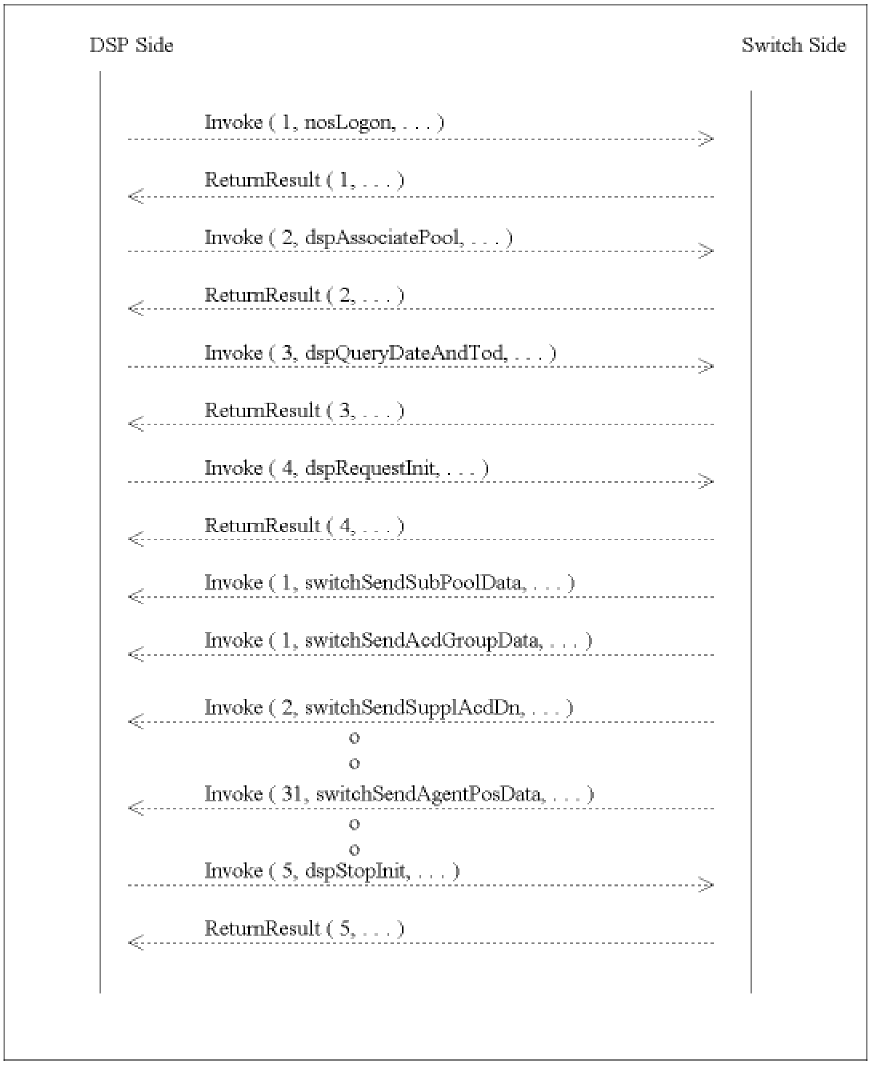

DSP Side
Switch Side
Invoke (1, nosLogon, . . .)
ReturnResult (1, . . .)
Invoke (2, dspAssociatePool, . . .)
ReturnResult (2, . . .)
Invoke (3, dspQueryDateAndTod, . . .)
ReturnResult (3, . . .)
Invoke (4, dspRequestInit, . . .)
ReturnResult (4, . . .)
Invoke (1, switchSendSubPoolData, . . .)
Invoke (1, switchSendAcdGroupData, . . .)
Invoke (2, switchSendSupplAcdDn, . . .)
o
o
Invoke (31, switchSendAgentPosData, . . .)
o
o
Invoke (5, dspStopInit, . . .)
ReturnResult (5, . . .)

Figure 8. Logon and Transfer

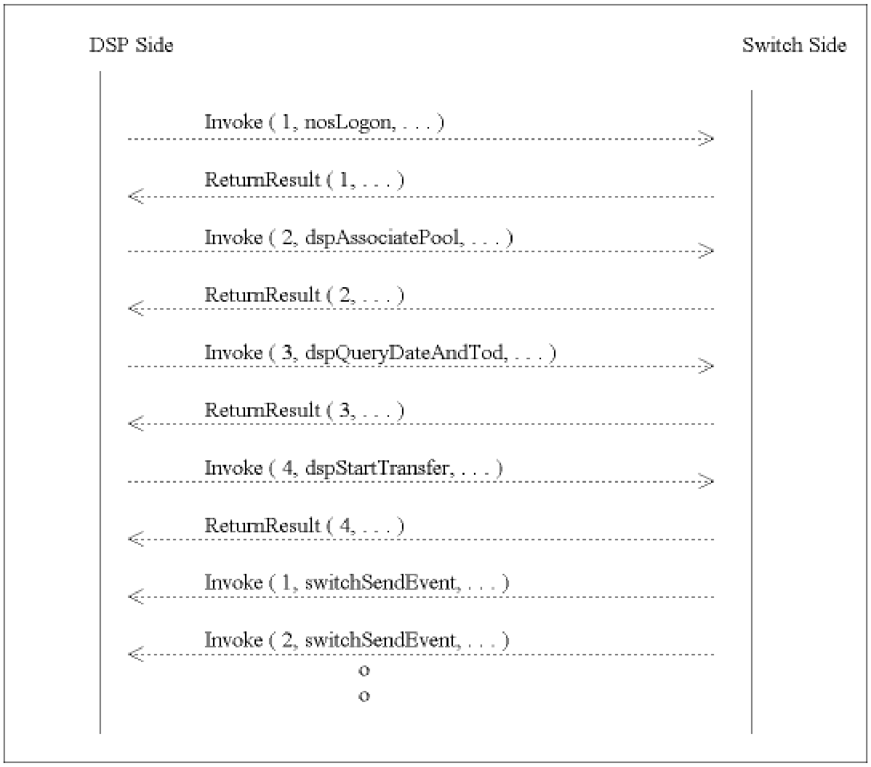

DSP Side
Switch Side
Invoke (1, nosLogon, . . .)
ReturnResult (1, . . .)
Invoke (2, dspAssociatePool, . . .)
ReturnResult (2, . . .)
Invoke (3, dspQueryDateAndTod, . . .)
ReturnResult (3, . . .)
Invoke (4, dspStartTransfer, . . .)
ReturnResult (4, . . .)
Invoke (1, switchSendEvent, . . .)
Invoke (2, switchSendEvent, . . .)
o
o

Figure 9. Exception in nosLogon Operation

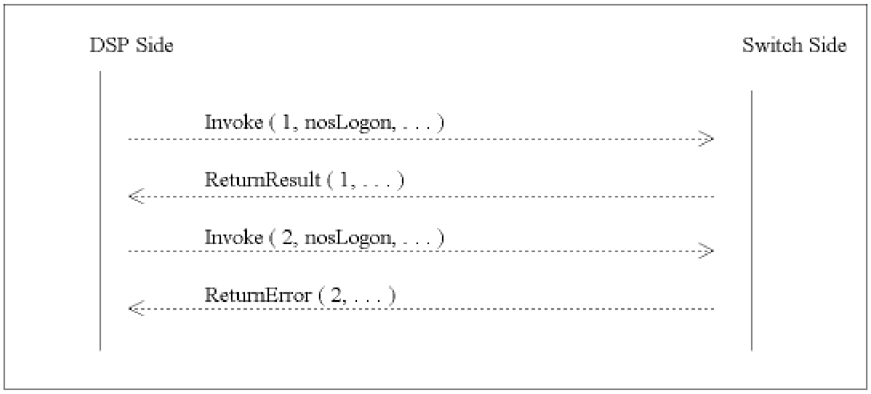

DSP Side
Switch Side
Invoke (1, nosLogon, . . .)
ReturnResult (1, . . .)
Invoke (2, nosLogon, . . .)
ReturnError (2, . . .)

Figure 10. InvokeId Assignment Exception

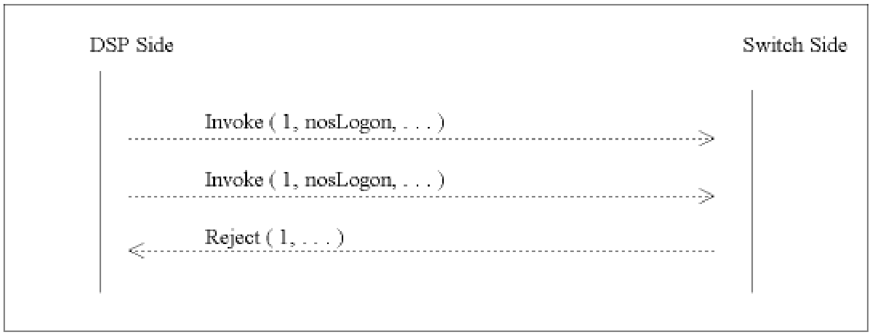
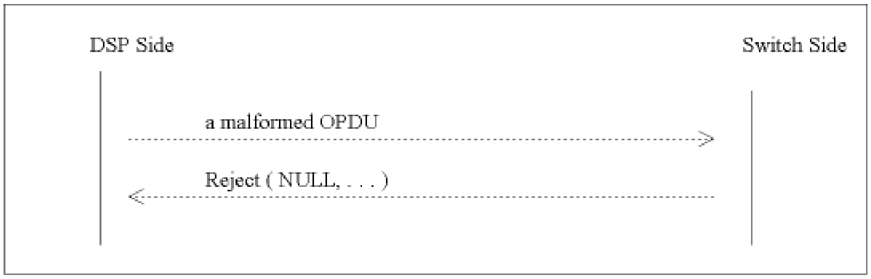


Figure 11. Switch response to a malformed OPDU



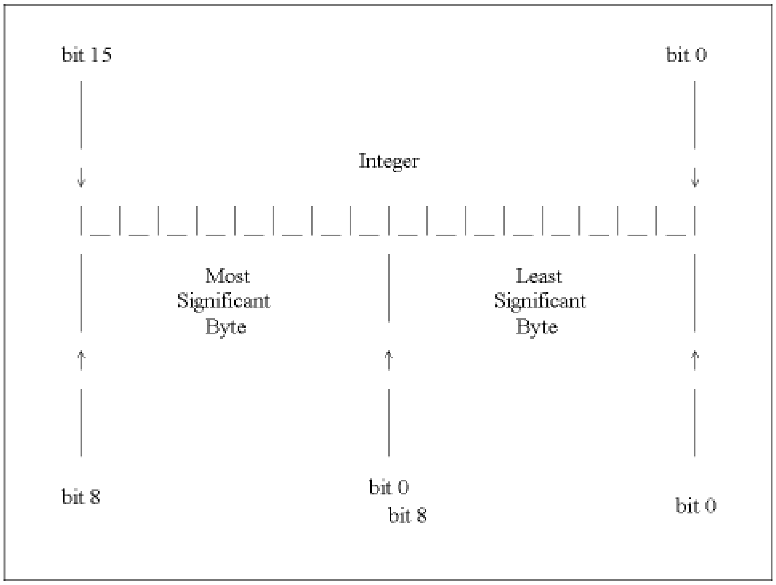
10. Detailed Argument Descriptions

The following sections describe the meaning and usage of the ACDSubPoolData, AcdGroupDAta, AcdSuppleDnData, AgentPositionData, and AcdEvent arguments in the switch originated ROs.

Within each byte, the bit designated “bit 0” is transmitted first, followed by bits 2, 3, 4, etc. Similarly, the byte shown at the top of each figure is sent first, followed by the second byte from the top, and so on. This also applies to the definitions in Section 11.2 as well as this section.

The following diagram demonstrates the bit assignments for individual bytes and two byte integers:

Figure 12. Bit Assignments



The labeling conventions used in the two byte integers are:

- MS_Byte = Most Significant Byte
- LS_Byte = Least Significant Byte

10.1. ACDEvent Detailed Description

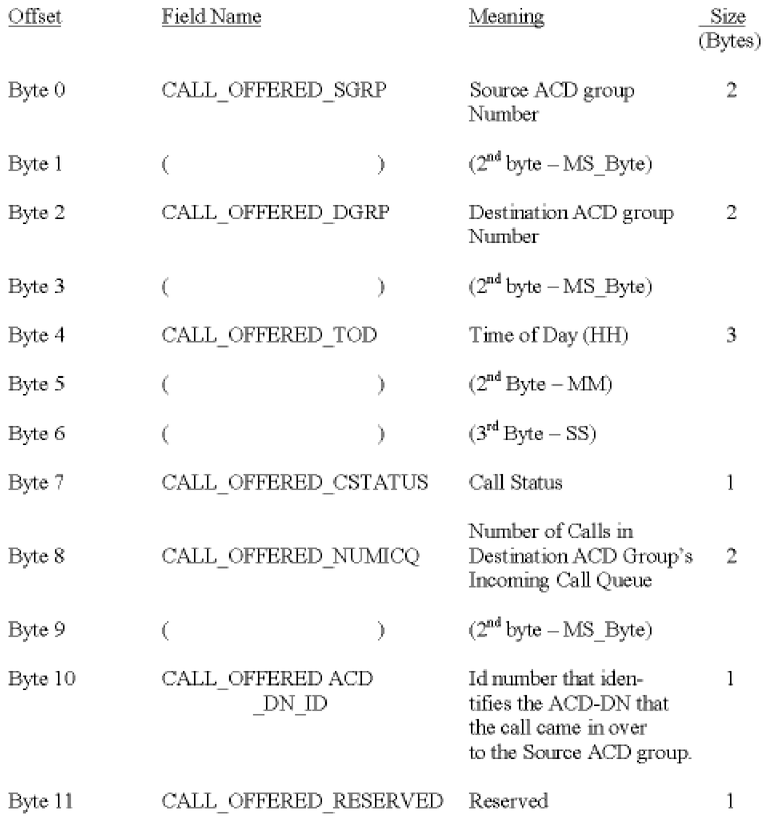
This section describes in detail the format of the ACD call event data.

10.1.1. acdCallOffered

Description: Provides information on each call offered to the destination ACD group.

When Sent: Send once for every call at the time it is offered to the destination ACD group.

Fields:



Offset	Field Name	Meaning	Size (Bytes)
Byte 0	CALL_OFFERED_SGRP	Source ACD group Number	2
Byte 1	( )	(2nd byte – MS_Byte)	
Byte 2	CALL_OFFERED_DGRP	Destination ACD group Number	2
Byte 3	( )	(2nd byte – MS_Byte)	
Byte 4	CALL_OFFERED_TOD	Time of Day (HH)	3
Byte 5	( )	(2nd Byte – MM)	
Byte 6	( )	(3rd Byte – SS)	
Byte 7	CALL_OFFERED_CSTATUS	Call Status	1
Byte 8	CALL_OFFERED_NUMICQ	Number of Calls in Destination ACD Group's Incoming Call Queue	2
Byte 9	( )	(2nd byte – MS_Byte)	
Byte 10	CALL_OFFERED ACD _DN_ID	Id number that identifies the ACD-DN that the call came in over to the Source ACD group.	1
Byte 11	CALL_OFFERED_RESERVED	Reserved	1

Fields Description:

1. Call Offered SGRP:

This is the internal number in the range zero to 255 inclusive that is assigned to each ACD group datafilled in the switch. The number is unique for each ACD group. The Source ACD group number identifies the initial ACD group within the switch that the ACD call was presented to. The Source ACD group number remains the same in all ACD call group number remains the same in all ACD call offered messages sent to the DSP for an ACD call. The source ACD group number could be either (1) different from the Destination ACD group number if the call was deflected to the Destination ACD group, or (2) the same group as the Destination ACD group if the call was not deflected to the Destination ACD group.

2. Call Offered DGRP:

This is the internal number of the ACD group to which this ACD call is offered. The group number is in the range 0 to 255.

3. Call Offered TOD:

This is used to indicate the time in hours, minutes and seconds (0 - 24 hrs) that this call was offered to the destination ACD group.

4. Call Offered CStatus:

This field has one of the following values to indicate the status of the call offered to the destination ACD group.

The call status information and description is as follows:

0 = indicates that the ACD call terminated on an agent position in the Destination ACD group without being queued.

1 = indicates that the ACD call is being queued in the Destination ACD group's Incoming Call Queue.

2 = indicates that the ACD call is being deflected to the Destination ACD group's overflow route because the Destination ACD group's Incoming Call Queue has overflowed.

3 = indicates that the ACD call cannot be overflowed again and is being given treatment.

4 = indicates that the ACD call is being deflected because the Destination ACD group is in night service.

5. Call Offered NUMICQ:

This is the number of calls queued in the Destination ACD group's incoming call queue at the time this call was offered to the Destination ACD group. The number in queue is in the range zero to 511 inclusive.

If the call is queued (i.e., call status = 1) then the number in this field includes this call.

6. Call Offered ACD DN ID:

This is a number between zero and 16 that identifies the ACD-DN in the Source ACD group that the ACD call came in over. The Source ACD group is the initial ACD group within the switch that the ACD call was presented to.

7. Call Offered Reserved:

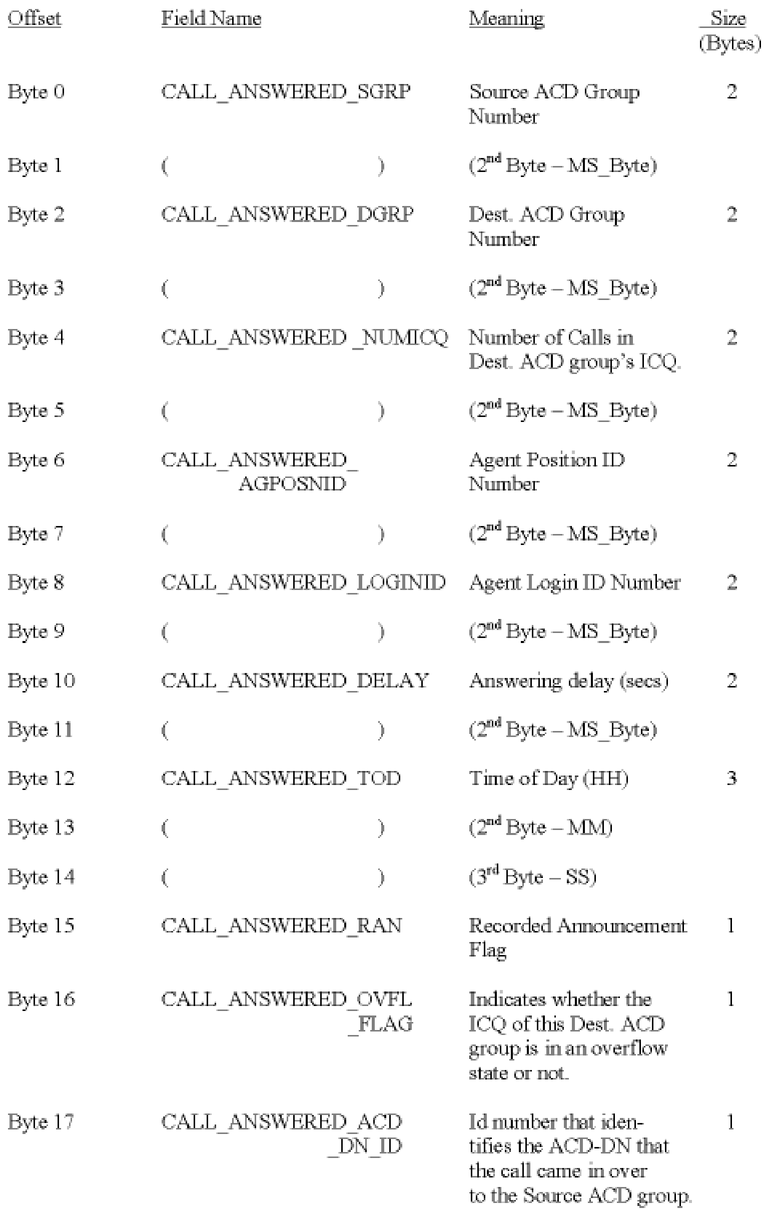
This field is reserved, and will be set to zero.

10.1.2. acdCallAnswered

Description: Provides information on ACD calls answered by agent positions.

When Sent: Sent once to the down stream processor when an ACD call is answered at an agent position.

Fields:



Offset	Field Name	Meaning	Size (Bytes)
Byte 0	CALL_ANSWERED_SGRP	Source ACD Group Number	2
Byte 1	( )	(2nd Byte – MS_Byte)	
Byte 2	CALL_ANSWERED_DGRP	Dest. ACD Group Number	2
Byte 3	( )	(2nd Byte – MS_Byte)	
Byte 4	CALL_ANSWERED _NUMICQ	Number of Calls in Dest. ACD group's ICQ.	2
Byte 5	( )	(2nd Byte – MS_Byte)	
Byte 6	CALL_ANSWERED_ AGPOSNID	Agent Position ID Number	2
Byte 7	( )	(2nd Byte – MS_Byte)	
Byte 8	CALL_ANSWERED_LOGINID	Agent Login ID Number	2
Byte 9	( )	(2nd Byte – MS_Byte)	
Byte 10	CALL_ANSWERED_DELAY	Answering delay (secs)	2
Byte 11	( )	(2nd Byte – MS_Byte)	
Byte 12	CALL_ANSWERED_TOD	Time of Day (HH)	3
Byte 13	( )	(2nd Byte – MM)	
Byte 14	( )	(3rd Byte – SS)	
Byte 15	CALL_ANSWERED_RAN	Recorded Announcement Flag	1
Byte 16	CALL_ANSWERED_OVFL _FLAG	Indicates whether the ICQ of this Dest. ACD group is in an overflow state or not.	1
Byte 17	CALL_ANSWERED_ACD _DN_ID	Id number that identifies the ACD-DN that the call came in over to the Source ACD group.	1

Fields Description:

1. Call Answered SGRP:

This is the internal number in the range zero to 255 inclusive that is assigned to each ACD group datafilled in the switch. The number is unique for each ACD group. The Source

AD group number identifies the initial ACD group within the switch that the ACD call was presented to. The Source ACD group number could either be (1) different from the Destination ACD group number if the call was deflected to the Destination ACD group, or (2) the same group as the Destination ACD group if the call was not deflected to the Destination ACD group.

2. Call Offered DGRP:

This is the internal number of the ACD group that has answered the call. The group number is in the range zero to 255 inclusive.

3. Call Answered NUMICQ:

This is the number of calls in the Destination ACD group's incoming call queue at the time this ACD call was answered. The number of calls queued is in the range zero to 511 inclusive.

4. Call Answered AGPOSNID:

This is a 4 digit position identification number (ID) in the range 0001-9999 that is assigned to the agent position when it is datafilled. The agent position numbers are unique for all ACD agent positions in a switch. This is the position identification of the agent position this ACD call was answered at. The agent position ID number is zero if the agent position that the call was answered at was not assigned a position ID at datafill time. It is up to the DSP as to how it wants to handle this situation.

5. Call Answered LOGINID:

This field is a 4 digit identification number in the range 0001-9999 that agents use to LOGIN to agent positions. The login ID sent in this message is the ID of the agent currently logged into the agent position identified by the agent position ID. A login ID can only be logged onto one agent position at any point in time.

6. Call Answered Delay:

This is the answer delay time in seconds. The answer delay is the difference between the time that the call was offered to the Source ACD group and the time that it was answered in the destination ACD group. The call answered delay includes both the time that the call was queued (if it was queued) and the time it was ringing the agent position before it was answered.

7. Call Answered TOD:

This is the time in hours, minutes, and seconds (0 - 24 hrs) when this ACD call was answered.

8. Call Answered RAN:

This field can have a value of zero to one. A zero indicates that recorded announcement was not given to this caller, whereas a one indicates that recorded announcement was given.

9. Call Answered OVFL FLAG:

This field can have a value of zero or one. A zero indicates that the incoming call queue if the Destination ACD group is not an overflow state. A one indicates that the incoming call queue of the Destination ACD group is in an overflow state.

10. Call Answered ACD DN ID:

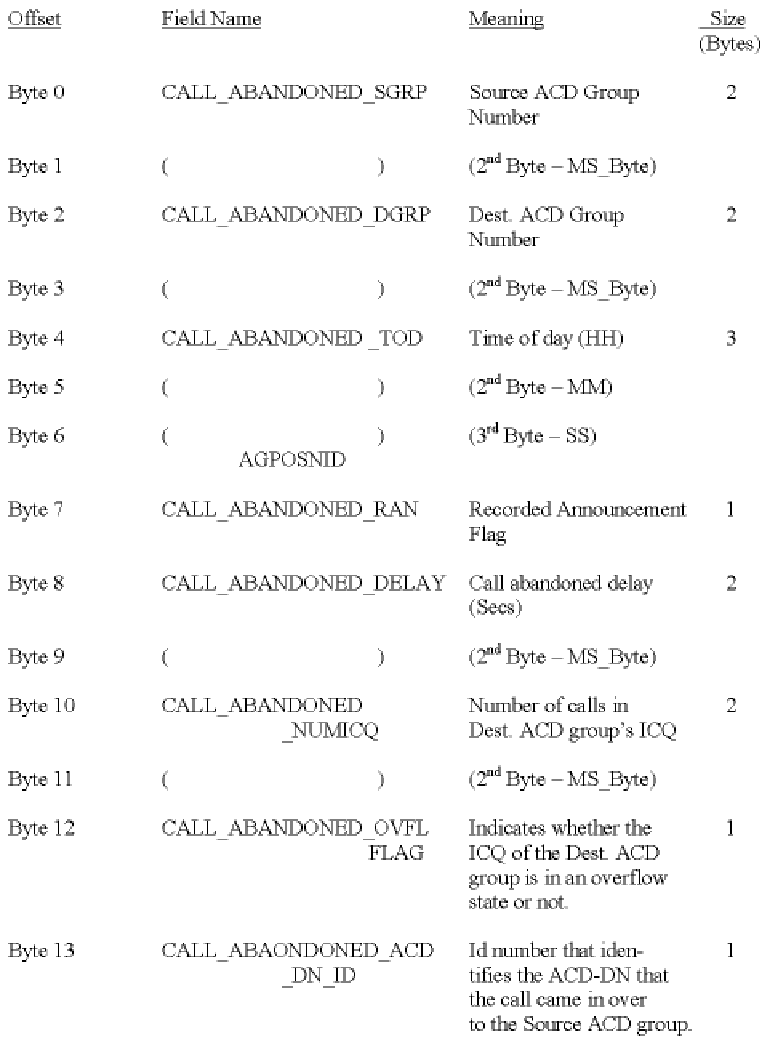
This is a number between zero and 16 that identifies the ACD-DN in the Source ACD group that the ACD call came in over. The Source ACD group is the initial ACD group within the switch that the ACD call was presented to.

10.1.3. acdCallAbandoned

Description: When the caller hangs up before the ACD call is answered, the ACD call is either (1) released from the ACD group's Incoming Call Queue, or (2) the agent position stops ringing. In both cases a Call Abandoned message is sent to the DSP. An ACD call can also be abandoned as a result of some other action (eg. maintenance action from a MAP position) being taken on the parties involved. In this case also, a Call Abandoned message is sent to the DSP.

When Sent: Sent once for each abandoned call.

Fields:



Offset	Field Name	Meaning	Size (Bytes)
Byte 0	CALL_ABANDONED_SGRP	Source ACD Group Number	2
Byte 1	( )	(2nd Byte – MS_Byte)	
Byte 2	CALL_ABANDONED_DGRP	Dest. ACD Group Number	2
Byte 3	( )	(2nd Byte – MS_Byte)	
Byte 4	CALL_ABANDONED _TOD	Time of day (HH)	3
Byte 5	( )	(2nd Byte – MM)	
Byte 6	( ) AGPOSNID	(3rd Byte – SS)	
Byte 7	CALL_ABANDONED_RAN	Recorded Announcement Flag	1
Byte 8	CALL_ABANDONED_DELAY	Call abandoned delay (Secs)	2
Byte 9	( )	(2nd Byte – MS_Byte)	
Byte 10	CALL_ABANDONED _NUMICQ	Number of calls in Dest. ACD group's ICQ	2
Byte 11	( )	(2nd Byte – MS_Byte)	
Byte 12	CALL_ABANDONED_OVFL FLAG	Indicates whether the ICQ of the Dest. ACD group is in an overflow state or not.	1
Byte 13	CALL_ABAONDONED_ACD _DN_ID	Id number that identifies the ACD-DN that the call came in over to the Source ACD group.	1

Fields Description:

1. Call Abandoned SGRP:

This is the internal number in the range zero to 255 inclusive that is assigned to each ACD group datafilled in the switch. The number is unique for each ACD group. The Source ACD group number identifies the initial ACD group within the switch that the ACD call was presented to. The Source ACD group number could either be (1) different from the Destination ACD group number if the call was deflected to the Destination ACD group, or (2) the same group as the Destination ACD group if the call was not deflected to the Destination ACD group.

2. Call Abandoned DGRP:

This is the internal number of the ACD group that accepted this ACD call. The group number is in the range zero to 255 inclusive.

3. Call Abandoned TOD:

This is the time in hours, minutes, and seconds (0 - 24 hrs) when this call was abandoned by the caller.

4. Call Abandoned RAN:

This field can have a value of zero or one. A zero indicates that recorded announcement was not given to this caller, whereas a one indicates that recorded announcement was given.

5. Call Abandoned Delay:

This is used to indicate the time in seconds that the caller waited before abandoning the call. The abandon delay is the time between the time that this call was offered to the Source ACD group and the time the caller abandoned.

The abandon delay includes one or both of the following times:

A. the time that the call spent in the incoming call queue (if the call was queued)

B. the time that the call was ringing an agent position (if it terminated directly on an agent position or if it was removed from the incoming call queue to terminate on an agent position).

6. Call Abandoned NUMICQ:

This is the number of calls queued in the Destination ACD group’s incoming call queue at the time the caller abandoned.

If the abandoned call had been queued then this number is the number of calls queued after the abandoned call has been removed from the queue. The number of calls queued is in the range zero to 511 inclusive.

7. Call Abandoned OVFL FLAG:

This field can have a value of zero or one. A zero indicates that the incoming call queue if the Destination ACD group is not in an overflow state. A one indicates that the incoming call queue of the Destination ACD group is in an overflow state.

8. Cal Abandoned ACD DN ID:

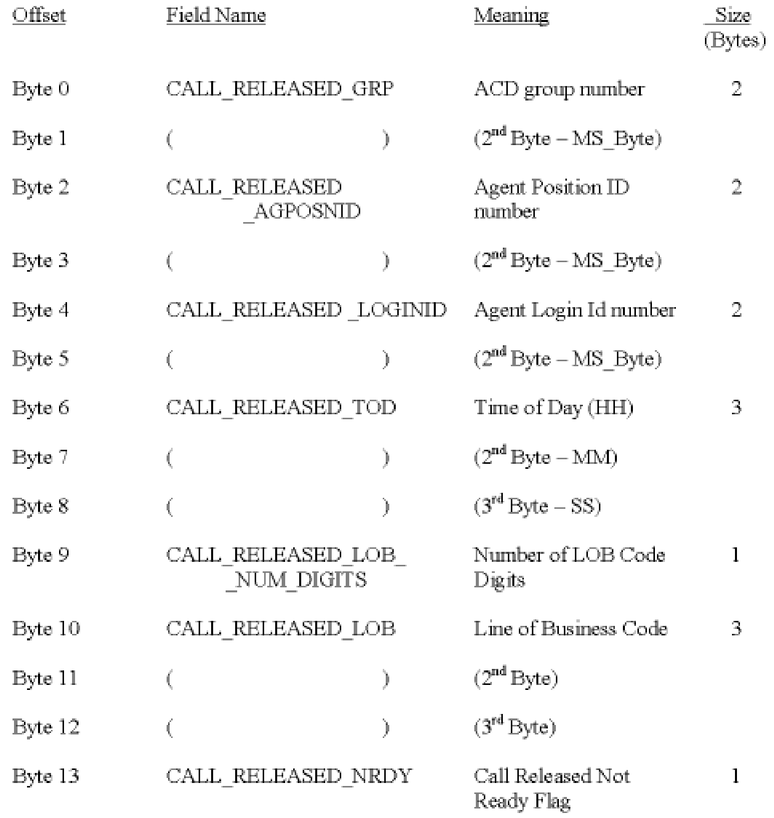
This is a number between zero and 16 that identifies the ACD-DN in the Source ACD group that the ACD call came in over. The Source ACD group is the initial ACD group within the switch that the ACD call was presented to.

10.1.4. acdCallreleased

Description: Provides information on answered ACD calls released either by the caller or the ACD agent.

When Sent: Sent once when the ACD call is released from the agent position because (1) the caller hung up and the agent position was release, or (2) because the agent released the call by pressing the Release (RLS) key, or (3) because the agent released the call by pressing the

Not Ready key. An ACD call can also be released as a result of some other action (e.g., maintenance action from a MAP position) being taken on the parties involved.



Offset	Field Name	Meaning	Size (Bytes)
Byte 0	CALL_RELEASED_GRP	ACD group number	2
Byte 1	( )	(2nd Byte – MS_Byte)	
Byte 2	CALL_RELEASED _AGPOSNID	Agent Position ID number	2
Byte 3	( )	(2nd Byte – MS_Byte)	
Byte 4	CALL_RELEASED _LOGINID	Agent Login Id number	2
Byte 5	( )	(2nd Byte – MS_Byte)	
Byte 6	CALL_RELEASED_TOD	Time of Day (HH)	3
Byte 7	( )	(2nd Byte – MM)	
Byte 8	( )	(3rd Byte – SS)	
Byte 9	CALL_RELEASED_LOB_ _NUM_DIGITS	Number of LOB Code Digits	1
Byte 10	CALL_RELEASED_LOB	Line of Business Code	3
Byte 11	( )	(2nd Byte)	
Byte 12	( )	(3rd Byte)	
Byte 13	CALL_RELEASED_NRDY	Call Released Not Ready Flag	1

Field Description:

1. Call Released GRP:

This is the internal number of the ACD group that has answered this call. The group number is in the range zero to 255 inclusive.

2. Call Released AGPOSNID:

This is a unique 4 digit position identification number (ID) in the range 0001-9999 that is assigned to the agent position when it is datafilled. The agent position numbers are unique for all ACD agent positions in a switch. This is the position ID number of the agent position this ACD call was released from. The agent position ID number is zero if the agent position that the call was released from was not assigned a position ID at datafill time. It is up to the DSP as to how it wants to handle this situation.

3. Call Released LOGINID:

This field is a 4 digit identification number in the range 0001-9999 that agents use to LOGIN to agent positions. The login ID sent in this message is the ID of the agent currently logged into the agent position identified by the agent position ID. A login ID can only be logged into one agent position at any point in time.

4. Call Released TOD:

This is the time in hours, minutes and seconds (0 - 24 hrs) when this ACD call was released.

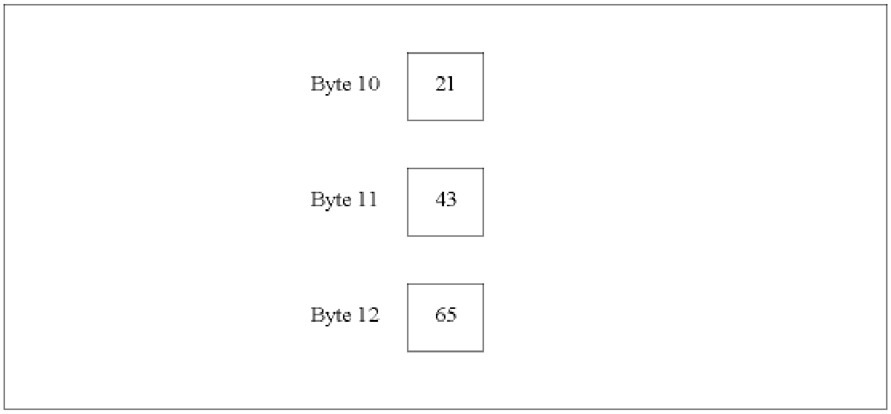
5. Call Release LOB NUM Digits:

This is the number of digits entered by the agent as the Line of Business (LOB) code associated with this ACD call. If the agent did not enter a LOB code for this ACD call then the value of this field is zero. Otherwise the value of this field is in the range 1-6.

6. Call Released LOB:

This is the Line of Business (LOB) code that the agent has associated with this ACD call. The number of digits in the code is given by the CALL_RELEASED_LOB_NUM_DIGITS field. If the number of digits entered is zero (i.e., the agent did not enter a LOB code for this call), then the CALL_RELEASED_LOB field is meaningless. Each digit occupies 4 bits. For example, if the LOB code ’123456’ was associated with an ACD call, the CALL_RELEASED_LOB field would look like:

Figure 13. LOB Code



Byte 10	21
Byte 11	43
Byte 12	65

7. Call Released NRDY:

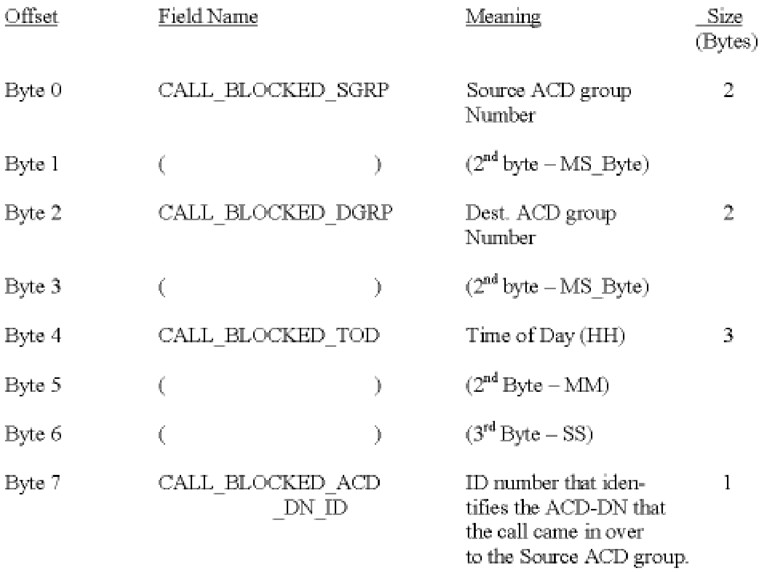
This field indicates whether or not the Not Ready key was used to release the ACD call. A zero means that the Not Ready key was not used to release the ACD call (i.e., the call was released by some other means). A one means that the Not Ready key was used to release the ACD call.

10.1.5. acdCallBlocked

Description: Provides information on ACD calls that are blocked because there are insufficient resources to service the call after it has been accepted by the Destination ACD group.

When Sent: Sent once when an ACD call is blocked from being answered by an agent position because there are insufficient resources to handle the call after it has been accepted by the Destination ACD group.

Fields:



Offset	Field Name	Meaning	Size (Bytes)
Byte 0	CALL_BLOCKED_SGRP	Source ACD group Number	2
Byte 1	( )	(2nd byte – MS_Byte)	
Byte 2	CALL_BLOCKED_DGRP	Dest. ACD group Number	2
Byte 3	( )	(2nd byte – MS_Byte)	
Byte 4	CALL_BLOCKED_TOD	Time of Day (HH)	3
Byte 5	( )	(2nd Byte – MM)	
Byte 6	( )	(3rd Byte – SS)	
Byte 7	CALL_BLOCKED_ACD _DN_ID	ID number that identifies the ACD-DN that the call came in over to the Source ACD group.	1

Fields Description:

1. Call Blocked SGRP:

This is the internal number in the range zero to 255 inclusive that is assigned to each ACD group datafilled in the switch. The group number is unique for each ACD group. The Source ACD group number identifies the initial ACD group within the switch that the ACD call was presented to. The Source ACD group number could either be (1) different from the Destination ACD group number if the call was deflected to the Destination

ACD group, or (2) the same group as the Destination ACD group if the call was not deflected to the Destination ACD group.

2. Call Blocked GRP:

This is the internal number of the ACD group that had accepted the call but then could not service it because of insufficient resources.

3. Call Blocked TOD:

This is the time in hours, minutes and seconds (0 - 24) when this ACD call was blocked.

4. Call Blocked ACD DN ID:

This is a number between zero and 16 that identifies the ACD-DN in the Source ACD group that the ACD call came in over. The Source ACD group is the initial ACD group within the switch that the ACD call was presented to.

10.1.6. agentPositionEtent

Description: Provides information on events that occur at an Agent position.

When Sent: Generated and sent when the following events occur:

A. Agent Login:

This event message is sent after the agent has completed the login procedure to login to an agent position.

B. Agent Logout:

This event message is sent after the agent has completed the logout procedure to logout from an agent position.

C. Activate ACD Not Ready

This event message is sent when the Not Ready key is pressed to activate the Not Ready feature.

D. Deactivate ACD Not Ready

This event message is sent when the ACD Not Ready key is pressed to deactivate the Not Ready feature.

E. Answer a DN Call

This event message is sent when an incoming DC call is answered by an agent position.

F. Originate a DN Call

This event message is sent when an agent accesses a secondary DN to make an outgoing call.

NOTE: This message is sent as soon as the agent presses the secondary DN key to originate a DN call, i.e., before the agent dials the digits.

G. Release a DN Call

This event message is sent either when a secondary DN automatically released because the caller hung up or when the agent presses the RLS key to release the DN call. A DN call also be released as a result of some other action (e.g., maintenance action from a MAP position) being taken on the parties involved.

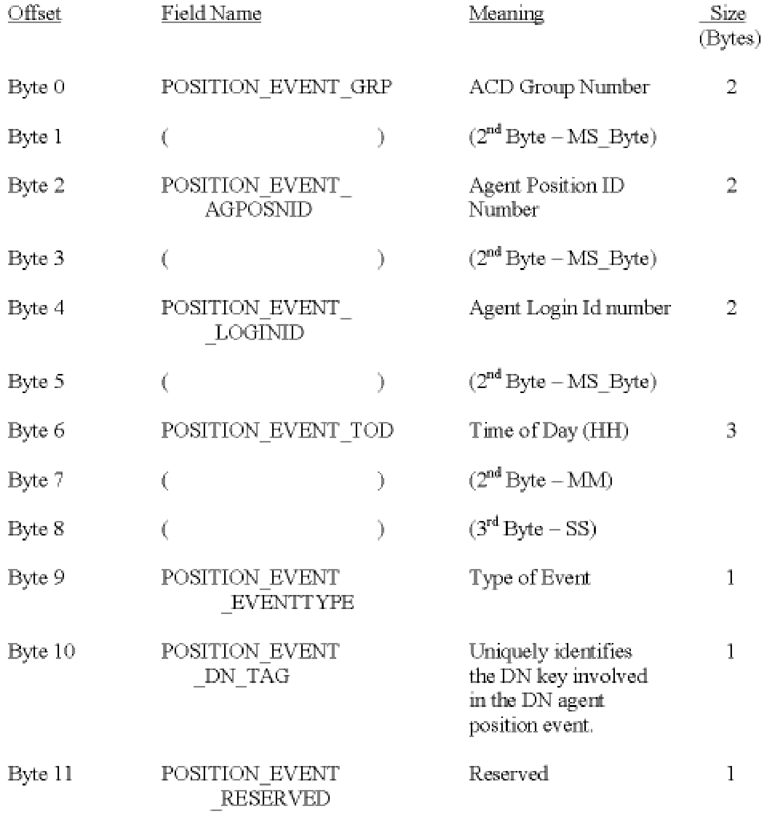
H. Activate MSB

This event is sent when the MSB key is pressed to activate the MSB feature.

I. Deactivate MSB

This event message is sent when the MSB key is pressed to deactivate the MSB feature.

Fields:



Offset	Field Name	Meaning	Size (Bytes)
Byte 0	POSITION_EVENT_GRP	ACD Group Number	2
Byte 1	( )	(2nd Byte – MS_Byte)	
Byte 2	POSITION_EVENT_ AGPOSNID	Agent Position ID Number	2
Byte 3	( )	(2nd Byte – MS_Byte)	
Byte 4	POSITION_EVENT_ _LOGINID	Agent Login Id number	2
Byte 5	( )	(2nd Byte – MS_Byte)	
Byte 6	POSITION_EVENT_TOD	Time of Day (HH)	3
Byte 7	( )	(2nd Byte – MM)	
Byte 8	( )	(3rd Byte – SS)	
Byte 9	POSITION_EVENT _EVENTTYPE	Type of Event	1
Byte 10	POSITION_EVENT _DN_TAG	Uniquely identifies the DN key involved in the DN agent position event.	1
Byte 11	POSITION_EVENT _RESERVED	Reserved	1

Fields Description:

1. Position Event GRP:

This is the internal number of the ACD group to which the agent position belongs. The group number is in the range zero to 255 inclusive.

2. Position Event AGPOSNID:

This is a unique 4 digit position identification number (ID) in the range 0001-9999 that is assigned to the agent position when it is datafilled. The agent position numbers are unique for all ACD agent positions in a switch. This is the position ID number of the agent position that the event occurred on. The agent position ID number is zero if the agent position the event occurred on was not assigned a position ID at datafill time. It is up to the DSP as to how it wants to handle this situation.

3. Position Event LOGINID:

This field is a 4 digit identification number in the range 0001-9999 of the agent that is logged into the agent position. When the login event occurs, this field will have the login ID of the agent logging in. When the logout event occurs, this field will have the login ID of the agent logging out.

4. Position Event TOD:

This is the time in hours, minutes and seconds (0 - 24 hrs) the event occurred at the agent position.

5. Position Event EVENTYPE:

This event field can have the possible values:

- 0 = Agent Login
- 1 = Agent Logout
- 2 = Activate ACD Not Ready
- 3 = Deactivate ACD Not Ready
- 4 = Answer a DN Call
- Originate a DN Call
- Release a DN Call
- Activate MSB
- Deactivate MSB

6. Position Event DN TAG:

This field uniquely identifies the DN key on the agent position specified by POSITION_EVENT_AGPOSNID that a DN related agent position event occurred on. The events that this field applies to are (1) Originate a DN Call, (2) Answer a DN call, and (3) Release a DN call. The POSITION_EVENT_DN_TAG field is meaningless for all other events specified in the POSITION_EVENT_EVENTTYPE field.

7. Position Event Reserved:

This field is reserved, and will be set to zero.

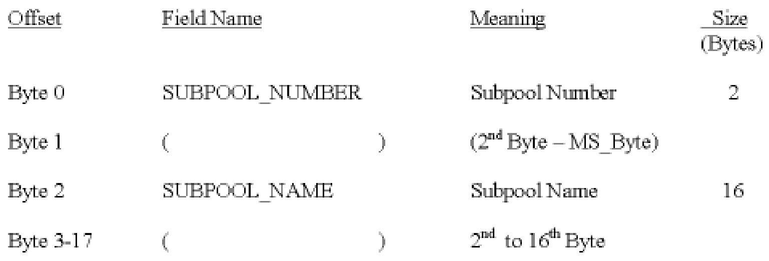
10.2. ACDSUBPOOLDATA

This section describes in detail the format of the AcdSubPoolData argument.

Description: Provides information on the mapping from an internal subpool number to a subpool CLLI.

When Sent: Generated and sent once for each subpool datafilled in the switch for the pool associated with the DSP when the DSP is initialized by the switch.

Fields:



Offset	Field Name	Meaning	Size (Bytes)
Byte 0	SUBPOOL_NUMBER	Subpool Number	2
Byte 1	( )	(2nd Byte – MS_Byte)	
Byte 2	SUBPOOL_NAME	Subpool Name	16
Byte 3-17	( )	2nd to 16th Byte	

Fields Description:

1. Subpool Number:

This is an internal number in the range zero to 127 that the subpool CLLI (subpool name) is associated with.

2. Subpool Name:

This is the Common Language Location Identifier (CLLI) that is used to specify the subpool when datafilling it. The CLLI can be up to 16 ASCII characters; one per byte. In cases where this is less than 16 characters then the end bytes of this field are padded with blanks. Valid characters for the subpool name are capital A to Z, and digits 0 to 9.

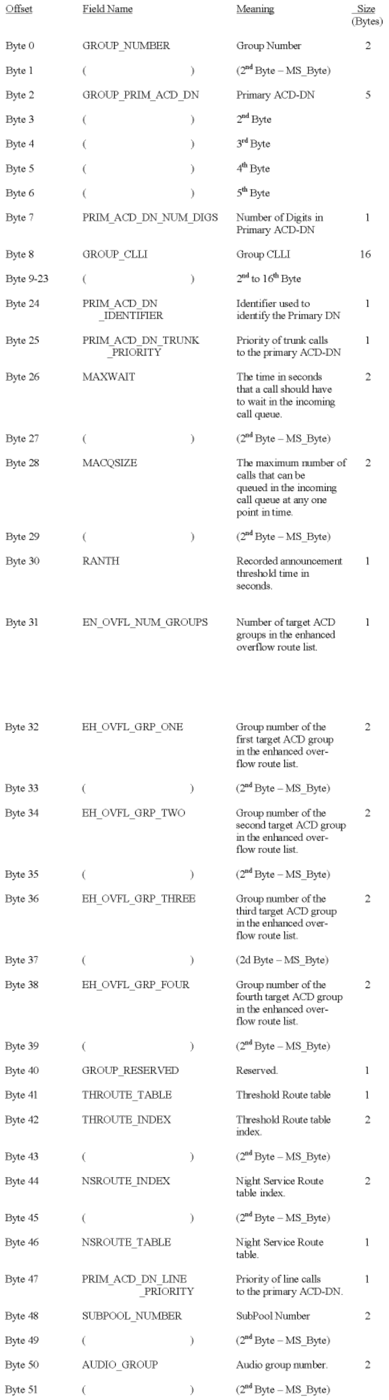
10.3. ACDGROUPDATA

This section described in detail the format of the AcdGroupData argument.

Description: Provides information on the mapping from an internal ACD group number to an ACD group CLLI and Primary ACD-DN. It also provides information on the current attributes of the ACD group.

When Sent: Generated and sent once for each ACD group datafilled in the switch and part of the pool associated with the DSP when the DSP is initialized by the switch.

Fields:



Offset	Field Name	Meaning	Size (Bytes)
Byte 0	GROUP_NUMBER	Group Number	2
Byte 1	( )	(2nd Byte – MS_Byte)	
Byte 2	GROUP_PRIM_ACD_DN	Primary ACD-DN	5
Byte 3	( )	2nd Byte	
Byte 4	( )	3rd Byte	
Byte 5	( )	4th Byte	
Byte 6	( )	5th Byte	
Byte 7	PRIM_ACD_DN_NUM_DIGS	Number of Digits in Primary ACD-DN	1
Byte 8	GROUP_CLLI	Group CLLI	16
Byte 9-23	( )	2nd to 16th Byte	
Byte 24	PRIM_ACD_DN _IDENTIFIER	Identifier used to identify the Primary DN	1
Byte 25	PRIM_ACD_DN_TRUNK _PRIORITY	Priority of trunk calls to the primary ACD-DN	1
Byte 26	MAXWAIT	The time in seconds that a call should have to wait in the incoming call queue.	2
Byte 27	( )	(2nd Byte – MS_Byte)	
Byte 28	MACQSIZE	The maximum number of calls that can be queued in the incoming call queue at any one point in time.	2
Byte 29	( )	(2nd Byte – MS_Byte)	
Byte 30	RANTH	Recorded announcement threshold time in seconds.	1
Byte 31	EN_OVFL_NUM_GROUPS	Number of target ACD groups in the enhanced overflow route list.	1
Byte 32	EH_OVFL_GRP_ONE	Group number of the first target ACD group in the enhanced overflow route list.	2
Byte 33	( )	(2nd Byte – MS_Byte)	
Byte 34	EH_OVFL_GRP_TWO	Group number of the second target ACD group in the enhanced overflow route list.	2
Byte 35	( )	(2nd Byte – MS_Byte)	
Byte 36	EH_OVFL_GRP_THREE	Group number of the third target ACD group in the enhanced overflow route list.	2
Byte 37	( )	(2d Byte – MS_Byte)	
Byte 38	EH_OVFL_GRP_FOUR	Group number of the fourth target ACD group in the enhanced overflow route list.	2
Byte 39	( )	(2nd Byte – MS_Byte)	
Byte 40	GROUP_RESERVED	Reserved.	1
Byte 41	THROUTE_TABLE	Threshold Route table	1
Byte 42	THROUTE_INDEX	Threshold Route table index.	2
Byte 43	( )	(2nd Byte – MS_Byte)	
Byte 44	NSROUTE_INDEX	Night Service Route table index.	2
Byte 45	( )	(2nd Byte – MS_Byte)	
Byte 46	NSROUTE_TABLE	Night Service Route table.	1
Byte 47	PRIM_ACD_DN_LINE _PRIORITY	Priority of line calls to the primary ACD-DN.	1
Byte 48	SUBPOOL_NUMBER	SubPool Number	2
Byte 49	( )	(2nd Byte – MS_Byte)	
Byte 50	AUDIO_GROUP	Audio group number.	2
Byte 51	( )	(2nd Byte – MS_Byte)	

Fields Description:

1. Group Number:

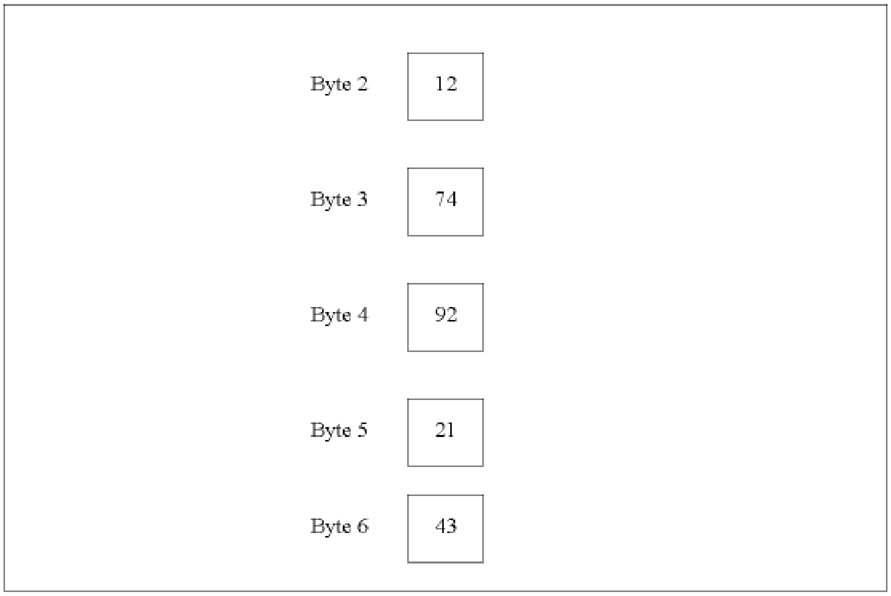
This is the internal ACD group number of the ACD group that the CLLI and primary ACD-DN are associated with. The group number is in the range zero to 255 inclusive.

2. Group PRIM ACD DN:

This is the primary ACD-DN of the ACD group. The primary ACD-DN can be up to 10 digits long; each digit occupying 4 bits. The primary ACD-DN is send in BCD format. If the value of the PRIM_ACD_DN_NUM_DIGS field is zero, then this field has no meaning.

Example: If the Primary ACD-DN is (214) 729-1234, the information in the 5 byte field is as follows:

Figure 14. Primary ACD-DN Example



Byte 2	12
Byte 3	74
Byte 4	92
Byte 5	21
Byte 6	43

3. PRIM ACD DN NUM DIGS:

This number specifies the number of digits in the ACD group's primary ACD-DN. The Primary ACD-DN can be up to 10 digits. If there is no primary ACD-DN associated with the ACD group then the value of this field is zero.

4. Group CCLI:

This is the Common Language Location Identifier (CLLI) that is used to specify the ACD group when datafilling it. The CLLI can be up to 16 ASCII characters; one per byte. In cases where this is less than 16 characters then the end bytes of this field are padded with blanks. Valid characters for the ACD group name are capital A to Z, and digits 0 to 9.

5. PRIM ACD DN Identifier:

This is an identifier between zero and 16 that is used to identify the Primary DN.

6. PRIM ACD DN Trunk Priority:

This is the priority that calls to the primary ACD-DN over trunks are assigned when they are queued in the incoming call queue. It can be in the range zero to three inclusive.

7. MAXWAIT:

This is the maximum time in seconds that a call should have to wait in the incoming call queue before being presented to an agent position to be answered. It can be in the range zero to 1800 inclusive.

8. MAXCQSIZE:

This is the maximum number of calls that can be queued in the incoming call queue at any one point in time. It can be in the range zero to 511 inclusive.

9. RANTH:

This is the recorded announcement threshold time in seconds. It can be zero, or in the range six to 60 inclusive. This field has no meaning if the value of the AUDIO_GROUP field is zero.

10. EH OVFL NUM GROUPS:

This field specifies the number of ACD groups in the Enhanced Overflow route list for the ACD group specified in the ACD_GROUP field. There can be up to four ACD groups specified in an Enhanced Overflow route list. If there are no ACD groups specified in an Enhanced Overflow route list for ACD_GROUP, the value of this field is zero.

NOTE: The switchSendAcdGroupData ROs for the ACD groups in the Enhanced Overflow route list of ACD_GROUP may not have been transmitted yet to the DSP. Hence, the DSP may not yet know about the Enhanced Overflow ACD groups. Also, there is no restriction that the ACD groups in the Enhanced

Overflow route list of ACD_GROUP be associated with the same pool of ACD groups as ACD_GROUP. Hence, the DSP may never get a switchSendAcdGroupData RO for an ACD group in the Enhanced Overflow route list. This is a switch administrative issue that must be dealt with by switch personnel.

11. EH OVFL GRP ONE:

This is the group number of the first ACD group in the Enhanced Overflow route list of ACD_GROUP. The group number is in the range zero to 255 inclusive.

12. EN OVFL GRP TWO:

This is the group number of the second ACD group in the Enhanced Overflow route list of ACD_GROUP. The group number is in the ranger zero to 255 inclusive.

13. EH OVFL GRP THREE:

This is the group number of the third ACD group in the Enhanced Overflow route list of ACD_GROUP. The group number is in the range zero to 255 inclusive.

14. EH OVFL GRP FOUR:

This is the group number of the fourth ACD group in the Enhanced Overflow route list of ACD_GROUP. The group number is in the range zero to 255 inclusive.

15. Group Reserved:

This field is reserved.

16. THROUTE Table:

This field specifies the routing table that is used for the Threshold route of ACD_GROUP. A value of zero means that table OFRT is used. A value of one means that table IBNRTE is used.

17. THROUTE Index:

This specifies where the route list is located in the routing table used for the Threshold route of ACD_GROUP. The index can be in the range zero to 1023 inclusive.

18. NSROUTE Index:

This specifies where the route list is located in the routing table used for the Night Service route of ACD_GROUP. The index can be in the range zero to 1023 inclusive.

19. NSROUTE Table:

This field specifies the routing table that is used for the Night Service route of ACD_GROUP. A value of zero means that table OFRT is used. A value of one means that table IBNRTE is used.

20. PRIM ACD DN Line Priority:

This is the priority that calls to the primary ACD-DN over lines are assigned when they are queued in the incoming call queue. It can be in the range zero to three inclusive.

21. SubPool Number:

This specifies the subpool that the ACD group belongs to. The subpool number is in the range zero to 127. The subpool name to subpool number mapping is transmitted to the DSP in the switchSendSubPoolData RO.

22. Audio Group:

This specifies the audio group that is used in giving announcement and/or music to callers queued in the incoming call queue. It can be in the range one to 15 inclusive. If no audio specified to give callers announcement and/or music in the incoming call queue, then this field will be set to zero.

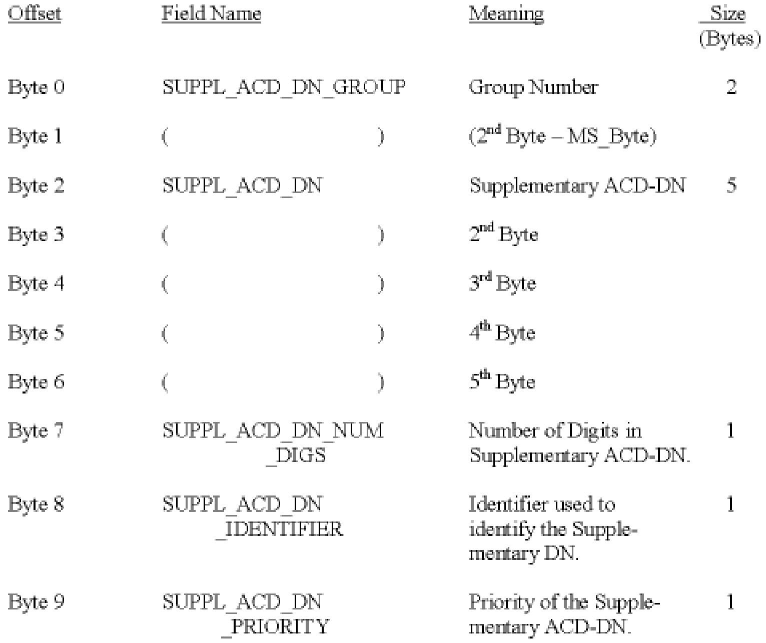
10.4. ACDSUPPLDNDATA

This section describes in detail the format of the AcdSupplDn Data argument.

Description: Provides information on the mapping from an internal ACD group number to a Supplementary ACD-DN.

When Sent: Generated and sent once for each Supplementary ACD-DN associated with an ACD group at the time when the DSP is initialized by the switch.

Fields:



Offset	Field Name	Meaning	Size (Bytes)
Byte 0	SUPPL_ACD_DN_GROUP	Group Number	2
Byte 1	( )	(2nd Byte – MS_Byte)	
Byte 2	SUPPL_ACD_DN	Supplementary ACD-DN	5
Byte 3	( )	2nd Byte	
Byte 4	( )	3rd Byte	
Byte 5	( )	4th Byte	
Byte 6	( )	5th Byte	
Byte 7	SUPPL_ACD_DN_NUM _DIGS	Number of Digits in Supplementary ACD-DN.	1
Byte 8	SUPPL_ACD_DN _IDENTIFIER	Identifier used to identify the Supplementary DN.	1
Byte 9	SUPPL_ACD_DN _PRIORITY	Priority of the Supplementary ACD-DN.	1

Fields Description:

1. SUPPL ACD DN Group:

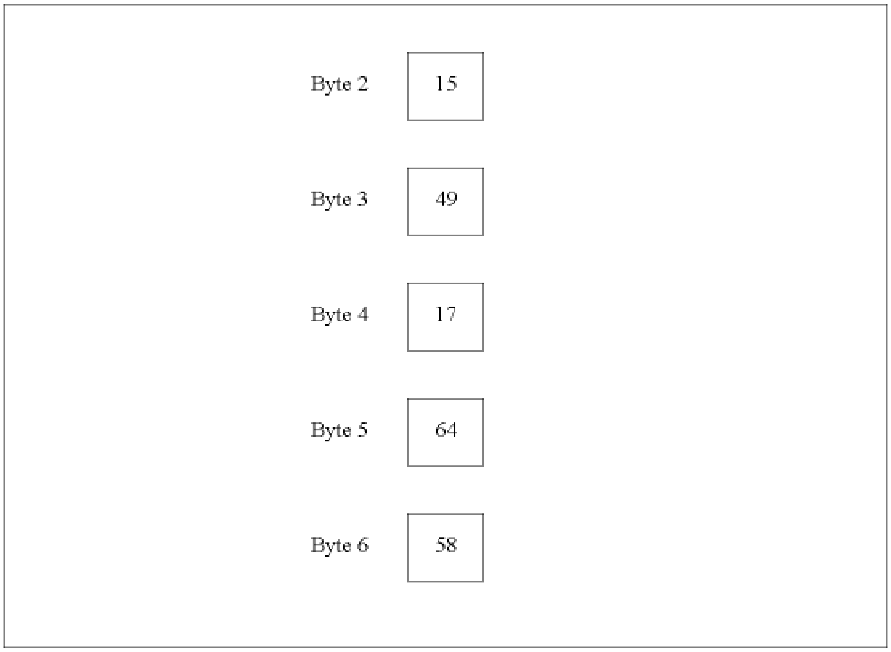
This is the internal ACD group number of the ACD group that the Supplementary ACD-DN is associated with.

2. SUPPL ACD DN:

This is a Supplementary ACD-DN of the ACD group identified by SUPPL_ACD_DN_GROUP. A Supplementary ACD-DN can be up to 10 digits long; each digit occupying 4 bits. The Supplementary ACD-DN is sent in BCD format. If the value of the SUPPL_ACD_DN_NUM_DIGS field is zero, then this field has no meaning.

Example: If a Supplementary ACD-DN is (519) 471-4685, the information in the 5 byte field is as follows:

Figure 15. Supplementary ACD-DN Example



Byte 2 15

Byte 3 49

Byte 4 17

Byte 5 64

Byte 6 58

3. SUPPL ACD DN NUM DIGS:

This number specifies the number of digits in the Supplementary ACD-DN. The supplementary ACD-DN can be up to 10 digits.

4. SUPPL ACD DN Identifier:

This is an identifier between zero and 16 that is used to identify the supplementary DN.

5. SUPPL ACD DN Priority:

This is the priority that calls to the supplementary ACD-DN are assigned when they are queued in the incoming call queue. It can be in the range zero to three inclusive.

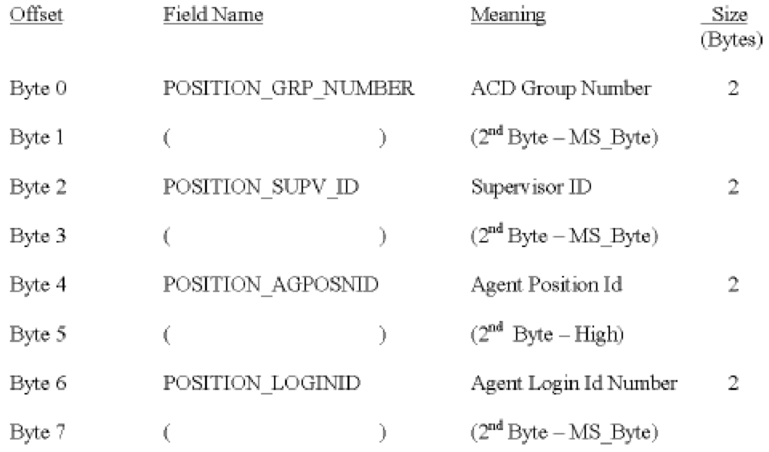
10.5. AGENTPOSITIONDATA

This section describes in detail the format of the AgentPosition Data argument.

Description: Sent to the DS processor to tell it about the current ACD configuration.

When Sent: Send once for each agent position datafilled in the switch at the time when the DSP is initialized by the switch.

Fields:



Offset	Field Name	Meaning	Size (Bytes)
Byte 0	POSITION_GRP_NUMBER	ACD Group Number	2
Byte 1	( )	(2nd Byte – MS_Byte)	
Byte 2	POSITION_SUPV_ID	Supervisor ID	2
Byte 3	( )	(2nd Byte – MS_Byte)	
Byte 4	POSITION_AGPOSNID	Agent Position Id	2
Byte 5	( )	(2nd Byte – High)	
Byte 6	POSITION_LOGINID	Agent Login Id Number	2
Byte 7	( )	(2nd Byte – MS_Byte)	

Fields Description:

1. Position Grp Number:

This is the internal ACD group number of the ACD group that the agent position belongs to. The group number is in the range zero to 255 inclusive.

2. Position Supv Id:

This is a unique 4 digit supervisor position identification number in the range 0001-9999 that is assigned to a supervisor position when it is datafilled. The supervisor and agent position Id numbers are taken from the same pool of available Id numbers. The supervisor position Id number identifies the supervisor position that the agent position has been associated with (if the agent position has been given this association). The supervisor position Id number is zero if the agent position has not been associated with a supervisor position.

3. Position AGPOSNID:

This is a unique 4 digit identification number in the range 0001-9999 assigned to each agent position at datafill time in the switch. The agent position ID number is zero if the agent position is not assigned a position ID. It is up to the DSP as to how it wants to handle this situation.

4. Position LoginId:

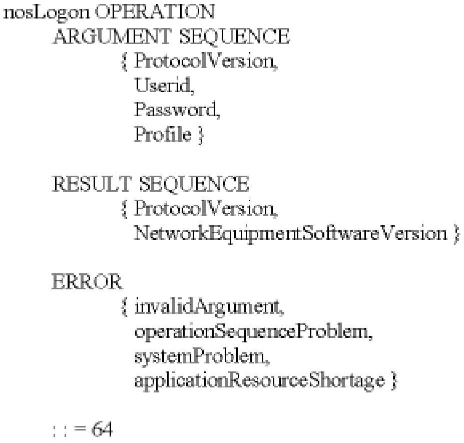
If an agent is logged into this agent position this message will contain the login Id of the agent. The login Id is a 4 digit identification number in the range 0001-9999. The login Id is zero if an agent is not logged into the agent position.

11. ACD MIS Remote Operation Definitions

The following are the RO and type definitions for the ACD Management Reports system described using the formal notation defined in CCITT Recommendation X.409.

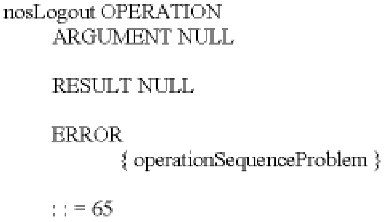
11.1. ACD MIS X.410 Operation Macros

11.1.1. nosLogon



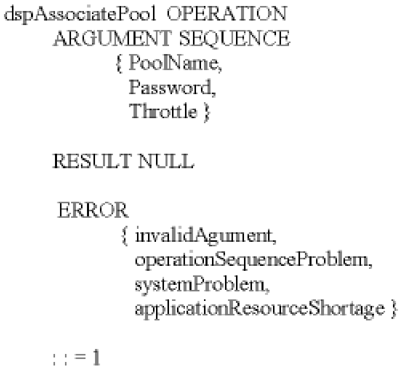
```
nosLogon OPERATION
      ARGUMENT SEQUENCE
               { ProtocolVersion,
                 Userid,
                 Password,
                 Profile }

      RESULT SEQUENCE
               { ProtocolVersion,
                 NetworkEquipmentSoftwareVersion }

      ERROR
               { invalidArgument,
                 operationSequenceProblem,
                 systemProblem,
                 applicationResourceShortage }

      : : = 64
```

11.1.2. nosLogout



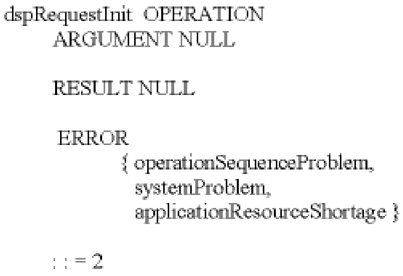
```
nosLogout OPERATION
      ARGUMENT NULL

      RESULT NULL

      ERROR
               { operationSequenceProblem }

      : : = 65
```

11.1.3. dspAssociatePool



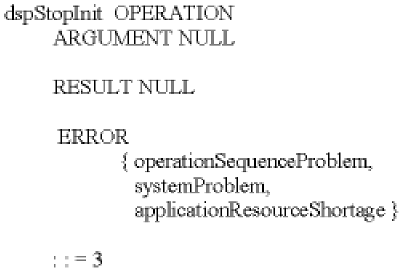
```
dspAssociatePool  OPERATION
       ARGUMENT SEQUENCE
                { PoolName,
                  Password,
                  Throttle }

       RESULT NULL

        ERROR
                 { invalidAgument,
                   operationSequenceProblem,
                   systemProblem,
                   applicationResourceShortage }

       : : = 1
```

11.1.4. dspRequestInit



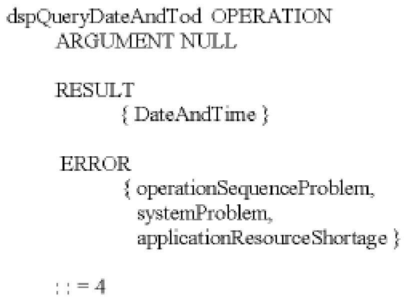
```
dspRequestInit  OPERATION
       ARGUMENT NULL

       RESULT NULL

        ERROR
                 { operationSequenceProblem,
                   systemProblem,
                   applicationResourceShortage }

       : : = 2
```

11.1.5. dspStopInit



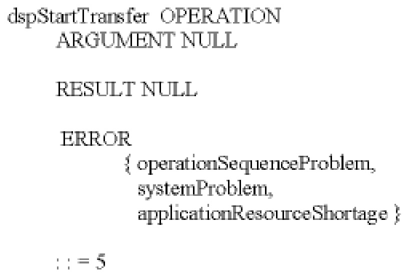
```
dspStopInit  OPERATION
       ARGUMENT NULL

       RESULT NULL

        ERROR
                 { operationSequenceProblem,
                   systemProblem,
                   applicationResourceShortage }

       : : = 3
```

11.1.6. dspQueryDateAndTod



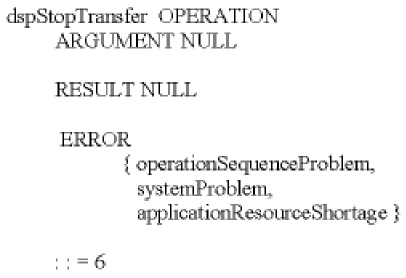
```
dspQueryDateAndTod OPERATION
       ARGUMENT NULL

       RESULT
                { DateAndTime }

        ERROR
                { operationSequenceProblem,
                  systemProblem,
                  applicationResourceShortage }

       : : = 4
```

11.1.7. dspStartTransfer



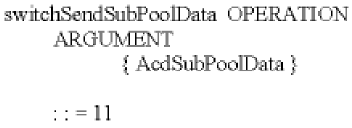
```
dspStartTransfer  OPERATION
       ARGUMENT NULL

       RESULT NULL

        ERROR
                { operationSequenceProblem,
                  systemProblem,
                  applicationResourceShortage }

       : : = 5
```

11.1.8. dspStopTransfer



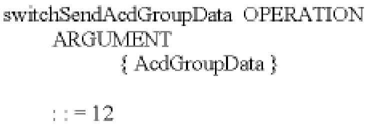
```
dspStopTransfer  OPERATION
       ARGUMENT NULL

       RESULT NULL

        ERROR
                { operationSequenceProblem,
                  systemProblem,
                  applicationResourceShortage }

       : : = 6
```

11.1.9. switchSendSubPoolData



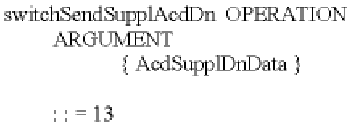
```
switchSendSubPoolData  OPERATION
       ARGUMENT
                { AcdSubPoolData }

       : : = 11
```

11.1.10. switchSendAcdGroupData



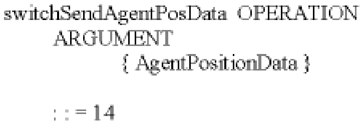
```
switchSendAcdGroupData  OPERATION
       ARGUMENT
                { AcdGroupData }

       : : = 12
```

11.1.11. switchSendSupplAcdDn



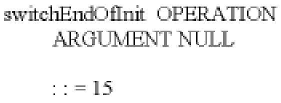
```
switchSendSupplAcdDn  OPERATION
       ARGUMENT
                { AcdSupplDnData }

       : : = 13
```

11.1.12. switchSendAgentPosData



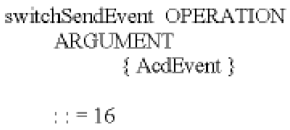
```
switchSendAgentPosData  OPERATION
       ARGUMENT
                { AgentPositionData }

       : : = 14
```

11.1.13. switchEndOfInit



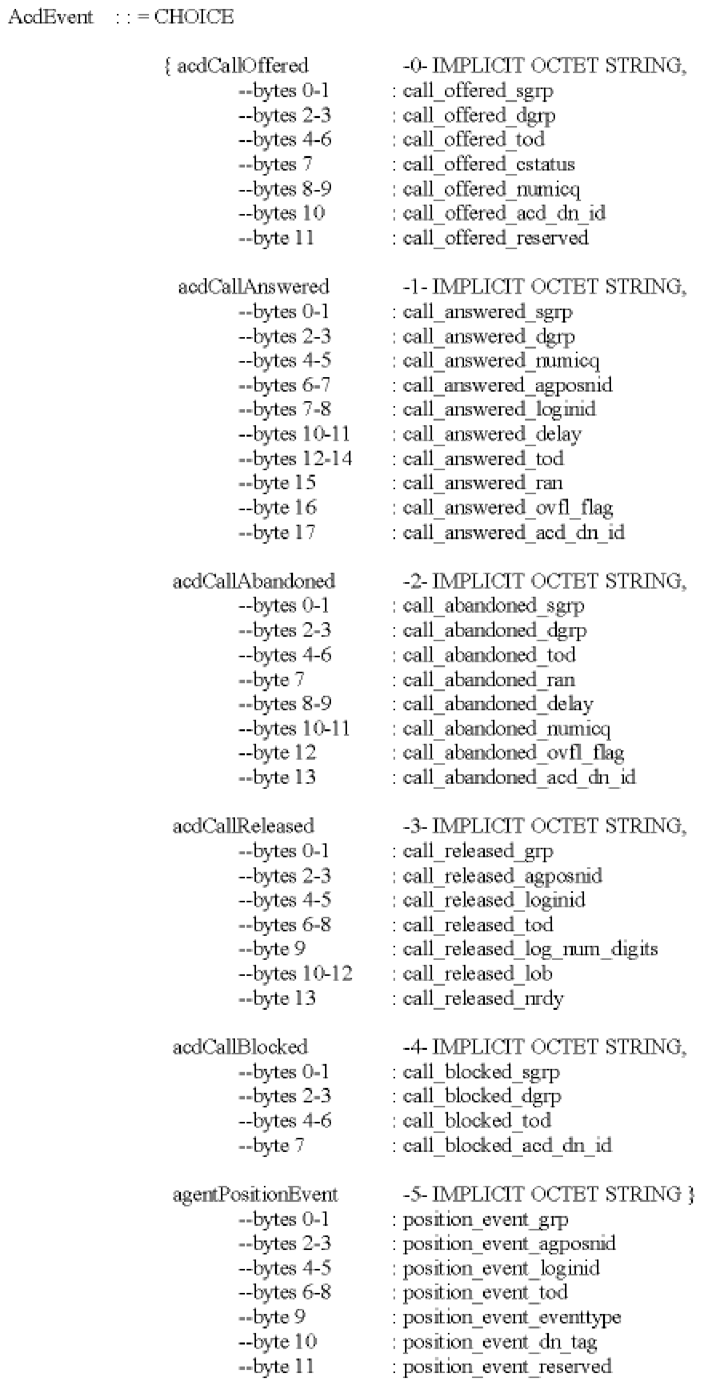
```
switchEndOfInit  OPERATION
       ARGUMENT NULL

       : : = 15
```

11.1.14. switchSendEvent

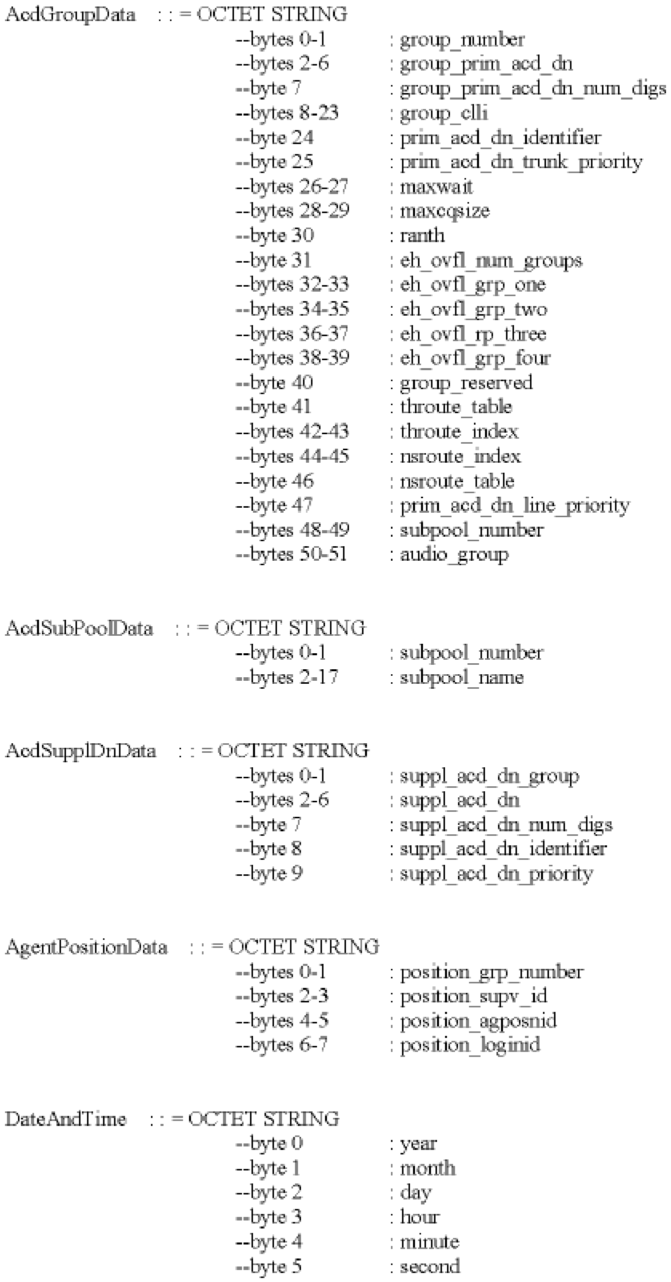


```
switchSendEvent  OPERATION
       ARGUMENT
                { AcdEvent }

      : : = 16
```

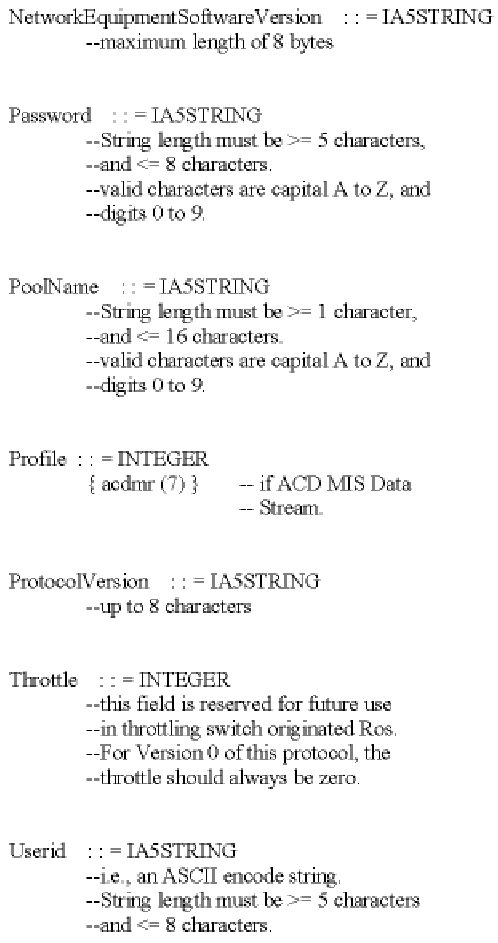
11.2. ACD MIS X.409 Type Definitions



```
AcdEvent   : : = CHOICE

                 { acdCallOffered          -0- IMPLICIT OCTET STRING,
                          --bytes 0-1     : call_offered_sgrp
                          --bytes 2-3     : call_offered_dgrp
                          --bytes 4-6     : call_offered_tod
                          --bytes 7       : call_offered_cstatus
                          --bytes 8-9     : call_offered_numicq
                          --bytes 10      : call_offered_acd_dn_id
                          --byte 11       : call_offered_reserved

                   acdCallAnswered         -1- IMPLICIT OCTET STRING,
                          --bytes 0-1     : call_answered_sgrp
                          --bytes 2-3     : call_answered_dgrp
                          --bytes 4-5     : call_answered_numicq
                          --bytes 6-7     : call_answered_agposnid
                          --bytes 7-8     : call_answered_loginid
                          --bytes 10-11   : call_answered_delay
                          --bytes 12-14   : call_answered_tod
                          --byte 15       : call_answered_ran
                          --byte 16       : call_answered_ovfl_flag
                          --byte 17       : call_answered_acd_dn_id

                   acdCallAbandoned        -2- IMPLICIT OCTET STRING,
                          --bytes 0-1     : call_abandoned_sgrp
                          --bytes 2-3     : call_abandoned_dgrp
                          --bytes 4-6     : call_abandoned_tod
                          --byte 7        : call_abandoned_ran
                          --bytes 8-9     : call_abandoned_delay
                          --bytes 10-11   : call_abandoned_numicq
                          --byte 12       : call_abandoned_ovfl_flag
                          --byte 13       : call_abandoned_acd_dn_id

                   acdCallReleased         -3- IMPLICIT OCTET STRING,
                          --bytes 0-1     : call_released_grp
                          --bytes 2-3     : call_released_agposnid
                          --bytes 4-5     : call_released_loginid
                          --bytes 6-8     : call_released_tod
                          --byte 9        : call_released_log_num_digits
                          --bytes 10-12   : call_released_lob
                          --byte 13       : call_released_nrdy

                   acdCallBlocked          -4- IMPLICIT OCTET STRING,
                          --bytes 0-1     : call_blocked_sgrp
                          --bytes 2-3     : call_blocked_dgrp
                          --bytes 4-6     : call_blocked_tod
                          --byte 7        : call_blocked_acd_dn_id

                   agentPositionEvent      -5- IMPLICIT OCTET STRING }
                          --bytes 0-1     : position_event_grp
                          --bytes 2-3     : position_event_agposnid
                          --bytes 4-5     : position_event_loginid
                          --bytes 6-8     : position_event_tod
                          --byte 9        : position_event_eventtype
                          --byte 10       : position_event_dn_tag
                          --byte 11       : position_event_reserved
```

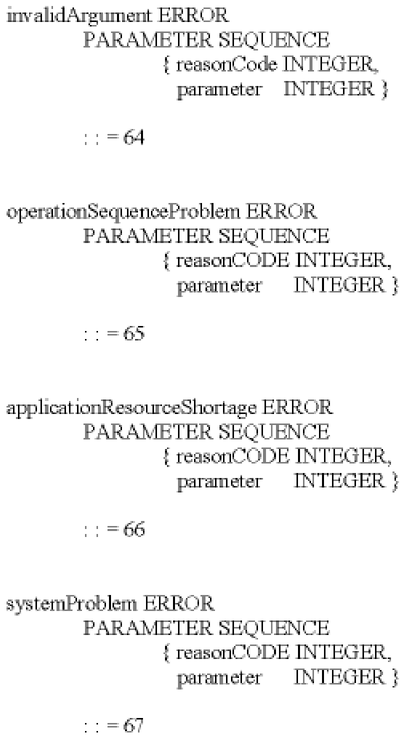


```
AcdGroupData   : : = OCTET STRING
                        --bytes 0-1       : group_number
                        --bytes 2-6       : group_prim_acd_dn
                        --byte 7          : group_prim_acd_dn_num_digs
                        --bytes 8-23      : group_clli
                        --byte 24         : prim_acd_dn_identifier
                        --byte 25         : prim_acd_dn_trunk_priority
                        --bytes 26-27     : maxwait
                        --bytes 28-29     : maxcqsize
                        --byte 30         : ranth
                        --byte 31         : eh_ovfl_num_groups
                        --bytes 32-33     : eh_ovfl_grp_one
                        --bytes 34-35     : eh_ovfl_grp_two
                        --bytes 36-37     : eh_ovfl_rp_three
                        --bytes 38-39     : eh_ovfl_grp_four
                        --byte 40         : group_reserved
                        --byte 41         : throute_table
                        --bytes 42-43     : throute_index
                        --bytes 44-45     : nsroute_index
                        --byte 46         : nsroute_table
                        --byte 47         : prim_acd_dn_line_priority
                        --bytes 48-49     : subpool_number
                        --bytes 50-51     : audio_group


AcdSubPoolData   : : = OCTET STRING
                        --bytes 0-1       : subpool_number
                        --bytes 2-17      : subpool_name


AcdSupplDnData   : : = OCTET STRING
                        --bytes 0-1       : suppl_acd_dn_group
                        --bytes 2-6       : suppl_acd_dn
                        --byte 7          : suppl_acd_dn_num_digs
                        --byte 8          : suppl_acd_dn_identifier
                        --byte 9          : suppl_acd_dn_priority


AgentPositionData   : : = OCTET STRING
                        --bytes 0-1       : position_grp_number
                        --bytes 2-3       : position_supv_id
                        --bytes 4-5       : position_agposnid
                        --bytes 6-7       : position_loginid


DateAndTime   : : = OCTET STRING
                        --byte 0          : year
                        --byte 1          : month
                        --byte 2          : day
                        --byte 3          : hour
                        --byte 4          : minute
                        --byte 5          : second
```



```
NetworkEquipmentSoftwareVersion    : : = IA5STRING
          --maximum length of 8 bytes


Password    : : = IA5STRING
          --String length must be >= 5 characters,
          --and <= 8 characters.
          --valid characters are capital A to Z, and
          --digits 0 to 9.


PoolName    : : = IA5STRING
          --String length must be >= 1 character,
          --and <= 16 characters.
          --valid characters are capital A to Z, and
          --digits 0 to 9.


Profile  : : = INTEGER
          { acdmr (7) }        -- if ACD MIS Data
                               -- Stream.


ProtocolVersion    : : = IA5STRING
          --up to 8 characters


Throttle    : : = INTEGER
          --this field is reserved for future use
          --in throttling switch originated Ros.
          --For Version 0 of this protocol, the
          --throttle should always be zero.


Userid    : : = IA5STRING
          --i.e., an ASCII encode string.
          --String length must be >= 5 characters
          --and <= 8 characters.
```

11.3. Error Macros



```
invalidArgument ERROR
          PARAMETER SEQUENCE
                    { reasonCode INTEGER,
                      parameter    INTEGER }

          : : = 64


operationSequenceProblem ERROR
          PARAMETER SEQUENCE
                    { reasonCODE INTEGER,
                      parameter     INTEGER }

          : : = 65


applicationResourceShortage ERROR
          PARAMETER SEQUENCE
                    { reasonCODE INTEGER,
                      parameter     INTEGER }

          : : = 66


systemProblem ERROR
          PARAMETER SEQUENCE
                    { reasonCODE INTEGER,
                      parameter     INTEGER }

          : : = 67
```

See Section 11.4 for defined reason codes and parameter codes returned by the switch.

11.4. Error Code Information

Error codes sent from the switch to the DSP are mapped into a standard template. The template has three fields:

1. TYPE

The type of the error is encoded as a NOP integer in the range 64 to 67 (decimal).

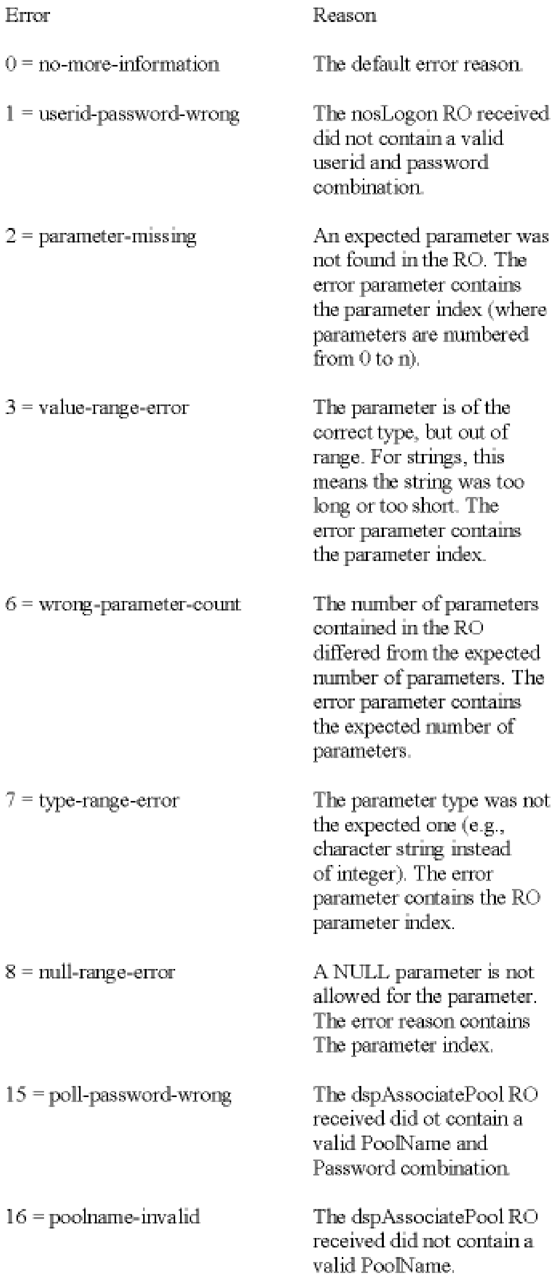
2. REASON

The error reason is a refinement of the type and provides the actual reason the error occurred. The error reason is encoded as a NOP integer in the range zero to 127 (decimal).

3. PARAMETER

11.4.1. Invalid Argument Errors

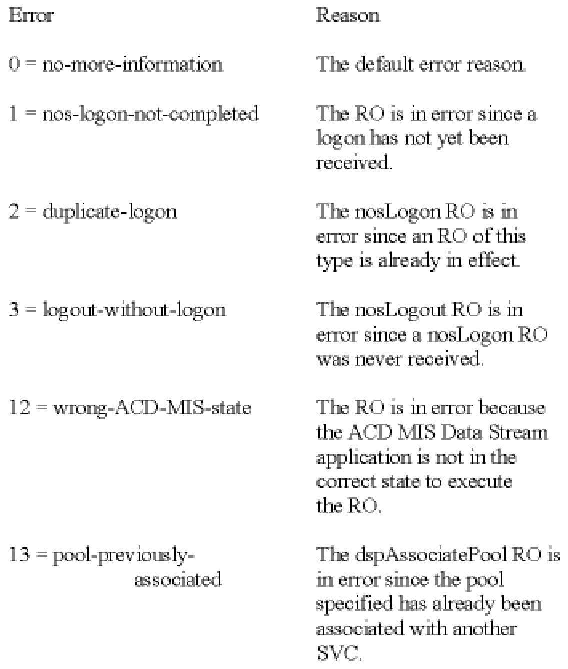
The RO parameters received from the DSP are checked before any other checks are performed. Errors detected at this stage stop further processing of the RO. The error reasons and their related parameters are outlined below:



Error	Reason
0 = no-more-information	The default error reason.
1 = userid-password-wrong	The nosLogon RO received did not contain a valid userid and password combination.
2 = parameter-missing	An expected parameter was not found in the RO. The error parameter contains the parameter index (where parameters are numbered from 0 to n).
3 = value-range-error	The parameter is of the correct type, but out of range. For strings, this means the string was too long or too short. The error parameter contains the parameter index.
6 = wrong-parameter-count	The number of parameters contained in the RO differed from the expected number of parameters. The error parameter contains the expected number of parameters.
7 = type-range-error	The parameter type was not the expected one (e.g., character string instead of integer). The error parameter contains the RO parameter index.
8 = null-range-error	A NULL parameter is not allowed for the parameter. The error reason contains The parameter index.
15 = poll-password-wrong	The dspAssociatePool RO received did ot contain a valid PoolName and Password combination.
16 = poolname-invalid	The dspAssociatePool RO received did not contain a valid PoolName.

11.4.2. Operation Sequence Errors

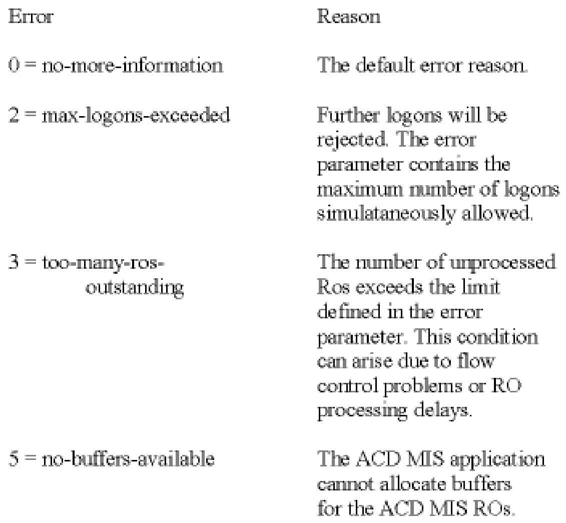
This error type is used to report errors that occur when ROs are received in an incorrect order. This can be caused by problems at either the switch or DSP end. If the problem persist, then a nosLogout RO should be performed since this RO is always processed by the switch. The error reasons follow:



Error	Reason
0 = no-more-information	The default error reason.
1 = nos-logon-not-completed	The RO is in error since a logon has not yet been received.
2 = duplicate-logon	The nosLogon RO is in error since an RO of this type is already in effect.
3 = logout-without-logon	The nosLogout RO is in error since a nosLogon RO was never received.
12 = wrong-ACD-MIS-state	The RO is in error because the ACD MIS Data Stream application is not in the correct state to execute the RO.
13 = pool-previously-associated	The dspAssociatePool RO is in error since the pool specified has already been associated with another SVC.

11.4.3. Application Resource Shortage Errors

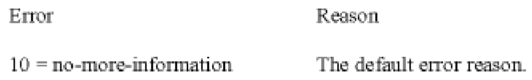
This error type relates to problems encountered with resource utilization on the switch. The error reasons are:



Error	Reason
0 = no-more-information	The default error reason.
2 = max-logons-exceeded	Further logons will be rejected. The error parameter contains the maximum number of logons simulataneously allowed.
3 = too-many-ros-outstanding	The number of unprocessed Ros exceeds the limit defined in the error parameter. This condition can arise due to flow control problems or RO processing delays.
5 = no-buffers-available	The ACD MIS application cannot allocate buffers for the ACD MIS ROs.

11.4.4. System Problem Errors

This error type encompasses all error reasons that can occur while processing an RO. The error reasons are:



Error	Reason
10 = no-more-information	The default error reason.

12. Implementation Restrictions

The following restrictions are placed on the ACD Management Information System Data Stream application protocol:

1. The largest LENGTH field in a switch generated RO is less than 127 (decimal). Therefore, only a short form for the length field will be needed.

2. Invoke Ids in both the switch and DSP generated OPDUs must be in the range of zero to 127 (decimal).

3. No more than 10 ROs invoked from the DSP can be outstanding and not responded to by the switch at any one point in time.

4. The ACD information sent relays details of ACD events that are occurring in the switch. The ACD event information is transmitted as soon as possible after the events occur. However, if the switch is heavily loaded with call traffic when an ACD event occurs, there may be a delay between the time that an ACD event occurs and the time that the event information is transmitted to the DSP. The transmission delay encountered is dependent on the call traffic load on the switch.

5. The ACD Management Information System Data Stream is a means of providing a DSP with information in near real time of ACD events occurring in a switch. If anything happens such that the switch cannot transmit event information to the DSP (e.g., datalink does down, DSP crashes), the information about events that occur while the switch cannot communicate with the DSP will be lost since the switch will not save the information until the communication is reestablished.

6. If DSP originated ROs are still outstanding when the switch receives a valid nosLogout RO, the responses to the DSP originated ROs will not be transmitted to the DSP. The DSP will only receive a response to the nosLogout RO.

7. There may only be one (1) Remote Operation (RO) X.25 packet.

13. Remote Operation Examples

This section contains encoding examples for the remote operations defined in Section 11.0.

NOTE: Unless otherwise specified, all numbers in these examples are hexadecimal.

13.1. X.409 Encoding Overview

The ’data element’ is a key concept in X.409 notation Section 3 of CCITT Recommendation X.409 provides a detailed description of data elements. A brief overview of data elements is given below.

A data element consists of:

- an identifier part,
- a length part, and
- a contents part.

The identifier part of a data element consists of:

- a Class part,

- a Form part, and
- an Id code.

The choices for the Class of an identifier are:

- Universal
- Application Wide
- Context specific
- Private use

The choices for the Form of an identifier are:

- primitive, i.e., contents are atomic
- constructor, i.e., contents are defined by other data elements

The Id code allows multiple identifiers with the same class and form to be defined. The following illustrates the usage of the identifier part:

- -APPLICATION 10-

This denotes an identifier whose class is Application wide and whose Id code is 10. The form of this identifier would be determined by the definition of the content part for the data element denoted by this identifier.

- -30-

This denotes an identifier whose class is Context specific and whose Id code is 30. As above, the form is dependent on the associated content part.

The assignment of identifiers in the Universal class is controlled by the CCITT. Application wide and Context specific class identifiers will be assigned to meet ACD Management Information System requirements. Private identifiers will not be used.

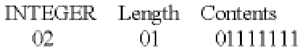
The length part of a data element gives the length of the contents part in bytes. (X.409 also allows for an indefinite length format where the contents part of a data element contains a unique element to indicate the end of the contents part.)

Data Element Example

Universal identifiers assigned in X.409 include:

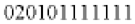
- BOOLEAN
- INTEGER
- OCTET STRING
- NULL
- etc.

A data element of type INTEGER with a decimal value of ’127’ would be encoded as:



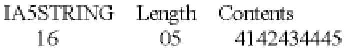
INTEGER	Length	Contents
02	01	01111111

The hex sequence sent to represent this data element would be:



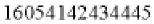
020101111111

A data element of type IA5STRING with a value of ’ABCDE’ would be encoded as:



IA5STRING	Length	Contents
16	05	4142434445

The hex sequence sent to represent this data element would be:



16054142434445

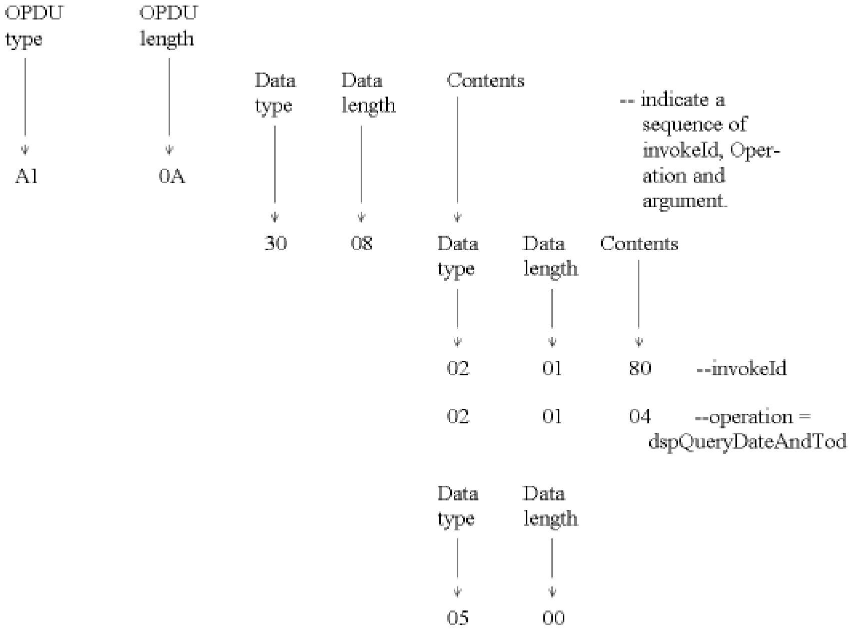
13.2. ACD MIS RO Encoding Detailed Description

The following is an explanation of how encoding is performed for the Invoke part of a Remote Operation. The dspQueryDateAndTod RO is used as the example.

An Invoke may be thought of as a ’message’, with a predefined structure and purpose. The OPDU type, in this case an Invoke, is flogged by the value ’A1’ in the first byte. The next byte contains the number of bytes contained in the message, in this example ’OA’. The next byte indicates that the information to follow is a sequence, or list, of discrete items. The next byte indicates the number of bytes in the sequence (excluding this byte), in this case ’08’. The next group of bytes contains the items that make up the sequence. The first byte of each item identifies the type of data contained in the item. The next byte contains the length of the data. The remaining byte(s) in each item make up the actual data. In this example, the sequence contains three items; the first two items are integers (indicated by the value ’02’) and the third item is NULL (indicated by ’05’).

The structure of this OPDU is formally defined in Section 11.1.6. Every OPDU defined for the ACD Management Information System protocol uses this basic structure.

Encoding Example:

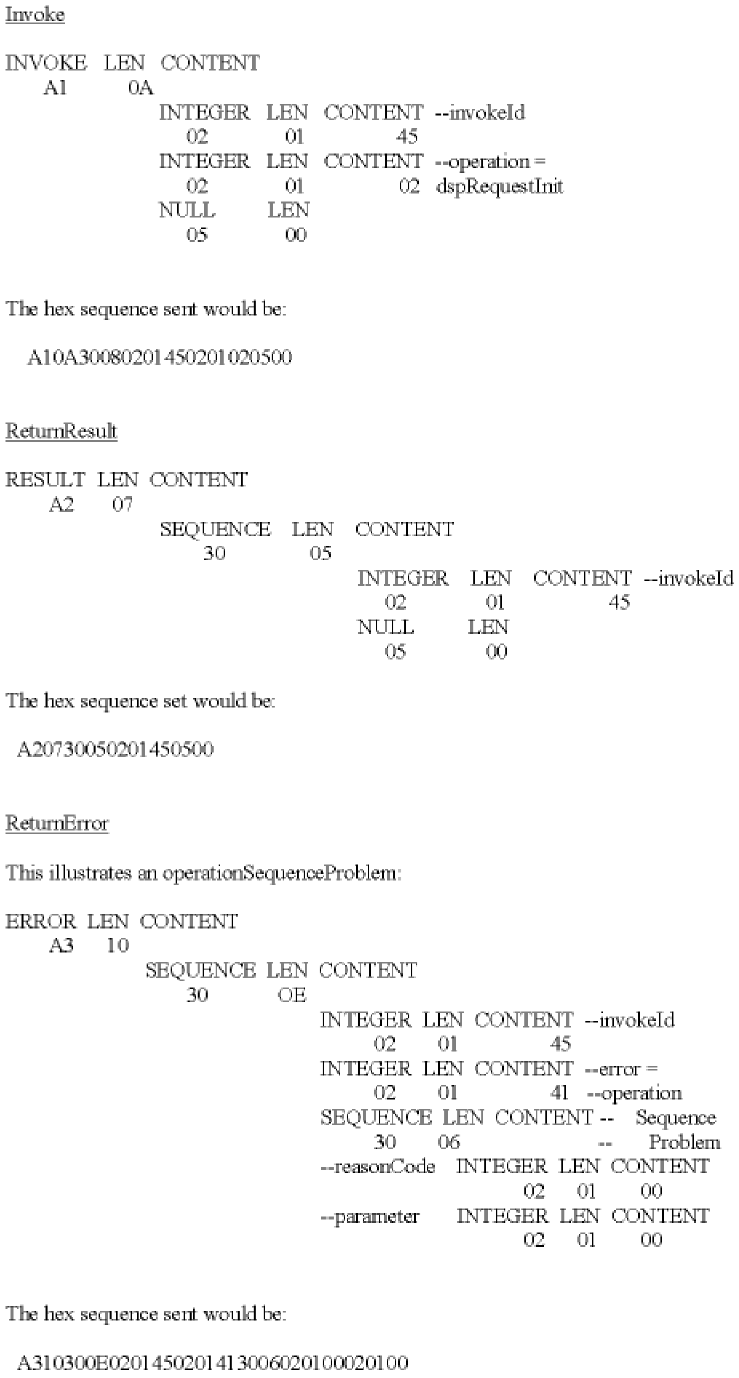


The hex sequence sent would be:

A10A30080201800201040500

13.3. ACD MIS RO Encoding Examples

13.3.1. dspRequestInit



Invoke

```
INVOKE  LEN  CONTENT
    A1       0A
                 INTEGER  LEN  CONTENT  --invokeId
                    02     01        45
                 INTEGER  LEN  CONTENT  --operation =
                    02     01        02  dspRequestInit
                 NULL      LEN
                    05     00
```

The hex sequence sent would be:

```
  A10A30080201450201020500
```

ReturnResult

```
RESULT  LEN  CONTENT
     A2     07
                 SEQUENCE   LEN   CONTENT
                     30       05
                                   INTEGER   LEN   CONTENT  --invokeId
                                      02       01         45
                                   NULL      LEN
                                      05       00
```

The hex sequence set would be:

```
 A2073005020145 0500
```

ReturnError

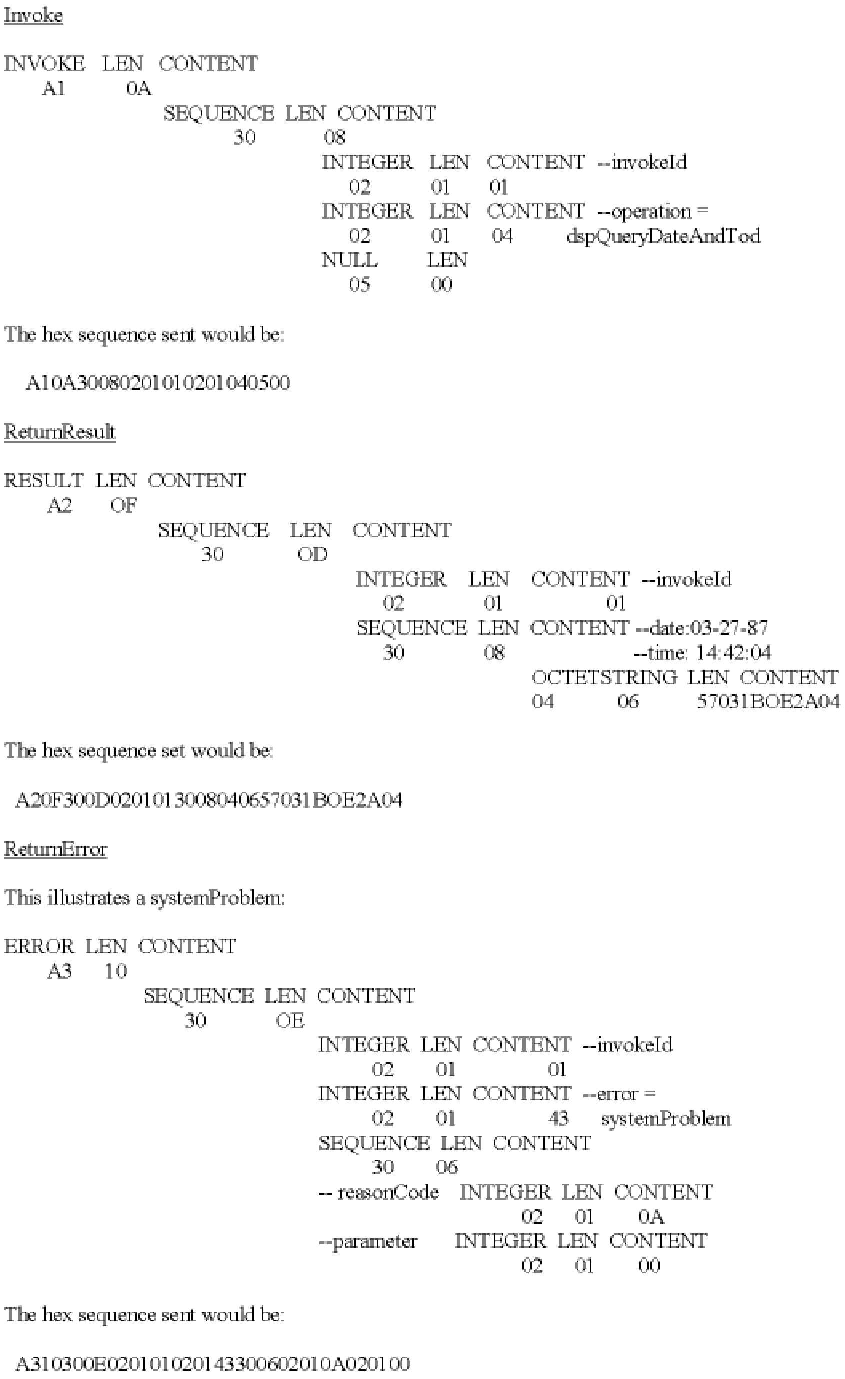
This illustrates an operationSequenceProblem:

```
ERROR  LEN  CONTENT
    A3    10
               SEQUENCE  LEN  CONTENT
                   30       OE
                               INTEGER  LEN  CONTENT  --invokeId
                                    02     01          45
                               INTEGER  LEN  CONTENT  --error =
                                    02     01          41  --operation
                               SEQUENCE  LEN  CONTENT --   Sequence
                                    30     06              --    Problem
                               --reasonCode   INTEGER  LEN  CONTENT
                                                   02     01      00
                               --parameter    INTEGER  LEN  CONTENT
                                                   02     01      00
```

The hex sequence sent would be:

```
 A310300E0201450201413006020100020100
```

13.3.2. dspQueryDateAndTod



Invoke

```
INVOKE  LEN  CONTENT
   A1      0A
             SEQUENCE  LEN  CONTENT
                   30       08
                            INTEGER  LEN   CONTENT  --invokeId
                               02     01    01
                            INTEGER  LEN   CONTENT  --operation =
                               02     01    04      dspQueryDateAndTod
                            NULL      LEN
                               05     00
```

The hex sequence sent would be:

```
  A10A3008020101020104050O
```

ReturnResult

```
RESULT  LEN  CONTENT
    A2    OF
            SEQUENCE   LEN   CONTENT
                 30        OD
                               INTEGER   LEN   CONTENT  --invokeId
                                  02      01          01
                               SEQUENCE  LEN  CONTENT --date:03-27-87
                                  30      08                --time: 14:42:04
                                              OCTETSTRING  LEN  CONTENT
                                              04        06       57031BOE2A04
```

The hex sequence set would be:

```
 A20F300D0201013008040657031BOE2A04
```

ReturnError

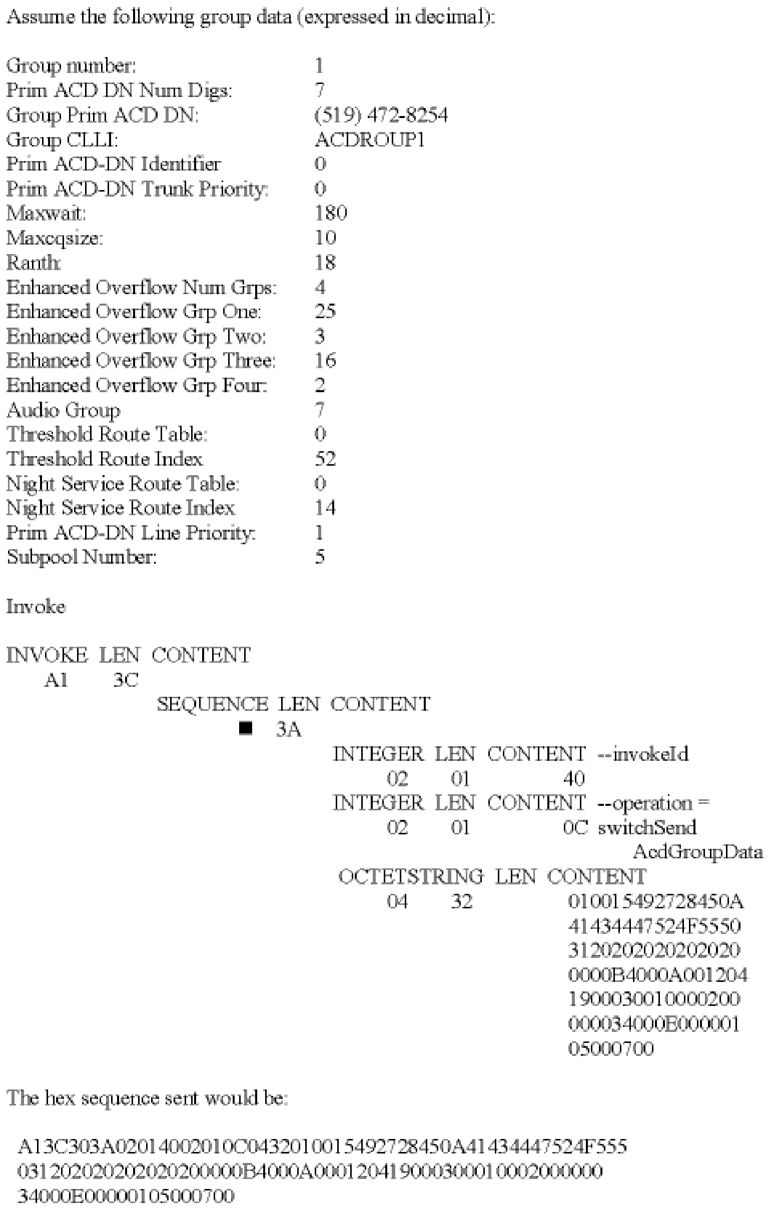
This illustrates a systemProblem:

```
ERROR  LEN  CONTENT
    A3    10
           SEQUENCE  LEN  CONTENT
                30        OE
                              INTEGER  LEN  CONTENT  --invokeId
                                   02    01         01
                              INTEGER  LEN  CONTENT  --error =
                                   02    01         43    systemProblem
                              SEQUENCE  LEN  CONTENT
                                   30    06
                              -- reasonCode   INTEGER  LEN  CONTENT
                                                   02    01     0A
                              --parameter    INTEGER  LEN  CONTENT
                                                   02    01     00
```

The hex sequence sent would be:

```
 A310300E020101020143300602010A020100
```

13.3.3. switchSendAcdGroupData



Assume the following group data (expressed in decimal):

Group number:	1
Prim ACD DN Num Digs:	7
Group Prim ACD DN:	(519) 472-8254
Group CLLI:	ACDROUP1
Prim ACD-DN Identifier	0
Prim ACD-DN Trunk Priority:	0
Maxwait:	180
Maxcqsize:	10
Ranth:	18
Enhanced Overflow Num Grps:	4
Enhanced Overflow Grp One:	25
Enhanced Overflow Grp Two:	3
Enhanced Overflow Grp Three:	16
Enhanced Overflow Grp Four:	2
Audio Group	7
Threshold Route Table:	0
Threshold Route Index	52
Night Service Route Table:	0
Night Service Route Index	14
Prim ACD-DN Line Priority:	1
Subpool Number:	5

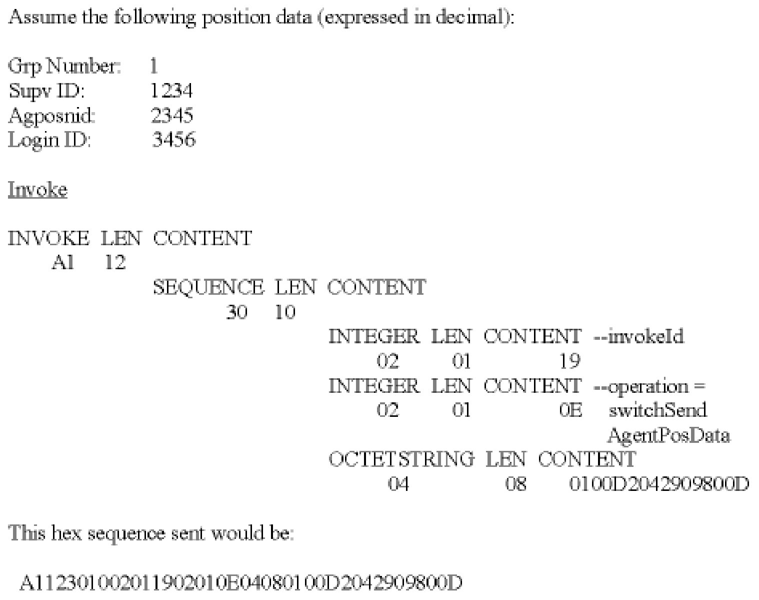
Invoke

```
INVOKE  LEN  CONTENT
   A1    3C
            SEQUENCE  LEN  CONTENT
                  ■    3A
                           INTEGER  LEN  CONTENT  --invokeId
                               02    01        40
                           INTEGER  LEN  CONTENT  --operation =
                               02    01        0C  switchSend
                                                      AcdGroupData
                            OCTETSTRING  LEN  CONTENT
                               04    32          010015492728450A
                                                 41434447524F5550
                                                 3120202020202020
                                                 0000B4000A001204
                                                 1900030010000200
                                                 000034000E000001
                                                 05000700
```

The hex sequence sent would be:

```
A13C303A020140020100C043201001549272845 0A41434447524F555
031202020202020200000B4000A000120419000300010002000000
34000E00000105000700
```

13.3.4. switchSendAgentPosData



Assume the following position data (expressed in decimal):

Grp Number:	1
Supv ID:	1234
Agposnid:	2345
Login ID:	3456

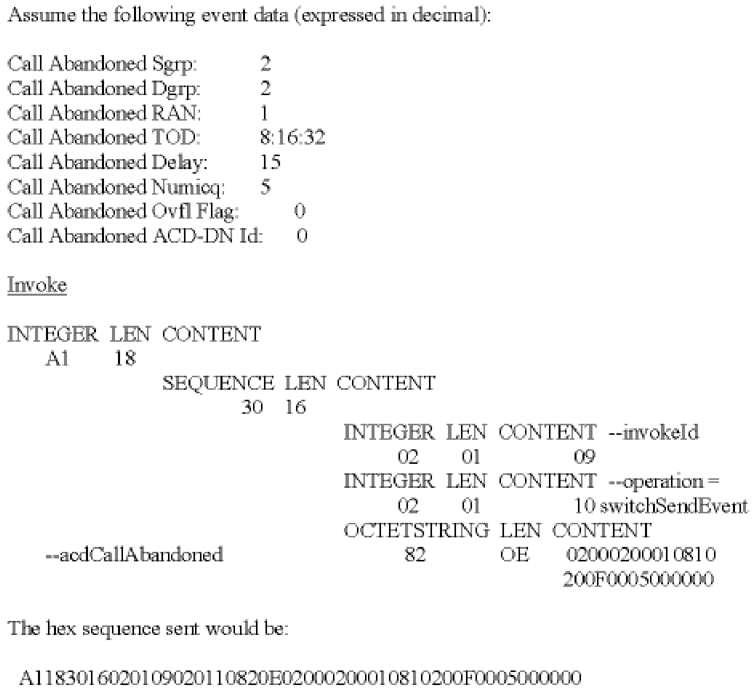
Invoke

```
INVOKE  LEN  CONTENT
     A1   12
                SEQUENCE  LEN  CONTENT
                        30    10
                                  INTEGER  LEN  CONTENT  --invokeId
                                        02      01          19
                                  INTEGER  LEN  CONTENT  --operation =
                                        02      01          0E   switchSend
                                                                 AgentPosData
                                  OCTETSTRING  LEN  CONTENT
                                         04          08      0100D2042909800D
```

This hex sequence sent would be:

```
A112301002011902010E04080100D2042909800D
```

13.3.5. switchSendEvent



```
Assume the following event data (expressed in decimal):

Call Abandoned Sgrp:          2
Call Abandoned Dgrp:          2
Call Abandoned RAN:           1
Call Abandoned TOD:           8:16:32
Call Abandoned Delay:         15
Call Abandoned Numicq:        5
Call Abandoned Ovfl Flag:        0
Call Abandoned ACD-DN Id:        0

Invoke

INTEGER  LEN  CONTENT
    A1      18
                SEQUENCE  LEN  CONTENT
                        30   16
                                INTEGER  LEN  CONTENT  --invokeId
                                     02      01            09
                                INTEGER  LEN  CONTENT  --operation =
                                     02      01            10 switchSendEvent
                                OCTETSTRING  LEN  CONTENT
    --acdCallAbandoned                82          OE      02000200010810
                                                         200F0005000000

The hex sequence sent would be:

 A1183016020109020110820E02000200010810200F0005000000
```

14. ACD Event Scenarios

The ACD event scenarios outlined in this section will not give the reader an exhaustive list of all possible ACD event message sequences.

It is intended to be used as a reference of possible event message sequences the DSP can receive when using the protocol defined in this specification.

14.1. TITLE

Action	ACD MIS switchSendEvent RO
A call is offered to an ACD group and is presented to an agent position without being queued in the incoming call queue.	acdCallOffered - Source and dstination ACD gorup fields are the same. Call status field is zero.
The ACD agent answers the call.	acdCallAnswered - Source and destination ACD group fields are the same.
The ACD agent releases the call using the Release key.	acdCallReleased - Not Ready flag field is zero.

14.2. ACD Event Scenario II

Action	ACD MIS switchSendEvent RO
A call is offered to an ACD group and is queued in the incoming call queue.	acdCallOffered - Source and destination ACD group fields are the same. Call status field is one.
An ACD agent position becomes available. The call is pressented to the agent position.	No ACD MIS RO is sent.
The ACD agent answers the call.	acdCallAnswered - Source and destination ACD group fields are the same.
The ACD agent releases the call using the Not Ready key.	acdCallReleased - Not Ready flag field is one.
The ACD agent deactivates the Not Ready key to make the agent position available for another ACD call.	agentPositionEvent - Event type field is three (Not Ready deactivation).

14.3. ACD Event Scenario III

Action	ACD MIS switchSendEvent RO
A call is offered to an ACD group but the group cannot accept the call because if is in night service. The call is deflected to the ACD group's night service route which is to another ACD group.	acdCallOffered - Source and destination ACD group fields are the same. Call status field is four.
The call is offered to the night service ACD group and presented to an agent position without being queued in the incoming call queue.	acdCallOffered - Source ACD group field is the initial ACD group that the call was offered to. Destination ACD group field is the group now servicing the call. Call status field is zero.
The ACD agent answers the call.	acdCallAnswered - Source and destination ACD group fields are as described in the box directly above.
The ACD group releases the call using the Release key.	acdCallReleased - Not Ready flag field is zero.

14.4. ACD Event Scenario IV

Action	ACD MIS switchSendEvent RO
A call is offered to an ACD group but the group cannot accept the call because its imcoming call queue is full. The call is deflected to the ACD group's threshold route which is to another ACD group.	acdCallOffered - Source and destination ACD group fields are the same. Call status field is zero.
The call is offered to the ACD group that the initial group overflowed to. This ACD group cannot queue the call either because its incoming call queue is full. The call is deflected to another ACD group through the enhanced overflow feature.	acdCallOffered - Source ACD group field is the initial ACD group that the call was offered to. Destination ACD group field is the group that is now deflecting the call. The call status field is two.

The call is offered to the ACD group that the enhanced overflow feature decided could accept it. The call is presented to an agent position without being queued in the incoming call queue.	acdCallOffered - Source ACD group field is the initial ACD group that the call was offered to. Destination ACD group is the group that is now servicing the call. The call status field is zero.
The ACD agent answers the call.	acdCallAnswered - Source ACD group field is the initial ACD group that the call was offered to. Destination ACD group is the group that is now servicing the call.
The ACD agent releases the call using the Release key.	acdCallReleased - Not Ready flag field is zero.

14.5. ACD Event Scenario V

Action	ACD MIS switchSendEvent RO
A call is offered to an ACD group but the group cannot accept the call because its incoming call queue is full. The call is deflected to the ACD group's threshold route which is to another ACD grouop.	acdCallOffered - Source and destination ACD group fields are the same. Call status field is two.
The call is offered to the ACD group that the initial group overflowed to. This ACD group cannot queue the call either because its incoming call queue is full. The ACD group is not equipped with the enhanced overflow feature. The call is routed to busy treatment.	acdCallOffered - Source ACD group field is the initial ACD group that the call was offered to. Destination ACD group field is the group that the call is now being presented to but cannot service the call. Call status field is three.

14.6. ACD Event Scenario VI

Action	ACD MIS switchSendEvent RO
A call is offered to an ACD group and is ringing an ACD agent position.	acdCallOffered - Source and destination ACD group fields are the same. Call status field is zero.
The caller tires of waiting and abandons the calls.	acdCallAbandoned - Source and destination ACD group fields are the same.

14.7. ACD Event Scenario VII

Action	ACD MIS switchSendEvent RO
An agent completes a valid login sequence to log into an ACD agent position. The Make Set Busy feature is deactivated, and the agent position is placed in the Not Ready state.	agentPositionEvent - Event type field is zero (agent login).
The agent presses the Not Ready key to remove the position from the Not Ready state and make it available to receive ACD calls.	agentPositionEvent - Event type field is three (Not Ready deactivation).

14.8. ACD Event Scenario VIII

Action	ACD MIS switchSendEvent RO
A call is offered to an ACD group and is presented to an agent position without being queued in the incoming call queue.	acdCallOffered - Source and destination ACD group fields are the same. Call status field is zero.
The ACD agent presses the Not Ready key to remove the ringing ACD call from the agent position.	agentPositionEvent - Event type field is two (Not Ready activation).
The call is presented to the next available agent position in the ACD group.	No ACD MIS RO is sent.

The agent answers the call.	acdCallAnswered - Source and destination ACD group fields are the same.
The ACD agent releases the call using the Release key.	acdCallReleased - Not Ready flag field is zero.

14.9. ACD Event Scenario IX

Action	ACD MIS switchSendEvent RO
A call is offered to an ACD group and is presented to an agent position without being queues in the incoming call queue.	acdCallOffered - Source and destination ACD group fields are the same. Call status field is zero.
The agent answers the ACD call.	acdCallAnswered - Source and destination ACD group fields are the same.
The ACD agent activates the Make Set Busy feature because the agent wants to tbe logged out after servicing the current ACD call.	agentPositionEvent - Event type field is seven (Make Set Busy activation).
The ACD agent releases the call using the Release key.	acdCallReleased - Not Ready flag field is zero. agent PositionEvent - Event type field is one (Agent logout).

14.10. ACD Event Scenario X

Action	ACD MIS switchSendEvent RO
An agent has already logged into an agent position. The agent presses a DN key on the agent position to answer an incoming DN call.	agentPositionEvent - Event type field is five (answer a DN call).
The agent finishes the DN call and presses the Release key to release the call.	agentPositionEvent - Event type field is six (release a DN call).

14.11. ACD Event Scenario XI

Action	ACD MIS switchSendEvent RO
An event occurs on an agent position that does not have an agent logged into it.	No ACD MIS RO is sent.

14.12. ACD Event Scenario XII

Action	ACD MIS switchSendEvent RO
A call is offered to an ACD group but the group cannot accept the call because its incoming call queue is full. The call is deflected to the ACD group's threshold route which is to an announcement.	acdCallOffered - Source and destination ACD group fields are the same. Call status field is two.

14.13. ACD Event Scenario XIII

Action	ACD MIS switchSendEvent RO
A call is offered to an ACD group and is queued in the ACD group's incoming call queue.	acdCallOffered - Source and destination ACD group fields are the same. Call status field is one.
The caller tires of waiting and abandons the call.	acdCallAbandoned - Source and destination ACD group fields are the same.

14.14. ACD Event Scenario XIV

Action	ACD MIS switchSendEvent RO
An agent has already logged into an agent position. The agent presses a DN key on the agent position to originate an outgoing DN call.	agentPositionEvent - Event type field is four (originate a DN call).
The agent finishes the DN call and presses the Release key to release the call.	agentPositionEvent - Event type field is six (release a DN call).