Oracle® Communications Network Charging and Control

ACS Protocol Implementation Conformance Statement

Release 15.2

January 2026

ORACLE®

Copyright

Contents

About This Document

Scope

This statement of compliance compares the Oracle Communications Network Charging and Control (NCC) Advanced Control Services (ACS) component to the following protocol specifications:

- ETSI INAP: *ETS 300 374-1 Intelligent Network Capability Set 1 (CS1); Core Intelligent Network Application Protocol (INAP);Part 1: Protocol specification*
- CS2 INAP: *ITU-T Q1228: Interface Recommendation for intelligent network Capability Set 2: Part 1*
- CAP2: *ETSI TS 101 046 V7.1.0 CAMEL Application Part (CAP) specification (GSM 09.78 version 7.1.0 Release 1998)*
- CAP3: *ETSI TS 129 078 V4.8.0 CAMEL Application Part (CAP) specification (3GPP TS 29.078 version 4.8.0 Release 4)*
- CAP4: *3GPP TS 29.078 V11.2.0 (CAMEL) Phase 4; CAMEL Application Part (CAP) specification (Release 11)*
- MAP: *GPP TS 29.002 V7.2.0 Mobile Application Part (MAP) specification; (Release 7)*

Audience

This document is intended for use by software engineers and testers who need a description of the message sequences used by ACS.

Related Documents

The following documents are related to this document:

- *Advanced Control Services Technical Guide*
- *Charging Control Services Technical Guide*
- *Charging Control Services User's Guide*
- *Service Management System Technical Guide*
- *Service Management System User's Guide*
- *Service Logic Execution Environment Technical Guide*

Document Conventions

Typographical Conventions

The following terms and typographical conventions are used in the Oracle Communications Network Charging and Control (NCC) documentation.

Formatting Convention	Type of Information
Special Bold	Items you must select, such as names of tabs.
	Names of database tables and fields.
Italics	Name of a document, chapter, topic or other publication.
	Emphasis within text.
Button	The name of a button to click or a key to press.
	Example: To close the window, either click **Close**, or press **Esc**.
Key+Key	Key combinations for which the user must press and hold down one key and then press another.
	Example: **Ctrl+P** or **Alt+F4**.
`Monospace`	Examples of code or standard output.
`Monospace Bold`	Text that you must enter.
variable	Used to indicate variables or text that should be replaced with an actual value.
menu option > menu option >	Used to indicate the cascading menu option to be selected.
	Example: **Operator Functions > Report Functions**
[hypertext link](#)	Used to indicate a hypertext link.

Compliance Statement

Overview

Introduction

This chapter introduces the Oracle Communications Network Charging and Control (NCC) Advanced Control Services (ACS) compliance limitations.

In this chapter

This chapter contains the following topics.

About ACS Compliance

Introduction to ACS Compliance

This document states the compliance of the NCC ACS component against the following INAP, CAP, and MAP protocol specification documents:

- ETSI INAP – *ETS 300 374-1 Intelligent Network Capability Set 1 (CS1); Core Intelligent Network Application Protocol (INAP); Part 1: Protocol specification*
- CS2 INAP – *ITU-T Q1228: Interface Recommendation for intelligent network Capability Set 2: Part 1*
- CAP2 – *ETSI TS 101 046 V7.1.0 CAMEL Application Part (CAP) specification (GSM 09.78 version 7.1.0 Release 1998)*
- CAP3 – *ETSI TS 129 078 V4.8.0 CAMEL Application Part (CAP) specification (3GPP TS 29.078 version 4.8.0 Release 4)*
- CAP4 – *3GPP TS 29.078 V11.2.0 (CAMEL) Phase 4; CAMEL Application Part (CAP) specification (Release 11)*
- MAP – *GPP TS 29.002 V7.2.0 Mobile Application Part (MAP) specification; (Release 7)*

This document states compliance only for the protocols listed above when used by the ACS component to handle voice calls. ACS uses all the protocols, except the MAP protocol, to communicate with Service Switching Points (SSPs) or Specialized Resource Functions (SRFs).

ACS uses the MAP protocol for retrieving location information. This document covers the use of only the MAP AnyTimeInterrogation operation that is sent by the slee_acs binary when charging for a voice call. See *Real Time Charge, Location Information Retrieval Message Sequence* (on page 13) for more information.

ACS also handles other protocols by using plugin shared libraries or interfaces that do protocol conversion. Many of these additional pieces of software are part of the NCC product. Protocols that are handled in this way include DIAMETER, MAP (USSD parts), MAP (SMS parts), SMPP, EMI, CAP3/GPRS and SIP. You can also write custom plugins and interfaces for handling protocols by using the NCC Software Developer's Kit (SDK), see *SDK Developer's Guide* for more information. The ACS Protocol Implementation Conformance Statement guide does not cover the use of these additional protocols.

ACS is not fully compatible with CS2, although some CS2 operations are implemented. Some SSPs implement a few CS2 operations and ACS is able to use these operations to provide additional functionality to the functionality that is available with CS1; for example, see *Post Answer Beep or Announcement Message Sequence* (on page 9).

Note: INAP is used for internal communication between ACS and various interfaces which perform protocol conversion. This use of INAP is not covered by this document. In addition, this document describes ACS compliance for the INAP or CAP protocol layer only, with some mention of the TCAP protocol layer. It does not describe ACS compliance with lower layers of the stack.

Relationship of ACS to INAP, CAP, and MAP Interfaces Diagram

The following diagram shows the relationship between ACS and the INAP, CAP, and MAP interfaces, (also known as layers) covered in this document.

Note: The INAP, CAP, MAP, and TCAP layers in this diagram are the only layers described by this document. This document does not describe ACS compliance with the SCCP and M3UA layers.
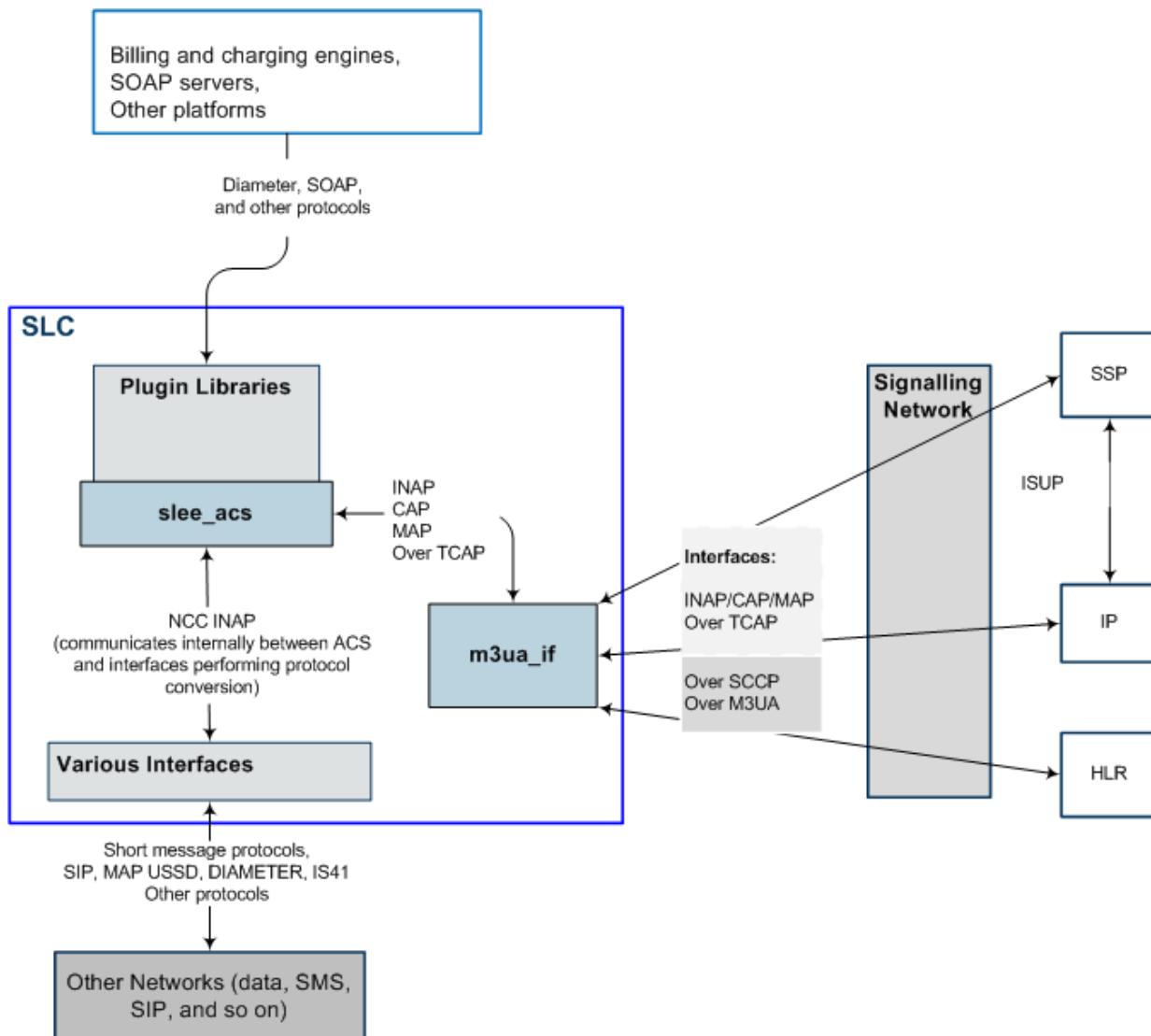
Example ACS Message Sequences

Overview

Introduction

This chapter describes example message sequences that can be sent between Oracle Communications Network Charging and Control (NCC) Advanced Control Services (ACS) and Service Switching Points (SSPs) or Specialized Resource Functions (SRFs) by using the INAP CS1, INAP CS2, CAP2, CAP3, CAP4, and MAP protocols.

In this chapter

This chapter contains the following topics.

Message Sequence Examples

Introduction to Message Sequence Examples

This section provides the following example message sequences used by ACS. The message sequences are ordered by level of complexity (from less complex to more complex) and by type of message sequence:

- *Unconditional Call Connection Message Sequence* (on page 4)
- *Release Call Message Sequence* (on page 4)
- *Divert on Busy Message Sequence* (on page 4)
- *CallInformationReport Message Sequence* (on page 6)
- *FurnishChargingInformation Message Sequence* (on page 6)
- *SendChargingInformation Message Sequence* (on page 7)
- *Play an Announcement (Internal SRF) Message Sequence* (on page 7)
- *Prompt for Digits Using External Intelligent Peripheral SRF Message Sequence* (on page 8)
- *Post Answer Beep or Announcement Message Sequence* (on page 9)
- *Real Time Charging, Caller Hangs Up Message Sequence* (on page 11) (CAP2, CAP3, or CAP4)
- *Real Time Charging, Funds Exhausted Message Sequence* (on page 12)
- *Real Time Charge, Disconnect B-party When Funds Exhausted Message Sequence* (on page 12) (CAP4)
- *Real Time Charge, Location Information Retrieval Message Sequence* (on page 13)
- *Call Initiation (CS1) Message Sequence* (on page 14)
- *Call Initiation (CAP4) Message Sequence* (on page 15)
- *CollectInformation Message Sequence* (on page 17)
- *Combined Message Sequence Example* (on page 17)

Unconditional Call Connection Message Sequence

ACS supports message sequences for unconditionally connecting a call, as shown in the following figure. The yellow box on the left represents the Unconditional Termination feature node. You use an Unconditional Termination feature node in the control plan to set the connection destination:
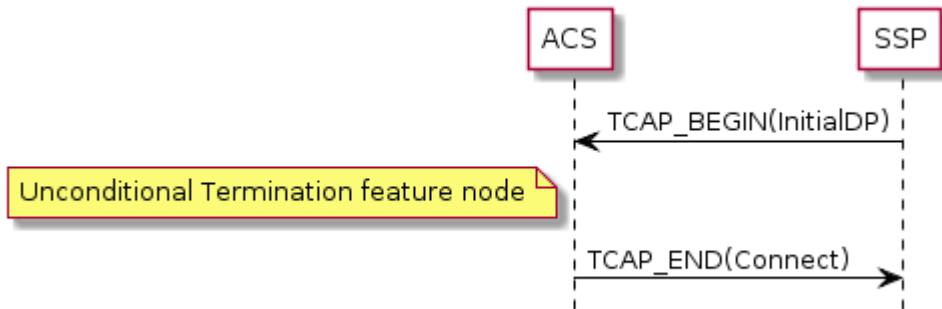
Release Call Message Sequence

ACS supports message sequences for releasing a call, as shown in the following figure. The yellow box on the left represents the Disconnect feature node. You use a Disconnect feature node in the control plan to release the call:
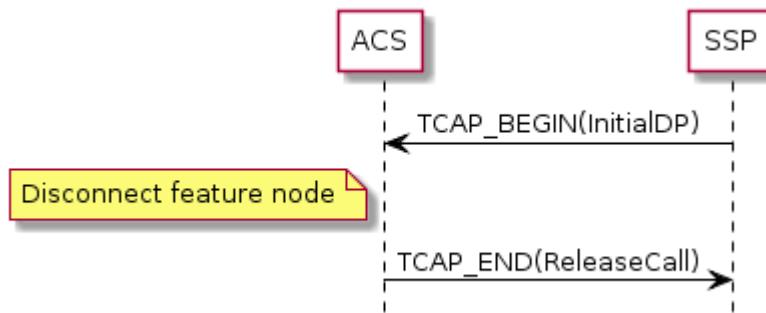
Divert on Busy Message Sequence

ACS supports message sequences for diverting a call when a busy response is received, as shown in the following figure. The yellow boxes on the left represent Attempt Termination feature nodes. You use Attempt Termination feature nodes in the control plan to set the connection destination, and divert on busy destination.
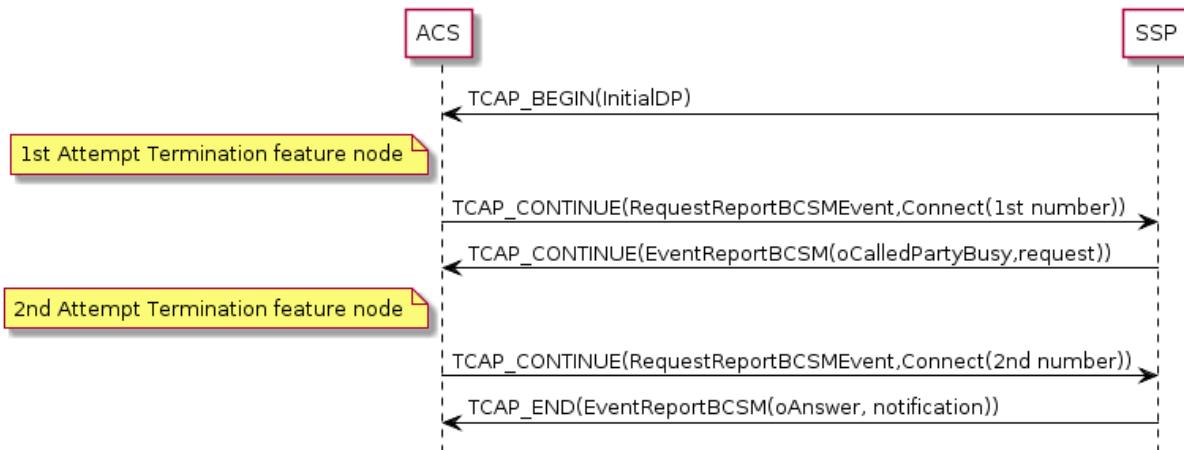
Divert on Busy RequestReportBCSMEvent Table

The table below lists eventTypeBCSM detection points for the RequestReportBCSMEvent operation, and shows how each event is armed in divert on busy message sequences.

Notes:

- If the InitialDP.eventTypeBCSM detection point indicates the terminating basic call model, then the originating event detection points in the table below are replaced with the corresponding terminating event detection points.

- If the EventReportBCSMEvent(oAnswer, notification) is sent in a TCAP_CONTINUE message response, then ACS assumes a pre-arranged end of the transaction.

eventTypeBCSM	monitorMode	legID	dpSpecificCriteria
oAnswer	notifyAndContinue	2	Not set
oCalledPartyBusy	interrupted	2	Not set
oAnswer	interrupted	2	Sets the applicationTimer parameter to the time in seconds that the SSP should wait before reporting noAnswer. You can configure ACS so that this parameter is not set. When applicationTimer is not set, the timeout is network specific.
oAbandon	notifyAndContinue	1	Not set
routeSelectFailure	interrupted	2	Not set

CallInformationReport Message Sequence

The following figure shows an example message sequence where ACS uses the CallInformationReport operation to collect information about the call to put in the call data record (CDR). Apart from call information reporting, the message sequence is the same as for Divert on Busy, and the same event detection points are armed. For details, see *Divert on Busy RequestReportBCSMEvent Table* (on page 5).
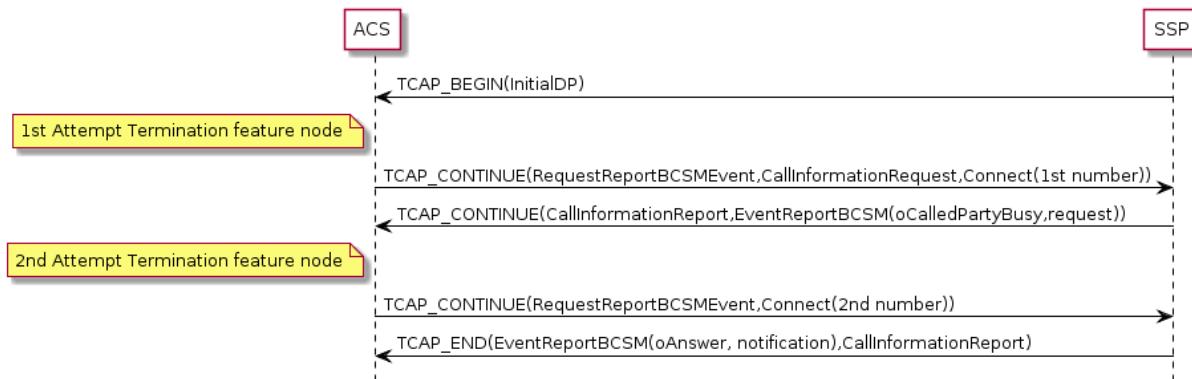
ACS supports receiving separate TCAP messages for CallInformationReport and EventReportBCSM operations. The SSP sends:

- CallInformationReport before EventReportBCSM when the event is armed as **interrupted**
- EventReportBCSM before CallInformationReport when the event is armed as **notifyAndContinue**

Some SSPs will not send CallInformationReport operations for **oCalledPartyBusy**, **oNoAnswer** and **routeSelectFailure**. You can configure ACS to handle this situation.

Note: You can include the CallInformationRequest and CallInformationReport operations in unconditional call connection message sequences by using a similar scenario.

FurnishChargingInformation Message Sequence

The following figure shows an example message sequence where ACS sends a FurnishChargingInformation operation before each Connect in order to provide the SSP with information to put in its call data record.

Notes:

- You can also configure ACS to send FurnishChargingInformation before ReleaseCall, ConnectToResource, and EstablishtemporaryConnection operations.

- When the MTP protocol is used to convey the INAP operations, each MTP packet is limited to 272 octets in length. A FurnishChargingInformation operation can sometimes exceed this limit; for example, if it has been combined with RequestReportBCSMEvent and Connect operations. You can configure ACS to always send FurnishChargingInformation in a separate TCAP_CONTINUE message to avoid this problem.
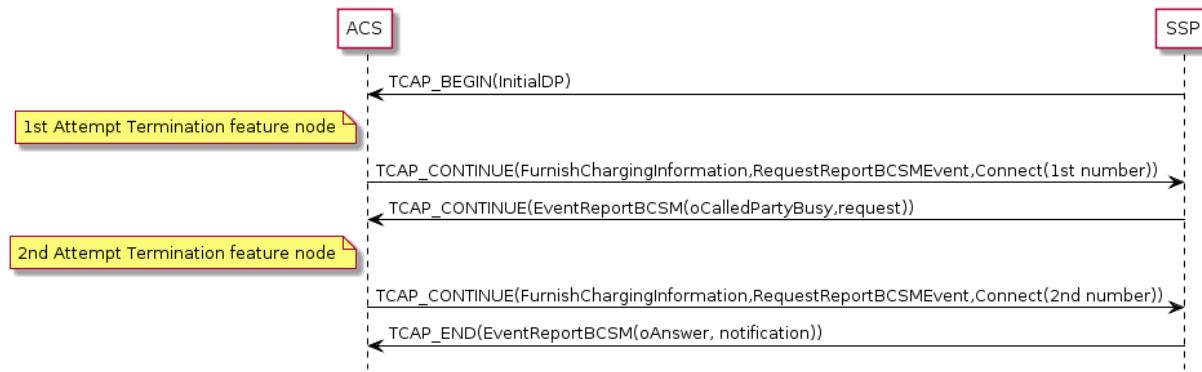
SendChargingInformation Message Sequence

The following figure shows an example message sequence where ACS sends a SendChargingInformation operation to the SSP before each Connect. The SendChargingInformation operation influences the way that the SSP charges for the call.

Note: You can also configure ACS to send SendChargingInformation before ReleaseCall, ConnectToResource, and EstablishtemporaryConnection operations.
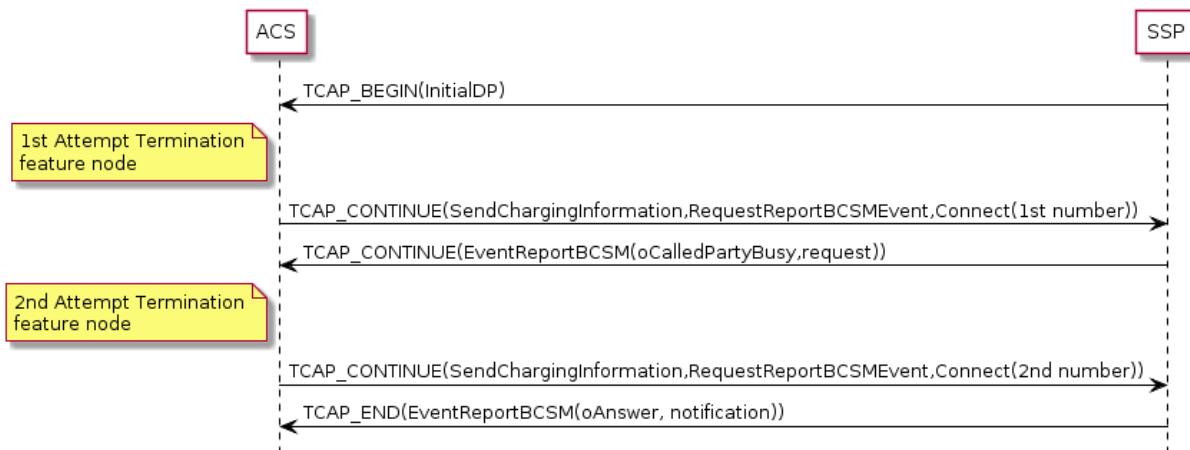
Play an Announcement (Internal SRF) Message Sequence

ACS supports message sequences for playing announcements. The example message sequence in the following figure plays two announcements to the caller before connecting the call. You configure the announcements to play in Play Announcement feature nodes in the control plan. You connect the call by using an Unconditional Termination feature node.

Note: The example uses an internal SRF as the SSP.
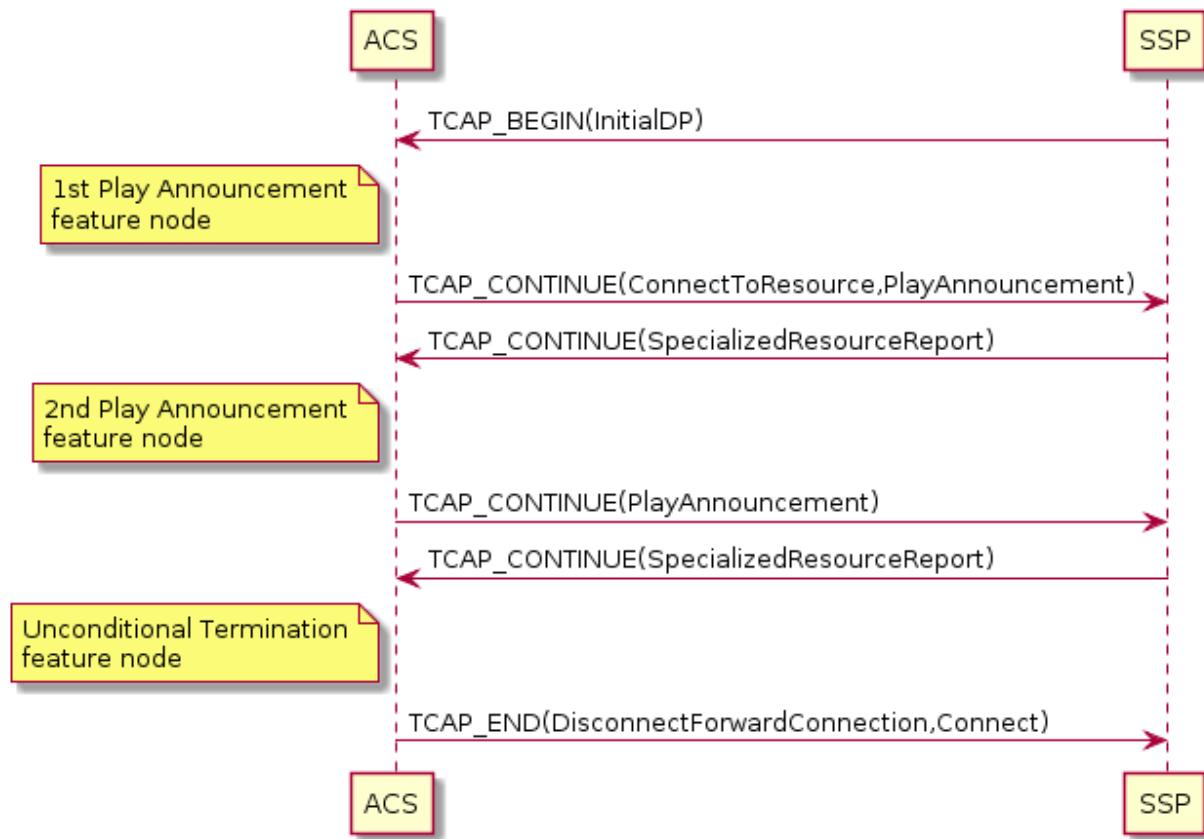
Prompt for Digits Using External Intelligent Peripheral SRF Message Sequence

ACS supports message sequences using external intelligent peripherals to issue prompts. In the following example message sequence ACS prompts the caller for a digit, using an external intelligent peripheral, and repeats the prompt after the caller enters an invalid option. Then, using a different external intelligent peripheral, ACS prompts the caller for the phone number to connect to.
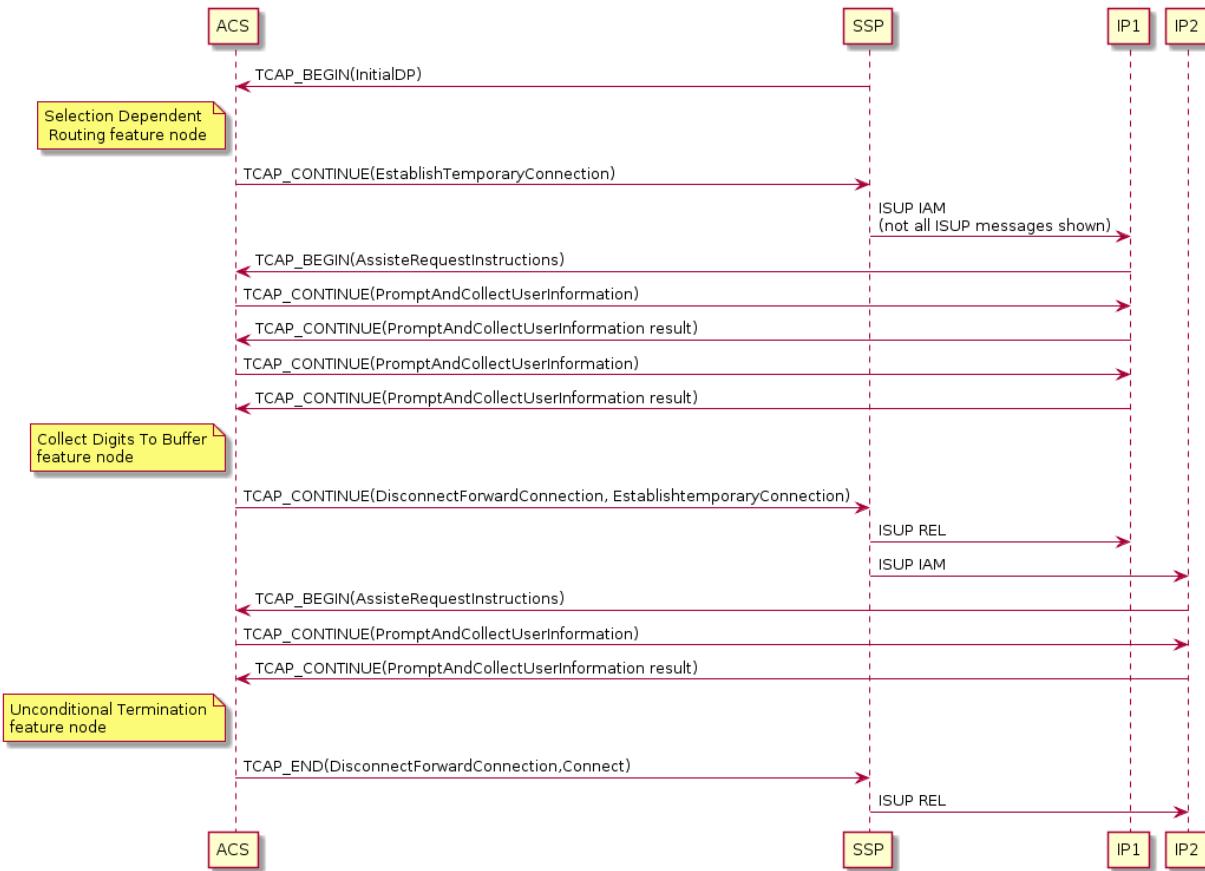
Post Answer Beep or Announcement Message Sequence

The following scenario uses some CS2 operations to provide a short announcement, such as a beep, to the B-party immediately after the call is answered; for example, to warn the B-party that they will be required to pay for the incoming call. In order to play the beep announcement, legs 1 and 2 (the A and B legs respectively) are temporarily split apart while leg 2 connects to the internal SRF in the SSP.

Note: This scenario uses a valid message sequence; however, only a minority of SSPs will support this message sequence.

Post Answer Announcement Example

This example shows the message sequence for a post answer announcement where:

- The leg 1 event detection points are armed in a separate RequestReportBCSMEvent operation from the leg 2 event detection points.
- All operations after the SplitLeg operation are sent in separate TCAP messages to ensure that all SSPs will be able to complete the message sequence correctly.

You configure the beep announcement in an Unconditional Termination feature node in the control plan.

See *RequestReportBCSMEvent Table* (on page 10) for details of how the events in the RequestReportBSCMEvent operation are armed and disarmed.

ACS   SSP

TCAP_BEGIN(InitialDP)

Unconditional Termination feature node, warning beep configured for service

TCAP_CONTINUE(RequestReportBCSMEvent, RequestReportBCSMEvent,Connect)

TCAP_CONTINUE(EventReportBCSM(oAnswer,request))

TCAP_CONTINUE(SplitLeg(leg 2))

TCAP_CONTINUE(SplitLeg result)

TCAP_CONTINUE(ConnectToResource(Leg2))

TCAP_CONTINUE(PlayAnnouncement(Leg2))

TCAP_CONTINUE(SpecializedResourceReport)

TCAP_CONTINUE(DisconnectForwardConnectionWithArgument(leg 2))

TCAP_CONTINUE(MergeCallSegments)

TCAP_CONTINUE(MergeCallSegments result)

TCAP_CONTINUE(RequestReportBCSMEvent)

TCAP_CONTINUE(RequestReportBCSMEvent)

TCAP_END(ContinueWithArgument(leg 1))

ACS   SSP

RequestReportBCSMEvent Table

This table lists the RequestReportBCSMEvent operations from the *Post Answer Announcement Example* (on page 9) in operation order, and provides details of how each event for the RequestReportBCSMEvent operation is armed.

Operation	eventTypeBCSM	monitorMode	legID	dpSpecificCriteria
First RequestReportBCSMEvent operation	oDisconnect	notifyAndContinue	1	Not set
Second RequestReportBCSMEvent operation	oAnswer oDisconnect	interrupted notifyAndContinue	2 2	Not set Not set

This table provides details of how each event for the RequestReportBCSMEvent operation is disarmed, in operation order.

Operation	eventTypeBCSM	monitorMode	legID	dpSpecificCriteria
Third RequestReportBCSMEvent operation	oDisconnect	transparent	1	Not set

Operation	eventTypeBCSM	monitorMode	legID	dpSpecificCriteria
Second RequestReportBCSMEvent operation	`oDisconnect`	transparent	2	Not set

Real Time Charging, Caller Hangs Up Message Sequence

ACS supports the CAP2, CAP3, and CAP4 message sequences used for charged calls, where the caller hangs up, as shown in the following figure:

Note: ACS supports sending the EventReportBCSM operation and final ApplyChargingReport operation in one TCAP_CONTINUE message, or as two separate TCAP messages.
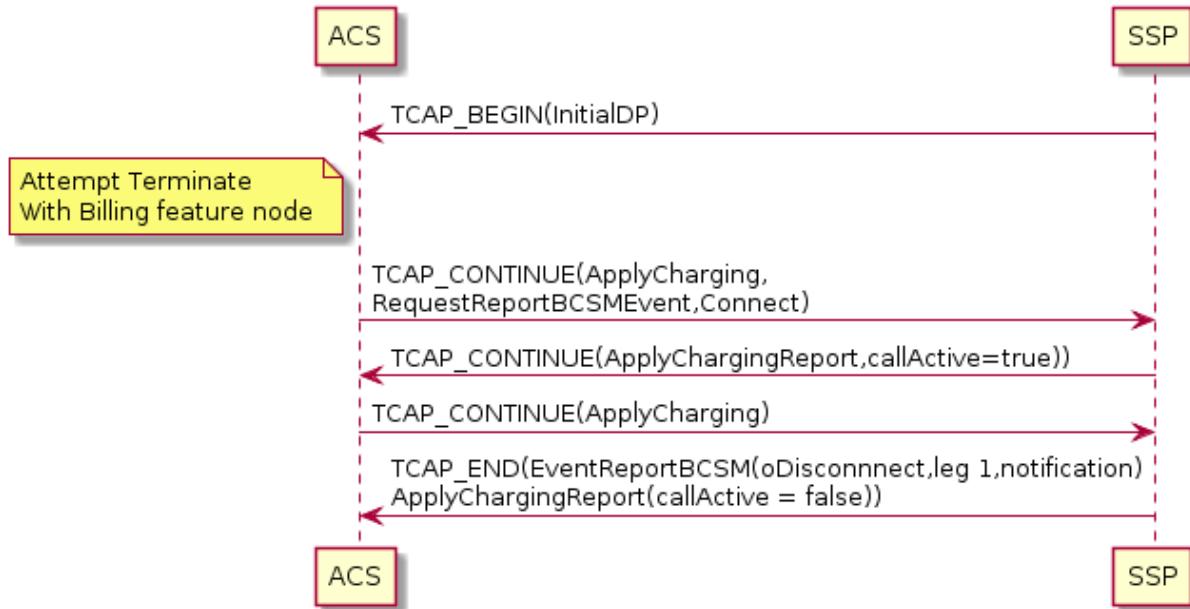
Caller Hangs Up RequestReportBCSMEvent Table

The following table provides details of how each event is armed for the RequestReportBCSMEvent operation in the example message sequence for a charged call where the caller hangs up.

eventTypeBCSM	monitorMode	legID	dpSpecificCriteria
`oCalledPartyBusy`	interrupted	2	Not set
`oAnswer`	interrupted	2	Sets the `applicationTimer` parameter to the time in seconds that the SSP should wait before reporting `noAnswer`. You can configure ACS so that this parameter is not set. When `applicationTimer` is not set, the timeout is network specific.
`oAbandon`	notifyAndContinue	1	Not set
`routeSelectFailure`	interrupted	2	Not set
`oDisconnect`	notifyAndContinue	1	Not set

eventTypeBCSM	monitorMode	legID	dpSpecificCriteria
oDisconnect	interrupted	2	Not set

Real Time Charging, Funds Exhausted Message Sequence

The following figure shows the message sequence used by ACS for real time charging when the caller's funds are exhausted. Note that the second ApplyCharging operation sets the `releaseIfdurationExceeded` parameter.

The `callReleaseAtTcpExiry` parameter that is available for CAP3, does not exist in CAP2. Therefore, if the B-party (leg 2) hangs up, the message sequence for CAP2 will be different from the example message sequence in the following ways:

- The callReleaseAtTcpExpiry operation will be absent
- An EventReportBCSM(oDisconnect,leg 2, request) will be received in another message from the SSP

This means that for CAP2, ACS does not know whether the timer has expired (because this is the pre-arranged end of the message sequence) or whether the B-party has hung up, and ACS should wait for the EventReportBCSM. You can configure ACS to recognize the difference between B-party hang-up and timer expiry for CAP2 in one of the following ways:

- By assuming that when the B-party hangs up, the EventReportBCSM(oDisconnect,leg 2, request) is always in the same TCAP message as the ApplyChargingReport(callActive=false)
- By assuming that the message sequence always ends with an empty TCAP_END
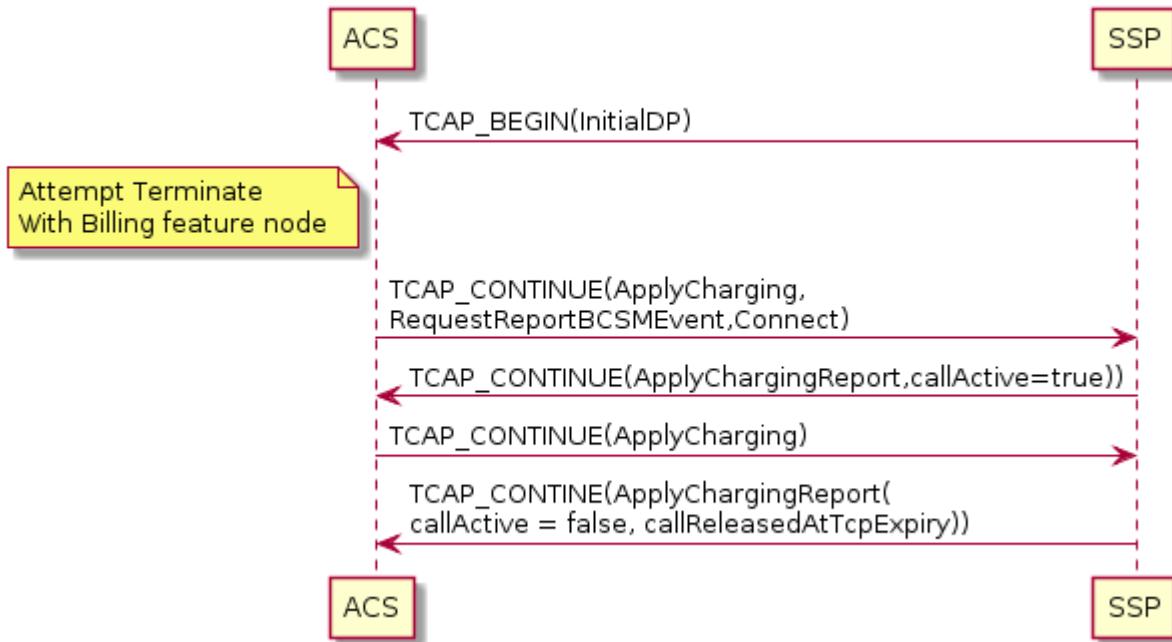
Real Time Charge, Disconnect B-party When Funds Exhausted Message Sequence

The following figure shows the message sequence for disconnecting only the B-party (leg 2) after the funds expire, so that ACS can play an announcement to the caller giving the cost of the call.

Note: In this message sequence the `releaseIfdurationExceeded` parameter is not specified in the ApplyCharging operation.
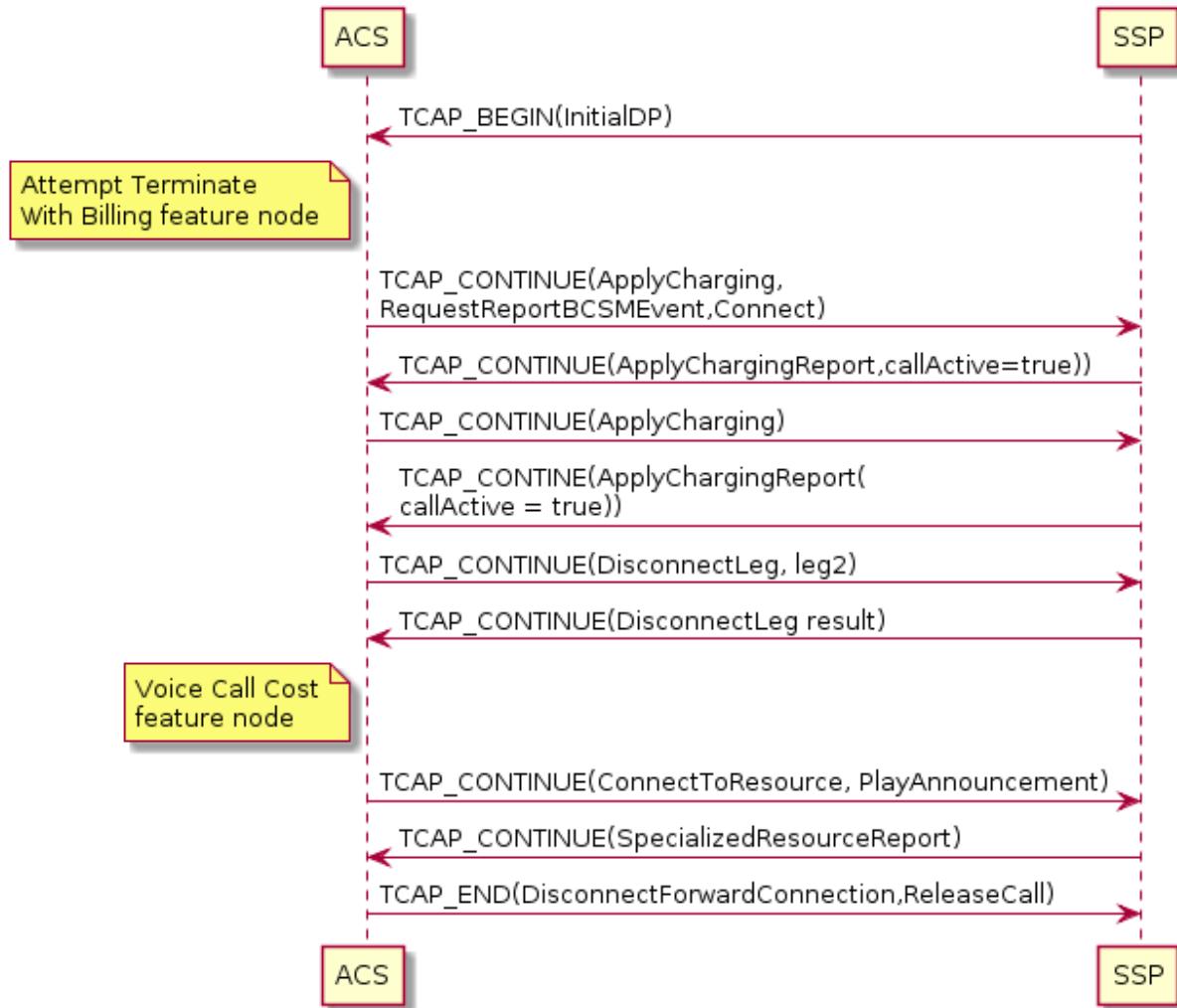
Real Time Charge, Location Information Retrieval Message Sequence

The following figure shows the message sequence ACS uses for retrieving the location of the caller. In the example message sequence, ACS is configured to use the MAP `AnyTimeInterrogation` parameter to retrieve the caller's location when it receives an ApplyChargingReport operation.

Note: The location information can affect the charging applied.
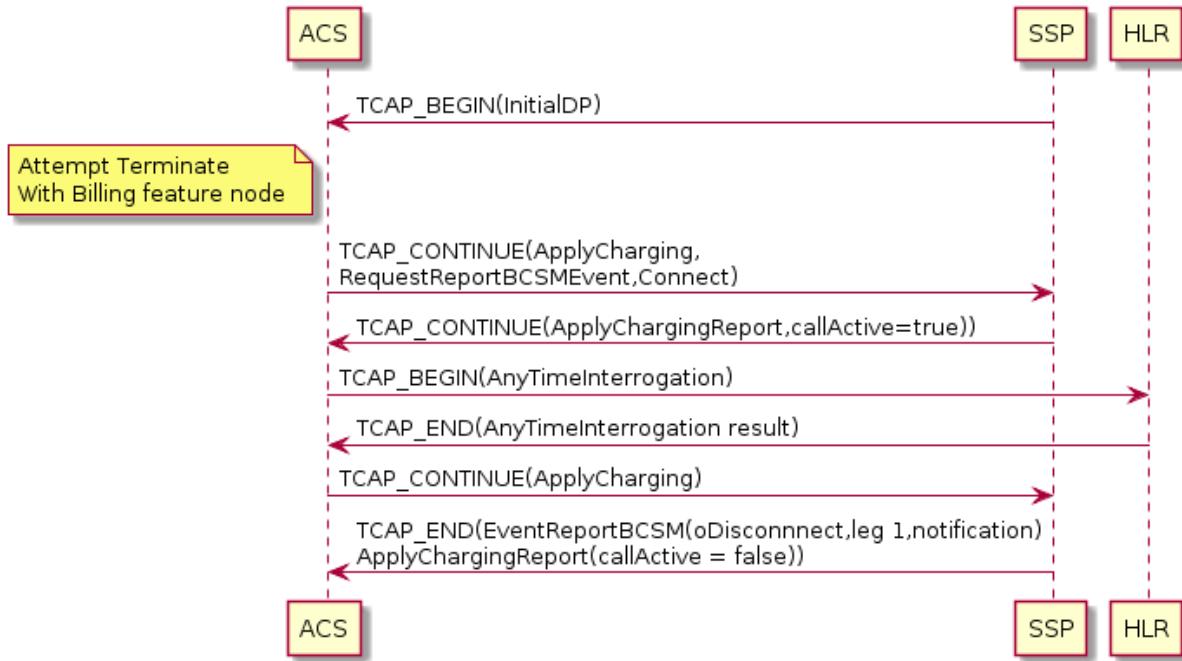
Call Initiation (CS1) Message Sequence

ACS supports call initiation message sequences that use INAP CS1. In the message sequence in the following figure, the USSD Gateway receives a MAP USSD operation (not shown), and then sends ACS an internal InitialDP to initiate a new call to the A-party. In the message sequence, the A-party (leg 1) is connected to another number, the B-party (leg 2), with no real time charging. You use a Call Initiation feature node in the control plan to initiate the call.
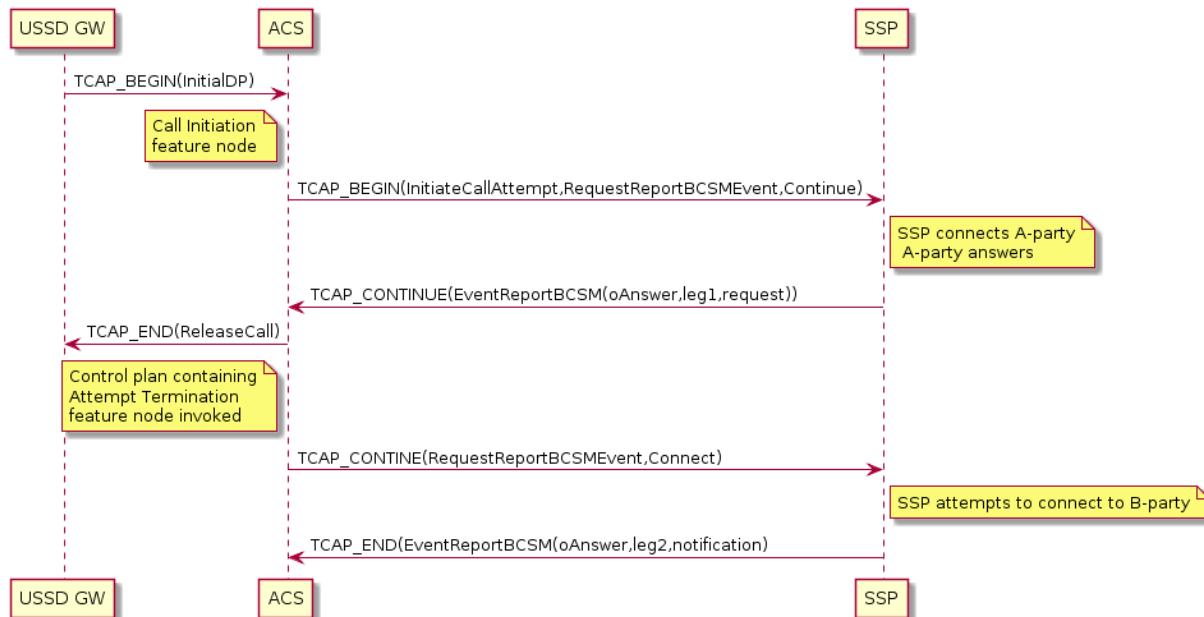
Initiate Call (CS1) RequestReportBCSMEvent Table

The following table provides details of how each event is armed for the first RequestReportBCSMEvent operation in the call initiation (CS1) message sequence example.

Note: For details of how the events are armed in the second RequestReportBCSMEvent operation, see *Divert on Busy RequestReportBCSMEvent Table* (on page 5).

eventTypeBCSM	monitorMode	legID	dpSpecificCriteria
oAnswer	interrupted	1	Not set
oNoAnswer	notifyAndContinue	1	Sets the applicationTimer parameter to the time in seconds that the SSP should wait before reporting noAnswer. You can configure ACS so that this parameter is not set. When applicationTimer is not set, the timeout is network specific.
routeSelectFailure	notifyAndContinue	1	Not set
oCalledPartyBusy	notifyAndContinue	1	Not set

Call Initiation (CAP4) Message Sequence

ACS supports call initiation message sequences that use CAP4. In the message sequence in the following figure, the USSD Gateway receives a MAP USSD operation (not shown), and then sends ACS an internal InitialDP to initiate a new call to the A-party (leg 1). In this example, the A-party is connected to another number, the B-party (leg 2), with real time charging. You use a Call Initiation feature node in the control plan to initiate the call.

Note: Although not recommended, you can configure ACS to use leg 3 instead of leg 1 for a call initiation (CAP4) message sequence. This is because the CAP4 specification explicitly states that leg 3 or higher must be used for InitiateCallAttempt operations.
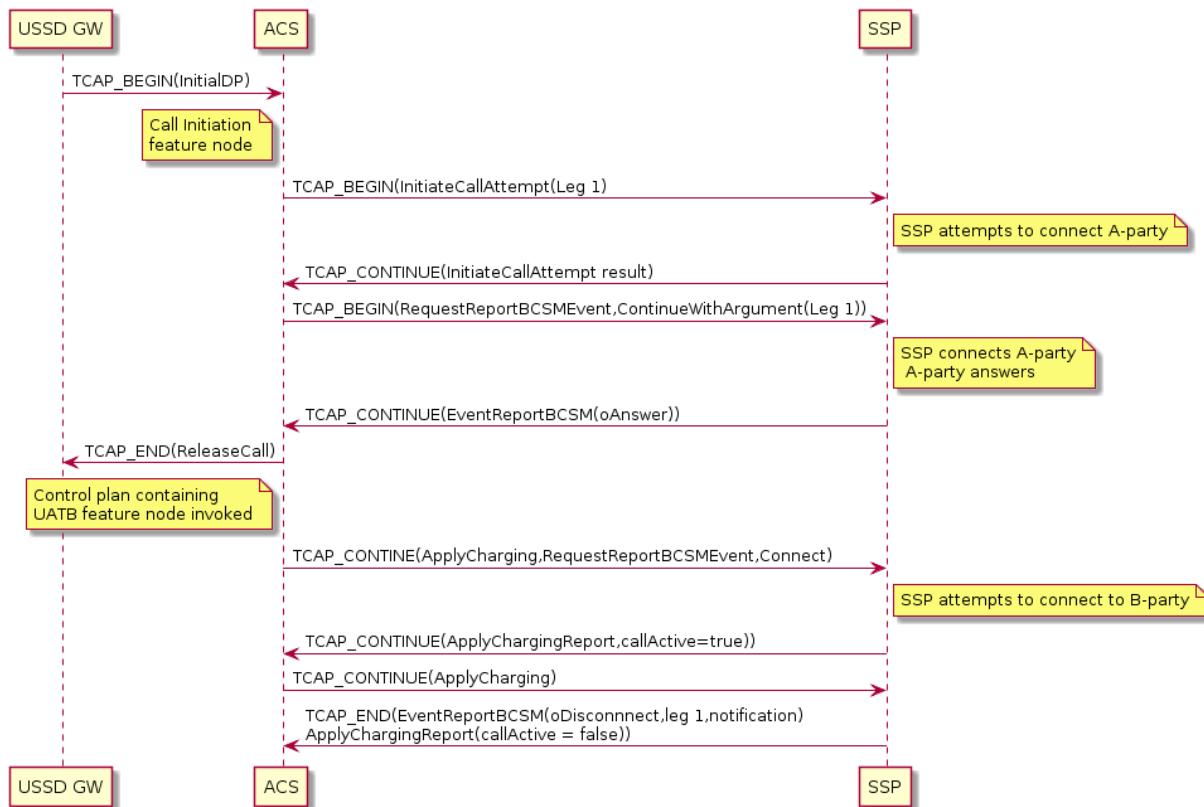
Initiate Call (CAP4) RequestReportBCSMEvent Table

The following table provides details of how each event is armed for the first RequestReportBCSMEvent operation in the call initiation (CAP4) message sequence example.

Note: For details of how the events are armed in the second RequestReportBCSMEvent operation, see *Divert on Busy RequestReportBCSMEvent Table* (on page 5).

eventTypeBCSM	monitorMode	legID	dpSpecificCriteria
oAnswer	interrupted	1	Not set
oNoAnswer	notifyAndContinue	1	Sets the `applicationTimer` parameter to the time in seconds that the SSP should wait before reporting `noAnswer`. You can configure ACS so that this parameter is not set. When `applicationTimer` is not set, the timeout is network specific.
routeSelectFailure	notifyAndContinue	1	Not set
oCalledPartyBusy	notifyAndContinue	1	Not set

CollectInformation Message Sequence

In the message sequence in the following figure, a service running on top of ACS, such as the NCC Virtual Private Network (VPN) service, determines that more digits are needed from the SSP before the correct action for the call can be taken. For more information about VPN, see *VPN Technical Guide*.
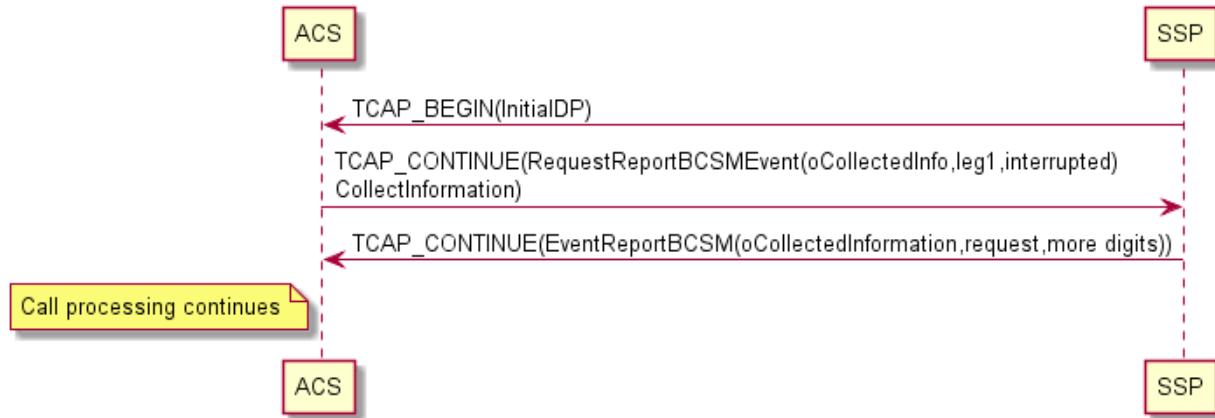
Combined Message Sequence Example

The message sequences in the following figures, when combined, show how several of the features included in the previous message sequences described in this section can be combined.

Diagram 1

```
         ACS                                SSP   IP1   IP2   HLR
          |                                  |     |     |     |
          |<---- TCAP_BEGIN(InitialDP) ------|     |     |     |
  ┌────────────────┐                         |     |     |     |
  │Play Announcement│                        |     |     |     |
  │feature node     │                        |     |     |     |
  └────────────────┘                         |     |     |     |
          |                                  |     |     |     |
          |---- TCAP_CONTINUE( ------------->|     |     |     |
          |     FurnishChargingInformation)  |     |     |     |
          |---- TCAP_CONTINUE( ------------->|     |     |     |
          |     ConnectToResource,PlayAnnouncement)  |     |     |
          |<--- TCAP_CONTINUE(SpecializedResourceReport) |   |
  ┌────────────────┐                         |     |     |     |
  │Selection Dependent│                      |     |     |     |
  │Routing Node       │                      |     |     |     |
  └────────────────┘                         |     |     |     |
          |---- TCAP_CONTINUE( ------------->|     |     |     |
          |     PromptAndCollectuserInformation)     |     |     |
          |<--- TCAP_CONTINUE( --------------|     |     |     |
          |     PromptAndCollectuserInformation      |     |     |
          |     result)                      |     |     |     |
  ┌──────────┐                               |     |     |     |
  │UATB      │                               |     |     |     |
  │feature node│                             |     |     |     |
  └──────────┘                               |     |     |     |
          |--- TCAP_CONTINUE(DisconnectForwardConnection) |   |
          |--- TCAP_CONTINUE(FurnishChargingInformation)  |   |
          |--- TCAP_CONTINUE(ApplyCharging, -->|    |     |     |
          |    RequestReportBCSMEvent,Connect) |    |     |     |
          |<-- TCAP_CONTINUE( ----------------|    |     |     |
          |    ApplyChargingReport,callActive=true)) |   |     |
          |--- TCAP_BEGIN(AnyTimeInterrogation) --------->|     |
          |<-- TCAP_END(AnyTimeInterrogation result) -----|     |
          |--- TCAP_CONTINUE(ApplyCharging) -->|    |     |     |
          |<-- TCAP_CONTINE(ApplyChargingReport( ---|    |     |
          |    callActive = true))            |     |     |     |
          |--- TCAP_CONTINUE( --------------->|     |     |     |
          |    DisconnectLeg, leg2)           |     |     |     |
          |<-- TCAP_CONTINUE( ----------------|     |     |     |
          |    DisconnectLeg result)          |     |     |     |
  ┌──────────┐                               |     |     |     |
  │(Continued)│                              |     |     |     |
  └──────────┘                               |     |     |     |
         ACS                                SSP   IP1   IP2   HLR
```

Diagram 2

Operations Supported by ACS

Overview

Introduction

This chapter lists the operations supported by the Oracle Communications Network Charging and Control (NCC) Advanced Control Services (ACS), and provides details about the supported parameters for each operation.

In this chapter

This chapter contains the following topics.

Operations and Results Sent from SSF to ACS

About Operations and Results Sent from SSF

This section lists the supported operations that can be sent from the Service Switching Function (SSF) to ACS, and provides details about the supported parameters for each operation. For example operations, see *Example ACS Message Sequences* (on page 3).

The following operations and results are the only operations and results sent from the SSF that can be received by ACS:

- *InitialDP Operation (CS1, CS2, CAP2, CAP3, CAP4)* (on page 22)
- *EventReportBCSM Operation* (on page 29)
- *ApplyChargingReport Operation* (on page 30)
- *CallInformationReport Operation* (on page 31)
- *InitiateCallAttempt Result* (on page 31)
- *Support for DisconnectLeg, SplitLeg and MergeCallegments Operation Results* (on page 31)

Notes:

- ACS replies with TCAP_ABORT if it receives an unsupported operation from the SSF.

- ACS ignores any unsupported parameters in operations sent by the SSF.

InitialDP Operation (CS1, CS2, CAP2, CAP3, CAP4)

This table lists the parameters used in InitialDP operations, and identifies which of these parameters are supported by ACS for which protocols. The parameters are listed in the order they appear in the specification documents.

Parameter	Protocol	Supported	Description
serviceKey	CS1, CS2, CAP2, CAP3, CAP4	Yes	ACS uses this parameter to determine which service and which control plan to run. The control plan can also make decisions based on this parameter and the contents of this parameter can be sent over other interfaces such as DIAMETER or SOAP.
calledPartyNumber	CS1, CS2, CAP2, CAP3, CAP4	Yes	ACS uses this parameter for some services (such as the 0800 service) to determine which control plan to run. ACS also uses this parameter in real time charging for rating the call. The control plan can also make decisions based on this parameter and the contents of this parameter can be sent over other interfaces such as DIAMETER or SOAP.
callingPartyNumber	CS1, CS2, CAP2, CAP3, CAP4	Yes	ACS uses this parameter in real time charging to identify the subscriber and for rating the call. See *About InitialDP Parameters and Real Time Charging* (on page 28). The control plan can also make decisions based on this parameter and the contents of this parameter can be sent over other interfaces such as DIAMETER or SOAP.
callingPartysCategory	CS1, CS2, CAP2, CAP3, CAP4	Yes	The control plan can make decisions based on this parameter and the contents of this parameter can be sent over other interfaces such as DIAMETER or SOAP.
cGEncountered	CS1, CS2, CAP3, CAP4	No	
iPSSPCapabilities	CS1, CS2, CAP2, CAP3, CAP4	No	
iPAvailable	CS1, CS2	No	

Parameter	Protocol	Supported	Description
locationNumber	CS1, CS2, CAP2, CAP3, CAP4	Yes	ACS uses this parameter in real time charging for rating the call. See *About InitialDP Parameters and Real Time Charging* (on page 28). The control plan can also make decisions based on this parameter and the contents of this parameter can be sent over other interfaces such as DIAMETER or SOAP.
locationInformation.ageOfLocationInformation	CAP2, CAP3, CAP4	No	
locationInformation.geographicalInformation	CAP2, CAP3, CAP4	No	
locationInformation.vlr-number	CAP2, CAP3, CAP4	Yes	ACS uses this parameter in real time charging for rating the call. See *About InitialDP Parameters and Real Time Charging* (on page 28). The control plan can also make decisions based on this parameter and the contents of this parameter can be sent over other interfaces such as DIAMETER or SOAP.
locationInformation.locationNumber	CAP2, CAP3, CAP4	Yes	ACS uses this parameter in real time charging for rating the call. See *About InitialDP Parameters and Real Time Charging* (on page 28). The control plan can also make decisions based on this parameter and the contents of this parameter can be sent over other interfaces such as DIAMETER or SOAP.
locationInformation.cellIdOrLAI	CAP2, CAP3, CAP4	Yes	The control plan can make decisions based on this parameter (such as "in the zone" logic) and the contents of this parameter can be sent over other interfaces such as DIAMETER or SOAP.
LocationInformation.locationInformationEPS.E-UTRANCellGlobalIdentity	CAP2, CAP3, CAP4	Yes	The control plan can make decisions based on this parameter.

Parameter	Protocol	Supported	Description
LocationInform ation.location InformationEPS .TrackingAreaI dentity	CAP2, CAP3, CAP4	Yes	The control plan can make decisions based on this parameter.
originalCalled PartyID	CS1, CS2, CAP2, CAP3, CAP4	Yes	ACS uses this parameter in real time charging to identify the subscriber and for rating the call. See *About InitialDP Parameters and Real Time Charging* (on page 28). The control plan can also make decisions based on this parameter and the contents of this parameter can be sent over other interfaces such as DIAMETER or SOAP.
extensions	CS1, CS2, CAP2, CAP3, CAP4	Yes	Up to 10 individual extensions can be extracted, based on configurable rules, into strings of characters. You can use one of these extensions for rating the call. See *About InitialDP Parameters and Real Time Charging* (on page 28). The control plan can also make decisions based on any of these extensions and the content of any of these extensions can be sent over other interfaces such as DIAMETER or SOAP.
highLayerCompa tibility	CS1, CS2, CAP2, CAP3, CAP4	Yes	The control plan can make decisions based on this parameter.
serviceInterac tionIndicators	CS1, CS2	No	
additionalCall ingPartyNumber	CS1, CS2, CAP2, CAP3, CAP4	Yes	ACS uses this parameter in real time charging to identify the subscriber and for rating the call. See *About InitialDP Parameters and Real Time Charging* (on page 28). The control plan can also make decisions based on this parameter and the contents of this parameter can be sent over other interfaces such as DIAMETER or SOAP.
forwardCallInd icators	CS1, CS2	No	
bearerCapabili ty	CS1, CS2, CAP2, CAP3, CAP4	Yes	ACS uses this parameter when determining which control plan and service to run. The control plan can also make decisions based on this parameter and the contents of this parameter can be sent over other interfaces such as DIAMETER or SOAP.

Parameter	Protocol	Supported	Description
eventTypeBCSM	CS1, CS2, CAP2, CAP3, CAP4	Yes	The value of this parameter, particularly whether it is from the originating basic call state model or the terminating basic call state model, is used to determine which event detection points to arm in subsequent RequestReportBCSMEvent operations. The control plan can also make decisions based on this parameter.
redirectingPartyID	CS1, CS2, CAP2, CAP3, CAP4	Yes	ACS uses this parameter in real time charging to identify the subscriber and for rating the call. See *About InitialDP Parameters and Real Time Charging* (on page 28). The control plan can also make decisions based on this parameter and the contents of this parameter can be sent over other interfaces such as DIAMETER or SOAP.
redirectionInformation	CS1, CS2, CAP2, CAP3, CAP4	Yes	The control plan can make decisions based on this parameter. See also redirectionInformation in the Connect operation.
dialledDigits	CS2	No	
callingPartyBusinessGroupID	CS2	No	
callingPartySubaddress	CS2	No	
miscCallInfo	CS2	No	
serviceProfileIdentifier	CS2	No	
terminalType	CS2	No	
triggerType	CS2	No	
cause	CS2, CAP3, CAP4	No	
componentType	CS2	No	
component	CS2	No	
componentCorrelationID	CS2	No	
iSDNAccessRelatedInformation	CS2	No	
iNServiceCompatibilityIndication	CS2	No	
genericNumbers	CS2	No	
serviceInteractionIndicatorsTwo	CS2, CAP3, CAP4	No	

Parameter	Protocol	Supported	Description
forwardGVNS	CS2	No	
createdCallSeg mentAssociatio n	CS2	No	
uSIServiceIndi cator	CS2	No	
uSIInformation	CS2	No	
iMSI	CAP2, CAP3, CAP4	Yes	ACS uses this parameter in real time charging to identify the subscriber and for rating the call. See *About InitialDP Parameters and Real Time Charging* (on page 28). The control plan can also make decisions based on this parameter and the contents of this parameter can be sent over other interfaces such as DIAMETER or SOAP.
subscriberStat e	CAP2, CAP3, CAP4	Yes	The control plan can make decisions based on this parameter.
ext- basicServiceCo de	CAP2, CAP3, CAP4	Yes	The control plan can make decisions based on this parameter.
callReferenceN umber	CAP2, CAP3, CAP4	Yes	The control plan can make decisions based on this parameter and the contents of this parameter can be sent over other interfaces such as DIAMETER or SOAP.
mscAddress	CAP2, CAP3, CAP4	Yes	ACS uses this parameter in real time charging for rating the call. See *About InitialDP Parameters and Real Time Charging* (on page 28). The control plan can also make decisions based on this parameter and the contents of this parameter can be sent over other interfaces such as DIAMETER or SOAP.
calledPartyBCD Number	CAP2, CAP3, CAP4	Yes	ACS uses this parameter, if present, instead of the calledPartyNumber parameter.
timeAndTimezon e	CAP2, CAP3, CAP4	No	
gsm- ForwardingPend ing	CAP2, CAP3, CAP4	Yes	The control plan can make decisions based on this parameter.
InitialDPArgEx tension.naCarr ierInformation	CAP2	No	
InitialDPArgEx tension.gmscAd dres	CAP2, CAP3, CAP4	No	

Parameter	Protocol	Supported	Description
carrier	CAP3, CAP4	No	
cug-Index	CAP3, CAP4	No	
cug-Interlock	CAP3, CAP4	No	
cug-OutgoingAccess	CAP3, CAP4	No	
initialDPArgExtension.forwardingDestinationNumber	CAP4	No	
initialDPArgExtension.ms-Classmark2	CAP4	No	
initialDPArgExtension.iMEI	CAP4	Yes	The control plan can make decisions based on this parameter and the contents of this parameter can be sent over other interfaces such as DIAMETER or SOAP.
initialDPArgExtension.supportedCamelPhases	CAP4	No	
initialDPArgExtension.offeredCamel4Functionalities	CAP4	No	
initialDPArgExtension.bearerCapability2	CAP4	No	
initialDPArgExtension.ext-basicServiceCode2	CAP4	No	
initialDPArgExtension.highLayerCompatibility2	CAP4	No	
initialDPArgExtension.lowLayerCompatibility	CAP4	No	
initialDPArgExtension.lowLayerCompatibility2	CAP4	No	
initialDPArgExtension.enhancedDialledServicesAllowed	CAP4	No	
initialDPArgExtension.uu-Data	CAP4	No	

Parameter	Protocol	Supported	Description
`initialDPArgEx tension.collec tInformationAl lowed`	CAP4	No	
`initialDPArgEx tension.releas eCallArgExtens ionAllowed`	CAP4	No	

About InitialDP Parameters and Real Time Charging

ACS uses the following optional parameters in an initialDP operation to identify and locate the subscriber for real time charging:

- callingPartyNumber
- locationNumber
- locationInformation.vlr-number
- locationInformation.locationNumber
- locationInformation.locationInformationEPS
- originalCalledPartyID
- extensions
- additionalCallingPartyNumber
- redirectingPartyID
- iMSI
- mscAddress

ACS uses the parameters to deduce two numbers:

- The number to use to identify the calling subscriber, and so determine who should pay for the call.
- The number which best describes the location of the subscriber, and so rate the call.

You configure the order in which to search for these parameters in ACS. ACS looks for the parameters in the initialDP in the specified order and uses the first one it finds. For example, you can configure ACS to identify the subscriber from one of the following parameters, in the order listed:

- redirectingPartyID
- originalCalledPartyID
- callingPartyNumber

If the redirectingPartyID does not exist, then ACS tries to use the originalCalledPartyID. If the originalCalled PartyID does not exist, then ACS uses the callingPartyNumber.

EventReportBCSM Operation

The following table lists the parameters used in EventReportBCSM operations, and identifies which parameters are supported by ACS for which protocols. The parameters are listed in the order they appear in the specification documents.

Parameter	Protocol	Supported	Description
eventTypeBCSM	CS1, CS2, CAP2, CAP3, CAP4	Yes	The following event detection points only are supported: • collectedInfo(2) • analyzedInformation(3) • routeSelectFailure(4) • oCalledPartyBusy(5) • oNoAnswer(6) • oAnswer(7) • oMidCall(8) • oDisconnect(9) • oAbandon(10) • termAttemptAuthorized(12) • tCalledPartyBusy(13) • tNoAnswer(14) • tAnswer(15) • tMidCall(16) • tDisconnect(17) • tAbandon(18) • oServiceChange(52) • tServiceChange(53) The control plan can make decisions based on this parameter.
eventSpecificInformationBCSM	CS1, CS2, CAP2, CAP3, CAP4	Yes	eventSpecificInformationBCSM is supported for these event detection points only: • collectedInfo(2) • routeSelectFailure(4) • oCalledPartyBusy(5) • oDisconnect(9) • tCalledPartyBusy(13) • tDisconnect(17) • oServiceChange(52) The control plan can make decisions based on this parameter.
legID	CS1, CS2, CAP2, CAP3, CAP4	Yes	The service logic uses this parameter
miscCallInfo	CS1, CS2, CAP2, CAP3, CAP4	Yes	The service logic uses this parameter

Parameter	Protocol	Supported	Description
extensions	CS1, CS2, CAP2, CAP3, CAP4	No	
bcsmEventCorrelationID	CS2	No	
componentType	CS2	No	
component	CS2	No	
componentCorrelationID	CS2	No	

ApplyChargingReport Operation

The ApplyChargingReport operation is defined in the CS1 and CS2 specifications. The CS1 and CS2 specifications do not define the contents of the ApplyChargingReport operation.

Note: ACS supports some non-standard CS1 implementations of ApplyChargingReport but this is beyond the scope of this compliance document.

The following table lists the parameters used in ApplyChargingReport operations, and identifies which parameters are supported by ACS for which protocols. The parameters are listed in the order they appear in the specification documents.

Parameter	Protocol	Supported	Description
partyToCharge	CAP2, CAP3, CAP4	No	
timeInformation.timeIfNoTariffSwitch	CAP2, CAP3, CAP4	Yes	Used to charge the call for real time charging.
timeInformation.timeIfTariffSwitch.timeSinceTariffSwitch	CAP2, CAP3, CAP4	Yes	ACS uses this parameter, if present, in real time charging to determine the charge for the call. **Note:** Because ACS does not set the tariffSwitchInterval parameter in ApplyCharging operations, this parameter should not be present in ApplyChargingReport operations.
timeInformation.timeIfTariffSwitch.tariffSwitchInterval	CAP2, CAP3, CAP4	No	
callActive	CAP2, CAP3, CAP4	Yes	Used by the service logic
callReleasedAtTcpExpiry	CAP3, CAP4	Yes	Used by the service logic
extensions	CAP3, CAP4	No	
aChChargingAddress	CAP4	No	

CallInformationReport Operation

The following table lists the parameters and protocols used in CallInformationReport operations, and identifies which parameters are supported by ACS for which protocols. The parameters are listed in the order they appear in the specification documents.

Parameter	Protocol	Supported	Description
`requestedInfor mationList.Req uestedInformat ion.requestedI nformationType`	CS1	Yes	ACS records information from this parameter in the ACS CDR.
`requestedInfor mationList.Req uestedInformat ion.requestedI nformationValu e`	CS1	Yes	ACS records information from this parameter in the ACS CDR.
`extensions`	CS1	No	
`legID`	CAP2	No	

InitiateCallAttempt Result

Note: Because ACS ignores the contents of the InitiateCallAttempt results returned for InitiateCallAttempt operations, the service designer must ensure that ACS is configured to send to SSFs only InitiateCallAttempt operations that have the desired functionality. When a result with any contents is received, ACS allows the call to proceed.

This table lists the parameters and protocols used in InitiateCallAttempt results, and identifies which parameters are supported by ACS for which protocols. The parameters are listed in the order they appear in the specification documents.

Parameter	Protocol	Supported	Description
`supportedCamel Phases`	CAP4	No	
`offeredCamel4F unctionalities`	CAP4	No	
`extensions`	CAP4	No	
`releaseCallArg ExtensionAllow ed`	CAP4	No	

Support for DisconnectLeg, SplitLeg and MergeCallegments Operation Results

This table lists operation results sent from SSF that are supported by ACS and that return no parameters.

Result	Protocol	Supported
DisconnectLeg result	CS2, CAP4	Yes

Result	Protocol	Supported
SplitLeg result	CS2	Yes
MergeCallegments result	CS2	Yes

Operations Sent from ACS to SSF

About Operations Sent from ACS to SSF

The following section lists the operations that can be sent from ACS to SSF, and provides details about the supported parameters for each operation. ACS includes only supported parameters in the operations that it sends to SSF, unsupported parameters are not included.

The following operations are the only operations that ACS sends to SSF. Operations that are not in the following list are never sent by ACS:

- *EstablishTemporaryConnection Operation (CS1, CS2, CAP2, CAP3, CAP4)* (on page 33)
- *DisconnectForwardConnection Operation (CS1, CS2, CAP2, CAP3, CAP4)* (on page 34)
- *ConnectToResource Operation (CS1, CS2, CAP2, CAP3, CAP4)* (on page 34)
- *Connect Operation (CS1, CS2, CAP2, CAP3, CAP4)* (on page 35)
- *ReleaseCall Operation (CS1, CS2, CAP2, CAP3, CAP4)* (on page 37)
- *RequestReportBCSMEvent Operation (CS1, CS2, CAP2, CAP3, CAP4)* (on page 37)
- *CollectInformation Operation (CS1, CS2, CAP4)* (on page 39)
- *InitiateCallAttempt Operation (CS1, CS2, CAP4)* (on page 40)
- *FurnishChargingInformation Operation (CS1, CS2, CAP2, CAP3, CAP4)* (on page 41)
- *ApplyCharging Operation (CAP2, CAP3, CAP4)* (on page 41)
- *CallInformationRequest Operation (CS1, CS2, CAP2, CAP3, CAP4)* (on page 43)
- *SendChargingInformation Operation (CS1, CS2, CAP2, CAP3, CAP4)* (on page 43)
- *DisconnectForwardConnectionWithArgument Operation (CS2)* (on page 44)
- *ContinueWithArgument Operation (CS2, CAP4)* (on page 45)
- *DisconnectLeg Operation (CS2, CAP4)* (on page 46)
- *SplitLeg Operation (CS2)* (on page 47)
- *MergeCallSegments Operation (CS2)* (on page 47)

Note: ACS replies with TCAP_ABORT if it receives an unsupported operation from SSF.

For example operations, see *Example ACS Message Sequences* (on page 3).

EstablishTemporaryConnection Operation (CS1, CS2, CAP2, CAP3, CAP4)

The following table lists the parameters and protocols used in EstablishTemporaryConnection operations, and identifies which parameters are supported by ACS for which protocols. The parameters are listed in the order they appear in the specification documents.

Parameter	Protocol	Supported	Notes
assistingSSPIRoutingAddress	CS1, CS2, CAP2, CAP3, CAP4	Yes	You can configure ACS to send either the correlationID and scfID parameters, or to include the correlation ID and SCF ID in the assistingSSPIPRoutingAddress parameter. The digits of the assistingSSPIPRoutingAddress, the digits of the SCF ID, and the number of digits in the correlation ID are all configurable.
correlationID	CS1, CS2, CAP2, CAP3, CAP4	Yes	You can configure ACS to send the correlationID parameter, or to include the correlation ID in the assistingSSPIPRoutingAddress parameter. The number of digits in the correlation ID are configurable.
scfID	CS1, CS2, CAP2, CAP3, CAP4	Yes	You can configure ACS to send the scfID parameter, or to include the SCF ID in the assistingSSPIPRoutingAddress parameter. The digits of the SCF ID are configurable.
extensions	CS1, CS2, CAP2, CAP3, CAP4	No	
serviceInteractionIndicators	CS1, CS2	No	
partyToConnect /callSegmentID	CS2, CAP4	No	ACS does not set this parameter even though there are some differences between CAP4 and CS2 for this parameter.
carrier	CS2, CAP4	No	
serviceInteractionIndicatorsTwo	CS2, CAP2, CAP3, CAP4	No	
na-info	CAP2, CAP3, CAP4	No	
chargeNumber	CAP3, CAP4	No	
originalCalledPartyID	CAP4	No	
callingPartyNumber	CAP4	No	

DisconnectForwardConnection Operation (CS1, CS2, CAP2, CAP3, CAP4)

ACS sends the DisconnectForwardConnection operation to SSF for protocols CS1, CS2, CAP2, CAP3, and CAP4. The DisconnectForwardConnection operation has no parameters.

ConnectToResource Operation (CS1, CS2, CAP2, CAP3, CAP4)

The following table lists the parameters and protocols used in ConnectToResource operations, and identifies which parameters are supported by ACS for which protocols. The parameters are listed in the order they appear in the specification documents.

Parameter	Protocol	Supported	Notes
resourceAddress.ipRoutingAddress	CS1, CS2, CAP2, CAP3, CAP4	Yes	This parameter is configurable per Specialized Resource Function (SRF). You can also configure ACS to set this parameter to none for a particular SRF.
resourceAddress.none	CS1, CS2, CAP2, CAP3, CAP4	Yes	This parameter is configurable per SRF. You can also configure the SRF to set this parameter to none.
extensions	CS1, CS2, CAP2, CAP3, CAP4	Yes	You can configure ACS to set the Nokia specific chargeableUI extension to 0, or to set the ZTE specific bothwayThroughConnectionInd extension to bothwayPathNotRequired. No other extensions are supported in this operation.
serviceInteractionIndicators	CS1, CS2	No	
resourceAddress.legID	CS2	Yes	ACS sets this parameter to **2** (for leg 2) only in post answer beep scenarios. See *Post Answer Beep or Announcement Message Sequence* (on page 9). Otherwise, ACS does not set this parameter.
resourceAddress.ipAddressAndLegID	CS2	No	
resourceAddress.callSegmentID	CS2	No	
resourceAddress.ipAddressAndCallSegment	CS2	No	
serviceInteractionIndicatorsTwo	CS2, CAP2, CAP3, CAP4	No	
callSegmentId	CAP4	No	

Connect Operation (CS1, CS2, CAP2, CAP3, CAP4)

The following table lists the parameters and protocols used in Connect operations, and identifies which parameters and protocols are supported by ACS. The parameters are listed in the order they appear in the specification documents.

Parameter	Protocol	Supported	Notes
destinationRou tingAddress	CS1, CS2, CAP2, CAP3, CAP4	Yes	(Required) Configurable in the control plan. ACS always sets this parameter to a value retrieved from the local database or from an external database via, for example, SOAP.
alertingPatter n	CS1, CS2, CAP2, CAP3, CAP4	No	
correlationID	CS1, CS2	No	
cutAndPaste	CS1, CS2	Yes	Configurable in the control plan. ACS does not set this parameter by default.
originalCalled PartyID	CS1, CS2, CAP2, CAP3, CAP4	Yes	You can set this parameter to be populated from a buffer. The buffer contents can be configured in the control plan or set to a value retrieved from the local database, or from an external database; for example, via SOAP.
routeList	CS1, CS2	No	
scfID	CS1, CS2	No	
extensions	CS1, CS2, CAP2, CAP3, CAP4	No	
serviceInterac tionIndicators	CS1, CS2	No	
callingPartyNu mber	CS1, CS2	Yes	You can set this parameter to be populated from a buffer. The buffer contents can be configured in the control plan or set to a value retrieved from the local database, or from an external database; for example, via SOAP.
callingPartysC ategory	CS1, CS2, CAP2, CAP3, CAP4	No	
redirectingPar tyID	CS1, CS2, CAP2, CAP3, CAP4	Yes	You can set this parameter to be populated from a buffer. The buffer contents can be configured in the control plan or set to a value retrieved from the local database, or from an external database; for example, via SOAP.

Parameter	Protocol	Supported	Notes
redirectionInf ormation	CS1, CS2, CAP2, CAP3, CAP4	Yes	You can configure ACS to set the redirection counter within `redirectionInformation` during control plan execution. ACS copies all the other parts of `redirectionInformation` from the InitialDP.
forwardingCond ition	CS2	No	
iSDNAccessRela tedInformation	CS2	No	
travellingClas sMark	CS2	No	
carrier	CS2, CAP3, CAP4	No	
serviceInterac tionIndicators	CS2	No	
displayInforma tion	CS2	No	
forwardCallInd icators	CS2	No	
genericNumbers	CS2, CAP2, CAP3, CAP4	Yes CS2 is not supported	`callingPartyNumber` is not a parameter in Connect operations for CAP protocols. However, if you configure ACS to send `callingPartyNumber` in Connect operations, ACS adds `genericNumbers` with a content of one additional calling party number for CAP2, CAP3, and CAP4 protocols.
serviceInterac tionIndicators Two	CS2, CAP3, CAP4	No	
iNServiceCompa tibilityRespon se	CS2	No	
forwardGVNS	CS2	No	
backwardGVNS	CS2	No	
chargeNumber	CS2, CAP3, CAP4	No	
callSegmentID	CS2	No	
legToBeCreated	CS2, CAP4	No	
suppressionOfA nnouncemen	CAP2, CAP3, CAP4	Yes	You can configure the control plan to set this parameter.
oCSIApplicable	CAP2, CAP3, CAP4	Yes	You can configure the control plan to set this parameter.
na-Info	CAP2, CAP3, CAP4	No	
cug-Interlock	CAP3, CAP4	No	

Parameter	Protocol	Supported	Notes
cug-OutgoingAccess	CAP3, CAP4	No	
bor-InterrogationRequested	CAP4	No	
suppress-N-CSI	CAP4	No	

ReleaseCall Operation (CS1, CS2, CAP2, CAP3, CAP4)

The following table lists the parameters and protocols used in ReleaseCall operations, and identifies which parameters and protocols are supported by ACS. The parameters are listed in the order they appear in the specification documents.

Parameter	Protocol	Supported	Notes
Cause (CS/1, CAP2,CAP3) initialCallSegment (CS/2) allCallSegments (CAP4)	CS1, CS2, CAP2, CAP3, CAP4	Yes	All the protocol specifications use the same coding of this parameter although the ASN.1 is different. ACS sets cause value to 31 by default (normal call clearing) but you can also specify the cause value in control plan configurations and, for error cases, in configuration files.
associatedCallSegment	CS2	No	
allCallSegments	CS2	No	
allCallSegmentsWithExtension	CAP4	No	

RequestReportBCSMEvent Operation (CS1, CS2, CAP2, CAP3, CAP4)

The following table lists the parameters and protocols used in RequestReportBCSMEvent operations, and identifies which parameters and protocols are supported by ACS. The parameters are listed in the order they appear in the specification documents.

Note: For non-standard Service Switching Points (SSPs) that work only when each RequestReportBCSMEvent operation contains only detection points for a single leg, configure ACS to send two RequestReportBCSMEvent operations in the same TCAP message instead of one. The leg 1 events must be in the first operation.

Parameter	Protocol	Supported	Notes
bcsmEvents.eventTypeBCSM	CS1, CS2, CAP2, CAP3, CAP4	Supported	ACS supports only the following event detection points: • collectedInfo(2) • analyzedInformation(3) • routeSelectFailure(4) • oCalledPartyBusy(5) • oNoAnswer(6), oAnswer(7) • oMidCall(8) • oDisconnect(9) • oAbandon(10) • termAttemptAuthorized(12) • tCalledPartyBusy(13) • tNoAnswer(14) • tAnswer(15) • tMidCall(16) • tDisconnect(17) • tAbandon(18) • oServiceChange(52) • tServiceChange(53) See the scenarios in *Example ACS Message Sequences* (on page 3) for details of which event section point is set in which circumstance.
bcsmEvents.monitorMode	CS1, CS2, CAP2, CAP3, CAP4	Supported	See the scenarios in *Example ACS Message Sequences* (on page 3) for details of how monitorMode is set.
bcsmEvents.dPSpecificCriteria.numberOfDigits	CS1, CS2	Supported	Set only for collectedInfo(2).
bcsmEvents.dPSpecificCriteria.applicationTimer	CS1, CS2, CAP2, CAP3, CAP4	Supported	Set only for oAnswer(7) and tNoAnswer(14). You can configure the timer value or use the configured default value. If you configure ACS to leave this parameter unset, then the no answer timer is determined by the network.
extensions	CS1, CS2, CAP2, CAP3, CAP4	No	
bcsmEventCorrelationID	CS2	No	

Parameter	Protocol	Supported	Notes
`bcsmEvents.aut omaticRearm`	CAP4	No	
`bcsmEvents.dPS pecificCriteri a.midCallContr olInfo`	CS2, CAP4	No	
`bcsmEvents.dPS pecificCriteri a.dpSpecificCr iteriaAlt`	CAP4	No	

CollectInformation Operation (CS1, CS2, CAP4)

ACS sends the CollectInformation operation to ask the SSP to collect more digits. See *CollectInformation Message Sequence* (on page 17) for more information

Except for CS2 protocol, the CollectInformation operation has no defined parameters and therefore ACS does not set any parameters.

The following table lists the parameters and protocols used in CollectInformation operations. The CollectInformation operation parameters are not supported by ACS for any protocol. The parameters are listed in the order they appear in the specification documents.

Parameter	Protocol	Supported	Notes
`extensions`	CS1, CS2, CAP4	No	
`numberingPlan`	CS2	No	
`originalCalled PartyID`	CS2	No	
`travellingClas sMark`	CS2	No	
`callingpartyNu mber`	CS2	No	
`dialledDigits`	CS2	No	
`serviceInterac tionIndicators`	CS2	No	
`iNServiceCompa tibilityRespon se`	CS2	No	
`forwardGVNS`	CS2	No	
`backwardGVNS`	CS2	No	
`serviceInterac tionIndicators Two`	CS2	No	
`callSegmentID`	CS2	No	
`legToBeCreated`	CS2	No	

InitiateCallAttempt Operation (CS1, CS2, CAP4)

Note: The InitiateCallAttempt operation has a return result only for CAP4. It does not have a return result for CS1 or CS2.

ACS uses the InitiateCallAttempt operation only to make a new call. It does not use the InitiateCallAttempt operation to make a new leg for an existing call. Logically, the legToBeCreated parameter should therefore be set to 1. However, the CAP4 specification states that the legToBeCreated parameter should be set to 3 or higher, and does not explicitly state that this does not apply for new calls. (This may be an oversight.) Some CAP4 MSCs may insist on legToBeCreated being set to 3 or higher, therefore you should configure to set legToBeCreated to 3 on a per SSP basis. If legToBeCreated is set to 3, leg 3 will be used instead of leg 1 for all other operations sent by ACS.

The following table lists the parameters and protocols used in InitiateCallAttempt operations, and identifies which parameters and protocols are supported by ACS. The parameters are listed in the order they appear in the specification documents.

Parameter	Protocol	Supported	Notes
destinationRoutingAddress	CS1, CS2, CAP4	Yes	(Required) Configurable in the control plan. ACS always sets this parameter to a value retrieved from the local database or from an external database via, for example, SOAP.
alertingPattern	CS1, CS2	No	
extensions	CS1, CS2, CAP4	Partially supported	For CS1 and CS2 (but not CAP4) ACS uses an NCC extension field to hold the category of the calling party. This can be used internally within an NCC system. No other extensions are supported.
serviceInteractionIndicators	CS1, CS2	No	
callingPartyNumber	CS1, CS2, CAP4	Yes	You can configure the source for calling party number in the control plan. It can be a fixed value or a value retrieved from the local database, or from an external database via, for example, SOAP.
iSDNAccessRelatedInformation	CS2	No	
travellingClassMark	CS2	No	
legToBeCreated	CS2, CAP4	Supported for CAP4 only	Set to either 1 or 3, depending on configuration. ACS always sets this parameter for CAP4.
newCallSegment	CS2, CAP4	Supported for CAP4 only	Set to 1. ACS always sets this parameter for CAP4.
iNServiceCompatibilityResponse	CS2	No	
serviceInteractionIndicatorsTwo	CS2	No	

Parameter	Protocol	Supported	Notes
callReferenceN umber	CAP4	Yes	Set to the SLEE call ID, represented in decimal as a string of ASCII digits.
gsmSCFAddress	CAP4	Yes	The gsmSCFAddress is configured on a per SSP basis. ACS always sets this parameter for CAP4. The gsmSCFAddress is always the same as the global title in the SCCP calling party number.
suppress-T-CSI	CAP4	No	

FurnishChargingInformation Operation (CS1, CS2, CAP2, CAP3, CAP4)

In the CS1 specification, the contents of the FurnishChargingInformation operation are undefined, as follows:

```
FurnishChargingInformationArg ::= FCIBillingChargingCharacteristics

FCIBillingChargingCharacteristics ::= OCTET STRING (SIZE
(minFCIBillingChargingLength ..maxFCIBillingChargingLength))
```

Where the FCIBillingChargingCharacteristics parameter indicates the billing and charging characteristics, and its content is network operator specific.

In the CAP2, CAP3, and CAP4 specifications the contents of FurnishChargingInformation are defined, but ACS does not provide explicit support for these definitions. Instead, you can configure a list of fixed binary Furnish Charging Information (FCI) contents for ACS. The control plan selects one of these FCI contents to put in a FurnishChargingInformation operation. To ensure that the configured binary contents conform to CAP2, CAP3, or CAP4, they must be coded by hand.

You can also use the NCC Software Developer's Kit (SDK) to create a custom plugin library to insert dynamic data, such as called and calling party numbers, into FurnishChargingInformation operations. For more information about creating custom plugins by using the SDK, see *SDK Developer's Guide*.

ACS contains some pre-built plugins to support some non-standard FurnishChargingInformation formats but these are unlikely to be generally useful and are beyond the scope of this document.

ApplyCharging Operation (CAP2, CAP3, CAP4)

Notes:

- ApplyCharging is defined as an operation in the CS1 and CS2 specifications but its contents are undefined. ACS does support some non-standard CS1 implementations of ApplyCharging, however this is beyond the scope of this compliance document.

- For most operations, the encoding of each individual parameter is consistent across all the specifications; for example, a parameter with the same tag that is present in CS1, CS2, CAP2, CAP3, and CAP4 will have the same format for each protocol. This is not the case for ApplyCharging where the encoding of the releaseIfDurationExceeded and tone tags are incompatible between CAP2, CAP3 and CAP4. ACS chooses the correct encoding based on the application context received in the TCAP message that contains the InitialDP operation.

The following table lists the parameters and protocols used in ApplyCharging operations, and identifies which parameters and protocols are supported by ACS. The parameters are listed in the order they appear in the specification documents.

Parameter	Protocol	Supported	Notes
aChBillingChargingCharacteristics.maxCallPeriodDuration	CAP2, CAP3, CAP4	Yes	You can configured ACS to set this parameter in control plans; for example, by using the Attempt Terminate with Pending TN Destination feature node in the control plan.
aChBillingChargingCharacteristics.releaseIfdurationExceeded.tone	CAP2	Yes	You can configured ACS to set this parameter in control plans. When set, a beep is played to the caller for a configurable number of seconds before funds for the call are exhausted.
aChBillingChargingCharacteristics.releaseIfdurationExceeded.extensions	CAP2	No	
aChBillingChargingCharacteristics.tariffSwitchInterval	CAP2, CAP3, CAP4	No	
partyToCharge	CAP2, CAP3, CAP4	Yes	You can configure ACS to always set this to the leg ID of the calling party (which will be leg 1 unless the call has been started with an InitiateCallAttempt for leg 2). By default this parameter is set to leg 1.
extensions	CAP2, CAP3, CAP4	No	
aChBillingChargingCharacteristics.releaseIfdurationExceeded	CAP3,CAP4	Yes	You can configured ACS to set this parameter and aChBillingChargingCharacteristics.tone in control plans. When set, a beep is played to the caller a configurable number of seconds before funds for the call are exhausted.
aChBillingChargingCharacteristics.tone	CAP3	Yes	You can configure ACS to set this parameter and aChBillingChargingCharacteristics.releaseIfdurationExceeded in control plans. When set, a beep is played to the caller a configurable number of seconds before funds for the call are exhausted.
aChBillingChargingCharacteristics.extensions	CAP3, CAP4	No	

Parameter	Protocol	Supported	Notes
`aChChargingAddress`	CAP4	Yes	ACS always sets this parameter to leg 2 for CAP4.
`aChBillingChargingCharacteristics.audibleIndicator`	CAP4	Yes	ACS omits this parameter or sets it to the value of `tone`, depending on configuration. No other values are supported.

CallInformationRequest Operation (CS1, CS2, CAP2, CAP3, CAP4)

This table lists the parameters used in CallInformationRequest operations, and identifies which parameters and protocols are supported by ACS. The parameters are listed in the order that they appear in the specification documents.

Parameter	Protocol	Supported	Notes
`requestedInformationTypeListone`	CS1, CS2, CAP2, CAP3, CAP4	Yes	By default, ACS includes all five requested information types defined in CS1: • callAttemptElapsedTime(0) • callStopTime(1) • callConnectedElapsedTime(2) • calledAddress(3) • releaseCause(30) You can configure ACS to omit one or more of these items from the request. Because CAP2, CAP3, and CAP4 do not support calledAddress, you must configure ACS to omit calledAddress from the request if a CAP protocol is being used.
`extensions`	CS1, CS2, CAP2, CAP3, CAP4	No	
`correlationID`	CS2	No	
`legID`	CS1, CS2, CAP2, CAP3, CAP4	No	

SendChargingInformation Operation (CS1, CS2, CAP2, CAP3, CAP4)

In the CS1 specification, the contents of SendChargingInformation are incompletely defined as follows:

```
SendChargingInformationArg ::= SEQUENCE {
    sCIBillingChargingCharacteristics[0] SCIBillingChargingCharacteristics,
    legID/partyToCharge [1] LegID,
    extensions [2] SEQUENCE SIZE(1..numOfExtensions) OF ExtensionField OPTIONAL
    -- ...
}

SCIBillingChargingCharacteristics ::= OCTET STRING (SIZE
(minSCIBillingChargingLength ..maxSCIBillingChargingLength))
```

Where the `SCIBillingChargingCharacteristics` parameter indicates the billing and charging characteristics, and its content is network operator specific.

In the CAP2, CAP3, and CAP4 specifications the contents of `SCIBillingChargingCharacteristics` are defined, but ACS does not provide explicit support for these definitions. Instead, you can configure a list of fixed binary SCIBillingChargingCharacteristics contents for ACS. The control plan selects one of these SCIBillingChargingCharacteristics contents to put in a SendChargingInformation operation. To ensure that the configured binary contents conform to CAP2, CAP3, or CAP4, they must be coded by hand.

You can also use the SDK to create a custom plugin library to insert dynamic data (such as called and calling party numbers) in to SCIBillingChargingCharacteristics. For more information about creating custom plugins by using the SDK, see *SDK Developer's Guide*.

ACS contains some pre-built plugins to support some non-standard SendChargingInformation formats but these are unlikely to be generally useful and are beyond the scope of this document.

The following table lists the parameters and protocols used in SendChargingInformation operations, and identifies which parameters and protocols are supported by ACS. The parameters are listed in the order they appear in the specification documents.

Parameter	Protocol	Supported	Notes
`sCIBillingChar gingCharacteri stics`	CS1, CS2, CAP2, CAP3, CAP4	Yes	You can configure a list of fixed binary SCIBillingChargingCharacteristics contents for ACS. The control plan selects one of these SCIBillingChargingCharacteristics contents to put in a SendChargingInformation operation. See the descriptive text above for more information.
`legID/partyToC harge`	CS1, CS2, CAP2, CAP3, CAP4	Yes	This parameter is named: • `legID` in CS1 specifications • `partytoCharge` in other specifications The coding of this parameter is the same in all cases. ACS always sets this parameter to the leg ID of the calling party (which will be leg 1 unless the call has been started with an InitiateCallAttempt for leg 3).
`extensions`	CS1, CS2, CAP2, CAP3, CAP4	No	

DisconnectForwardConnectionWithArgument Operation (CS2)

Note: ACS supports only the CS2 version of this operation. It does not support the CAP4 version.

The following table lists the parameters and protocols used in DisconnectForwardConnectionWithArgument operations, and identifies which parameters and protocols are supported by ACS. The parameters are listed in the order they appear in the specification documents.

Parameter	Protocol	Supported	Notes
partyToDisconnect.legID	CS2	Yes	ACS always sets this parameter to leg 2 because this operation is used only by the *Post Answer Beep or Announcement Message Sequence* (on page 9).
partyToDisconnect.callSegmentID / callSegmentID	CS2, CAP4	No	Although the ASN.1 for this parameter differs between CS2 and CAP4, the encoding is the same.
extensions	CS2, CAP4	No	

ContinueWithArgument Operation (CS2, CAP4)

The following table lists the parameters and protocols used in ContinueWithArgument operations, and identifies which parameters and protocols are supported by ACS. The parameters are listed in the order they appear in the specification documents.

Parameter	Protocol	Supported	Notes
legID	CS2	Yes	ACS always sets a leg ID, and uses the application context to work out whether to use this parameter (for CS2) or continueWithArgumentArgExtension.legOrCallSegment.legID (for CAP4).
alertingPattern	CS2, CAP4	No	
genericName	CS2	No	
iNServiceCompatibilityResponse	CS2	No	
forwardGVNS	CS2	No	
backwardGVNS	CS2	No	
extensions	CS2, CAP4	No	
serviceInteractionIndicatorsTwo	CS2, CAP4	No	
callingPartysCategory	CAP4	No	
genericNumbers	CAP4	No	
cug-Interlock	CAP4	No	

Parameter	Protocol	Supported	Notes
cug-OutgoingAccess	CAP4	No	
chargeNumber	CAP4	No	
carrier	CAP4	No	
suppressionOfAnnouncement	CAP4	No	
naOliInfo	CAP4	No	
bor-InterrogationRequested	CAP4	No	
suppress-O-CSI	CAP4	No	
continueWithArgumentArgExtension.suppress-D-CSI	CAP4	No	
continueWithArgumentArgExtension.suppress-N-CSI	CAP4	No	
continueWithArgumentArgExtension.suppressOutgoingCallBarring	CAP4	No	
continueWithArgumentArgExtension.legOrCallSegment.callSegmentID	CAP4	No	
continueWithArgumentArgExtension.legOrCallSegment.legID	CAP4	Yes	This parameter is always set to a leg ID, and not a call segment ID. ACS always sets a leg ID and uses the application context to work out whether to use this parameter (for CAP4) or the legID parameter (for CS2).

DisconnectLeg Operation (CS2, CAP4)

This table lists the parameters and protocols used in DisconnectLeg operations, and identifies which parameters and protocols are supported by ACS. The parameters are listed in the order they appear in the specification documents.

Parameter	Protocol	Supported	Notes
legToBeReleased	CS2, CAP4	Yes	ACS always sets this parameter to leg 2.
releaseCause	CS2, CAP4	Yes	ACS always sets this parameter to 31 for normal call clearing.

Parameter	Protocol	Supported	Notes
extensions	CS2, CAP4	No	

SplitLeg Operation (CS2)

Note: Although the SplitLeg operation is supported in CAP4 and CS2, ACS supports only the CS2 version. ACS uses the SplitLeg operation only in the "Post answer beep or announcement" scenario, and this scenario requires the CS2 MergeCallSegments operation that is No in CAP4.

The following table lists the parameters and protocols used in SplitLeg operations, and identifies which parameters and protocols are supported by ACS. The parameters are listed in the order they appear in the specification documents.

Parameter	Protocol	Supported	Notes
legToBeSplit	CS2	Yes	ACS always sets this parameter to leg 2.
newCallSegment	CS2	No	
extensions	CS2	No	

MergeCallSegments Operation (CS2)

The following table lists the parameters and protocols used in MergeCallSegments operations, and identifies which parameters and protocols are supported by ACS. The parameters are listed in the order they appear in the specification documents.

Parameter	Protocol	Supported	Notes
sourceCallSegment	CS2	Yes	ACS sets this parameter to 2 in post answer beep or announcement message sequences. See *Post Answer Beep or Announcement Message Sequence* (on page 9).
targetCallSegment	CS2	Yes	ACS sets this parameter to 1 in post answer beep or announcement message sequences. See *Post Answer Beep or Announcement Message Sequence* (on page 9).
extensions	CS2	No	

Operations Sent from ACS to an SRF

About Operations Sent from ACS to an SRF

The following section lists the operations that can be sent from ACS to a Specialized Resource Function (SRF), and provides details about the supported parameters for each operation. Unless explicitly stated, the SRF can be an SSF based (internal) SRF or an SRF on an external intelligent peripheral that is connected by using an EstablishTemporaryConnection operation. Operations not listed are not sent by ACS to an SRF.

The following operations are the only operations that ACS sends to an SRF. Operations that are not in the following list are not sent by ACS:

- *PromptAndCollectUserInformation Operation (CS1, CS2, CAP2, CAP3, CAP4)* (on page 50)
- *PromptAndCollectUserInformation Operation (CS1, CS2, CAP2, CAP3, CAP4)* (on page 50)

For example operations, see *Example ACS Message Sequences* (on page 3).

PlayAnnouncement Operation (CS1, CS2, CAP2, CAP3, CAP4)

The following table lists the parameters and protocols used in PlayAnnouncement operations, and identifies which parameters and protocols are supported by ACS. The parameters are listed in the order they appear in the specification documents.

Parameter	Protocol	Yes	Notes
`informationToSend.inbandinfo.messageID.elementaryMessageID`	CS1, CS//2, CAP2, CAP3, CAP4	Yes	You configure announcement name and language pairs to link to elementary message IDs in ACS. You select the announcement names in control plans and ACS maps them to the appropriate elementary message IDs.
`informationToSend.inbandinfo.messageID.text.messageContent`	CS1, CS//2, CAP2, CAP3, CAP4	Yes	You configure to put a Voice XML URL in this parameter in the ACS control plan.
`informationToSend.inbandinfo.messageID.text.attributes`	CS1, CS//2, CAP2, CAP3, CAP4	Yes	ACS sets `textAttributes` to "vxmlURL" when it sets the `messageContent` parameter.
`informationToSend.inbandinfo.messageID.elementaryMessageIDs`	CS1, CS//2, CAP2, CAP3, CAP4	Yes	ACS sets more than one elementary message ID for some functions. You select and map the announcement names to elementary message IDs as described for the `elementaryMessageID` parameter.
`informationToSend.inbandinfo.messageID.variableMessage.elementaryMessageID`	CS1, CS2, CAP2, CAP3, CAP4	Yes	ACS uses variable part announcements for some functions. You select and map the announcement names to variable elementary message IDs as described for the `elementaryMessageID` parameter.
`informationToSend.inbandinfo.messageID.variableMessage.variableParts`	CS1, CS2, CAP2, CAP3, CAP4	Yes	ACS supports all five types of variable part (time, price, date, number, and integer). The contents of these parts can be: • A fixed value • Derived from the database or an external platform such as a billing engine

Parameter	Protocol	Yes	Notes
`informationToS end.inbandinfo .numberOfRepet itions`	CS1, CS2, CAP2, CAP3, CAP4	Yes	Configurable in the control plan
`informationToS end.inbandinfo .duration`	CS1, CS2, CAP2, CAP3, CAP4	Yes	Configurable in the control plan
`informationToS end.inbandinfo .interval`	CS1, CS2, CAP2, CAP3, CAP4	Yes	ACS sets this parameter to zero (0)
`informationToS end.tone.toneI D`	CS1, CS2, CAP2, CAP3, CAP4	No	
`informationToS end.tone.durat ion`	CS1, CS2, CAP2, CAP3, CAP4	No	
`informationToS end.displayInf ormation`	CS1, CS2	No	
`disconnectFrom IPForbidden`	CS1, CS2, CAP2, CAP3, CAP4	Yes	ACS normally does not set this parameter, in which case the SRF should use the default value of TRUE. However, if the type of SRF is "ZTE" then ACS sets this parameter to FALSE.
`requestAnnounc ementComplete`	CS1, CS2, CAP2, CAP3, CAP4	Yes	ACS normally does not set this parameter, in which case the SRF should use the default value of TRUE. However, if the type of SRF is "ZTE" then ACS sets this parameter to TRUE.
`extensions`	CS1, CS2, CAP2, CAP3, CAP4	Yes	You can configure ACS to set the integer language ID in extension type 3 if the type of SRF for the announcement is "NAP" . Support for the extension type includes Unisys Network Application Protocol (NAP) SRFs as well as other SRFs.
`connectedParty .legID`	CS2	Yes	ACS sets this parameter to leg 2 in post answer beep or announcement message sequences. See *Post Answer Beep or Announcement Message Sequence* (on page 9) for more information.
`connectedParty .callSegmentID`	CS2	No	
`requestAnnounc ementStartedNo tification`	CAP4	No	

PromptAndCollectUserInformation Operation (CS1, CS2, CAP2, CAP3, CAP4)

This table lists the parameters and protocols used in PromptAndCollectUserInformation operations, and identifies which parameters and protocols are supported by ACS. The parameters are listed in the order they appear in the specification documents.

Parameter	Protocol	Yes	Notes
`collectedInfo.collectedDigits.minimumNbOfDigits`	CS1, CS2, CAP2, CAP3, CAP4	Yes	ACS always sets this parameter.
`collectedInfo.collectedDigits.maximumNbOfDigits`	CS1, CS2, CAP2, CAP3, CAP4	Yes	ACS always sets this parameter.
`collectedInfo.collectedDigits.endOfReplyDigit`	CS1, CS2, CAP2, CAP3, CAP4	Yes	This parameter is configurable globally. ACS will not set this parameter if the type of SRF is "NORTEL". You can also configure ACS so that it never sets this parameter.
`collectedInfo.collectedDigits.interDigitTimeOut`	CS1, CS2, CAP2, CAP3, CAP4	Yes	This parameter is configurable globally. It may also be overridden in a control plan when collecting a single digit, in which case ACS sets `firstDigitTimeOut` and `interDigitTimeOut` to the same value. ACS will not set this parameter if the type of SRF is "NORTEL". You can also configure ACS so that it never sets this parameter.
`collectedInfo.collectedDigits.errortreatment`	CS1, CS2, CAP2, CAP3, CAP4	No	
`collectedInfo.collectedDigits.interruptableAnnInd`	CS1, CS2, CAP2, CAP3, CAP4	Yes	ACS always sets this parameter to be TRUE always, or to be FALSE always, depending on configuration.
`collectedInfo.collectedDigits.voiceInformation`	CS1, CS2, CAP2, CAP3, CAP4	No	
`collectedInfo.collectedDigits.voiceBack`	CS1, CS2, CAP2, CAP3, CAP4	No	
`collectedInfo.iA5Information`	CS2	No	
`disconnectFromIPForbidden`	CS1, CS2, CAP2, CAP3, CAP4	No	ACS does not set this parameter, and therefore the default of TRUE should be assumed.

Parameter	Protocol	Yes	Notes
`informationToSend.inbandinfo.messageID.elementaryMessageID`	CS1, CS2, CAP2, CAP3, CAP4	Yes	The informationToSendParameters for PromptAndCollectUserInformation are set in the same way as for PlayAnnouncement.
`informationToSend.inbandinfo.messageID.text.messageContent`	CS1, CS2, CAP2, CAP3, CAP4	No	
`informationToSend.inbandinfo.messageID.text.attributes`	CS1, CS2, CAP2, CAP3, CAP4	No	
`informationToSend.inbandinfo.messageID.elementaryMessageIDs`	CS1, CS2, CAP2, CAP3, CAP4	Yes	You configure announcement name and language pairs to link to elementary message IDs in ACS. You select the announcement names in control plans and ACS maps them to the appropriate elementary message IDs.
`informationToSend.inbandinfo.messageID.variableMessage.elementaryMessageID`	CS1, CS2, CAP2, CAP3, CAP4	Yes	ACS uses variable part announcements for some functions. You select and map the announcement names to variable elementary message IDs as described for the `elementaryMessageID` parameter.
`informationToSend.inbandinfo.messageID.variableMessage.variableParts`	CS1, CS2, CAP2, CAP3, CAP4	Yes	ACS supports all five types of variable part (time, price, date, number, and integer). The contents of these parts can be: • A fixed value • Derived from the database or an external platform such as a billing engine
`informationToSend.inbandinfo.numberOfRepetitions`	CS1, CS2, CAP2, CAP3, CAP4	Yes	Configurable in the control plan
`informationToSend.inbandinfo.duration`	CS1, CS2, CAP2, CAP3, CAP4	Yes	Configurable in the control plan
`informationToSend.inbandinfo.interval`	CS1, CS2, CAP2, CAP3, CAP4	Yes	ACS sets this parameter to zero (0)
`informationToSend.tone.toneID`	CS1, CS2, CAP2, CAP3, CAP4	No	

Parameter	Protocol	Yes	Notes
informationToSend.tone.duration	CS1, CS2, CAP2, CAP3, CAP4	No	
extensions	CS1, CS2, CAP2, CAP3, CAP4	Yes	If the type of SRF is "NAP", then ACS adds the Unisys NAP type extension, type 3. See *PromptAndCollectUserInformation NAP Extension Added by ACS* (on page 52) for more information.
callSegmentID	CS2, CAP4	No	
requestAnnouncementStartedNotification	CAP4	No	

PromptAndCollectUserInformation NAP Extension Added by ACS

The PromptAndCollectUserInfromation NAP Extension that ACS adds is valid for various SRFs including the Unsisys NAP type of SRF. The ASN.1 for the PromptAndCollectuserInformation extension type is equivalent to the following extension type definition:

```
NapExtension = SEQUENCE {

    language [0] INTEGER (1..99) OPTIONAL,
    treatments [1] SEQUENCE {

        treat1stCancelDig ENUMERATED { -- Treatment of first cancel digit entered.

            rePrompt(0),
            rePromptAndReturnDigitToSCF(1),
            returnDigitToSCF(2)

        } OPTIONAL,

        treat1stEORDig ENUMERATED { -- Treatment of first end of reply digit entered.

            erroneousInput(0),
            returnDigitsToSCF(1)

        } OPTIONAL,

        treatErroneousInput ENUMERATED { -- Treatment of erroneous input.

            returnErrorToSCF(1),
            repromptThenReturnPatialEntr(2)

        } OPTIONAL,

        treatCancelDig ENUMERATED { -- treatment of non-first cancel digit.

            rePrompts(0),
            announceAndRePrompt(1)

        } OPTIONAL
    }
}
```

For a NAP type SRF, ACS always adds all the announcement values defined in the treatments section. These values are globally configurable. For each announcement, you can also configure whether or not ACS adds the language.

Operations and Results Sent from an SRF to ACS

About Operations and Results Sent from an SRF to ACS

This section lists the operations and results that can be sent from a Specialized Resource Function (SRF) to ACS, and provides details about the supported parameters for each operation or result. Unless explicitly stated, the SRF can be an SSF based SRF (internal) or an SRF on an external intelligent peripheral that is connected to ACS by using an EstablishTemporaryConnection operation.

The following operations and results are the only operations and results sent from an SRF that can be received by ACS:

- *AssistRequestInstructions Operation (CS1, CS2, CAP2, CAP3, CAP4)* (on page 53) (external intelligent peripheral only)
- *PromptAndCollectUserInformation Result (CS1, CS2, CAP2, CAP3, CAP4)* (on page 54)
- *SpecializedResourceReport (CS1, CS2, CAP2. CAP3, CAP4)* (on page 54)

Notes:

- ACS raises an alarm and replies with TCAP_ABORT if it receives an unsupported operation from an SRF.
- ACS ignores any unsupported parameters in operations sent by the SRF.

AssistRequestInstructions Operation (CS1, CS2, CAP2, CAP3, CAP4)

The following table lists the parameters and protocols used in AssistRequestInstruction operations, and identifies which parameters and protocols are supported by ACS. The parameters are listed in the order they appear in the specification documents.

Note: The AssistRequestInstruction operation can be sent to ACS only from an external intelligent peripheral.

Parameter	Protocol	Supported	Notes
correlationID	CS1, CS2, CAP2, CAP3, CAP4	Yes	Identifies the call to which this operation relates.
iPAvailable	CS1, CS2	No	
iPSSPCapabilit ies	CS1, CS2, CAP2, CAP3, CAP4	No	
extensions	CS1, CS2, CAP2, CAP3, CAP4	No	

PromptAndCollectUserInformation Result (CS1, CS2, CAP2, CAP3, CAP4)

This table lists the parameters and protocols used in PromptAndCollectUserInformation results, and identifies which parameters and protocols are supported by ACS. The parameters are listed in the order they appear in the specification documents.

Parameter	Protocol	Supported	Notes
digitsResponse	CS1, CS2, CAP2, CAP3, CAP4	Yes	ACS adds the digits to its announcement list in the chassis context where possible.
iA5Response	CS2	No	

SpecializedResourceReport (CS1, CS2, CAP2. CAP3, CAP4)

The following table lists the parameters and protocols used in SpecializedResourceReport operations, and identifies which parameters and protocols are supported by ACS. The parameters are listed in the order they appear in the specification documents.

Note: ACS supports the SpecializedResourceReport operation for CS1, CS2, CAP2, CAP3, and CAP4, however only CAP4 specifies parameters for this operation.

Parameter	Protocol	Supported	Notes
allAnnouncementsComplete	CAP4	No	
firstAnnouncementStarted	CAP4	No	

Operations and Results Sent Between ACS and an HLR

About Operations and Results Sent Between ACS and an HLR

The MAP AnyTimeInterrogation operation is the only operation used between core ACS and an Home Location Register (HLR).

This section provides details about the supported parameters for the AnyTimeInterrogation operation.

AnyTimeInterrogation Operation (MAP Protocol) Sent from ACS to HLR

The following table lists the parameters used in AnyTimeInterrogation operations sent from ACS to an HLR using the MAP protocol, and identifies which parameters are supported by ACS. The parameters are listed in the order they appear in the MAP specification documents.

Parameter	Protocol	Yes	Notes
subscriberIdentity.imsi	MAP	No	
subscriberIdentity.msisdn	MAP	Yes	ACS always sets this parameter
requestedInfo.locationInformation	MAP	Yes	ACS always sets this parameter

Parameter	Protocol	Yes	Notes

`requestedInfo.`
`subscriberStat`
`e`
	MAP	Yes	ACS always sets this parameter

`requestedInfo.`
`extensionConta`
`iner`
	MAP	No	

`gsmSCF-Address`	MAP	Yes	ACS always sets this parameter. Its value is configured globally.
`extensionConta` `iner`	MAP	No	

AnyTimeInterrogation Operation (MAP protocol) Sent from HLR to ACS

This table lists the parameters used in AnytimeInterrogation operations sent from an HLR to ACS using the MAP protocol, and identifies which parameters are supported by ACS. The parameters are listed in the order they appear in the MAP specification documents.

Parameter	Protocol	Supported	Notes

`subscriberInfo`
`.locationInfor`
`mation.ageOfLo`
`cationInformat`
`ion`
	MAP	No	

`subscriberInfo`
`.locationInfor`
`mation.geograp`
`hicalInformati`
`on`
	MAP	No	

`subscriberInfo`
`.locationInfor`
`mation.vlr-`
`number`
	MAP	Yes	Handled by ACS and can be stored in the chassis context

`subscriberInfo`
`.locationInfor`
`mation.locatio`
`nNumber`
	MAP	Yes	Handled by ACS and can be stored in the chassis context

`subscriberInfo`
`.locationInfor`
`mation.cellIdO`
`rLAI`
	MAP	Yes	Handled by ACS and can be stored in the chassis context

`subscriberInfo`
`.locationInfor`
`mation.extensi`
`onContainer`
	MAP	No	

`subscriberInfo`
`.subscriberSta`
`te`
	MAP	No	

Parameter	Protocol	Supported	Notes
subscriberInfo.extensionContainer	MAP	No	
extensionContainer	MAP	No	