

Business Applications Support System (BASS)

Interface Specification

Revision 1.0

Scientific-Atlanta, Inc.

Unpublished Works of Scientific-Atlanta, Inc.

Copyright© 1997,1998, 1999, 2000 Scientific-Atlanta, Inc. All Rights Reserved.

NOTICE OF DISCLAIMER

This interface definition/system overview is published by Scientific-Atlanta, Inc. (S-A) to inform the industry of the requirements and operational characteristics for interaction with the S-A Digital Broadband Delivery System for the delivery of broadcast and/or interactive video services.

S-A reserves the right to revise this document at any time for any reason, including but not limited to, conformity with standards promulgated by various agencies or industry associations, 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.

S-A makes no representation or warranty, express or implied, of any kind with respect to this document or the any of the information contained herein. **YOUR USE OF OR RELIANCE UPON THIS DOCUMENT AND THE INFORMATION CONTAINED HEREIN IS AT YOUR OWN RISK.** S-A shall not be liable to you for any damage or injury incurred by any person arising out of the your use of these materials.

This document is not to be construed as conferring by implication, estoppel or otherwise any license or right under any copyright or patent whether or not the use of any information in this document necessarily employs an invention claimed in any existing or later issued patent.

Table of Contents

Revision 1.0

1. Introduction	1-1
1.1 Purpose	1-1
1.2 Scope	1-1
1.3 Definitions, Acronyms, and Abbreviations	1-1
1.4 References	1-2
1.5 Revision History	1-2
2. BASS Interface Principles of Operations	2-1
2.1 Protocols	2-1
2.1.1 Message Protocol	2-1
2.1.2 Transport Protocol	2-1
2.1.3 Digital Storage Module Command and Control	2-1
2.1.4 Descriptors	2-1
2.2 Reference Architecture	2-1
2.2.1 Digital Network Control System (DNCS)	2-2
2.2.2 S-A Applications Platform	2-5
2.2.3 BOSS Server	2-5
2.2.4 Administrative Gateway (AG)	2-5
2.3 BASS Interface Managed Objects	2-5
2.3.1 WatchTv Service	2-5
2.3.2 Virtual Channel Service	2-6
2.3.3 Emergency Alert	2-6
2.3.4 DHCT Configuration	2-6
2.3.5 PPV Service, PPV Event, PPV Authorization	2-6
3. BASS Interface Coding Standards	3-1
4. BASS Descriptors	4-1
4.1 BASS Interface Descriptor Summary	4-1
4.2 BASS Interface Descriptor Definitions	4-12
4.2.1 AdvertisingWindow	4-12
4.2.2 AlertType	4-12
4.2.3 BillingId	4-12
4.2.4 BlockingPin	4-12
4.2.5 DatabaseOperation	4-12
4.2.6 DefaultAdvertisingWindow	4-13
4.2.7 DefaultEventCost	4-13
4.2.8 DefaultEventOrderPhoneNumber	4-13

4.2.9	DefaultMarketingWindow	4-13
4.2.10	DhctMacAddr	4-13
4.2.11	DhctMacAddrList	4-14
4.2.12	DhctMessage	4-14
4.2.13	DhctParameter	4-14
4.2.14	DisplayType	4-14
4.2.15	EventInfo	4-14
4.2.16	EventList	4-14
4.2.17	EventOrderPhoneNumber	4-14
4.2.18	EventUseService	4-14
4.2.19	ForceTypeType	4-14
4.2.20	GroupName	4-15
4.2.21	ImpulsePayPerView	4-15
4.2.22	InterstitialServiceName	4-15
4.2.23	LimitedDuration	4-15
4.2.24	Logo	4-15
4.2.25	LongDescription	4-15
4.2.26	MarketingWindow	4-15
4.2.27	MessageTime	4-15
4.2.28	MessageType	4-16
4.2.29	MultiLingualTitle	4-16
4.2.30	OrderSource	4-16
4.2.31	PackageAuthorization	4-16
4.2.32	PackageName	4-16
4.2.33	PayPerView	4-16
4.2.34	PurchasePin	4-16
4.2.35	RecordCount	4-16
4.2.36	Result	4-17
4.2.37	ServiceName	4-19
4.2.38	ShortDescription	4-19
4.2.39	SourceId	4-19
4.2.40	SourceUrl	4-19
4.2.41	StartRecordNumber	4-19
4.2.42	SubscriptionServiceName	4-19
4.2.43	Text	4-20
4.2.44	UrgencyType	4-20
4.2.45	UploadFileSize	4-20
4.2.46	VodArchivePath	4-20
4.2.47	VodPurchase	4-20
4.2.48	VodUploadFile	4-20
4.2.49	VodUploadPath	4-20
5.	<i>BASS Interface Transactions</i>	<i>5-1</i>
5.1	BASS Transaction Summary	5-1
5.2	General Transaction Format	5-14
5.2.1	Request Format	5-14
5.2.2	Simple Response Format	5-14
5.2.3	Complex Response Format	5-14

5.3	WatchTv Service Transactions	5-15
5.3.1	DefineWatchTvService	5-15
5.3.2	RetireWatchTvService	5-15
5.4	Music Service Transactions	5-15
5.4.1	DefineMusicService	5-15
5.4.2	RetireMusicService	5-16
5.5	Virtual Channel Service Transactions	5-16
5.5.1	DefineVirtualChannelService	5-16
5.5.2	RetireVirtualChannelService	5-16
5.6	DHCT Configuration Transactions	5-17
5.6.1	SetPin	5-17
5.6.2	SetDhctParameter	5-17
5.7	Pay Per View Transactions	5-17
5.7.1	DefinePpvService	5-18
5.7.2	RetirePpvService	5-18
5.7.3	DefinePpvEvent	5-18
5.7.4	RetirePpvEvent	5-19
5.7.5	AddRppvAuthorization	5-19
5.7.6	RemoveRppvAuthorization	5-19
5.8	Messaging Transactions	5-20
5.8.1	DefineMessageGroup	5-20
5.8.2	AddDhctToGroup	5-20
5.8.3	RemoveDhctFromGroup	5-21
5.8.4	RetireMessageGroup	5-21
5.8.5	SendSamMessage	5-21
5.8.6	SendMessage	5-22
5.9	Multi-Media Messaging Transactions	5-22
5.9.1	SendMultiMediaMessage	5-22
5.9.2	DefineMultiMediaMessageConfiguration	5-23
5.9.3	RetireMultiMediaMessageConfiguration	5-23
5.10	VOD Service Transactions	5-23
5.10.1	DefineVodService	5-23
5.10.2	RetireVodService	5-24
5.11	VOD Upload Transactions	5-24
5.11.1	InitializeVodUpload	5-24
5.11.2	VodUploadComplete	5-24
5.11.3	GetVodRecords	5-24
5.12	Usage Data Interface	5-25

1. Introduction

1.1 Purpose

This document describes the Business and Applications Support System (BASS) interface, which is the interface between Scientific-Atlanta-supplied Application Servers and the Administrative Gateways of Access Network Providers (ANPs) for both Analog Broadcast Services and Digital Broadcast Services.

The intended audience of this document consists of:

- Customers of the BASS Interface both internal and external
- Developers of Application Server software components
- Test Engineers, tasked with authoring the BASS Interface Test Plan and Test Procedure, and with performing the tests
- Technical Writers, tasked with documenting the implementation and usage of Application Servers.

1.2 Scope

This document describes BASS Interface transactions at the messaging level. It suggests an Applications Programmatic Interface (API) for use with the BASS interface, but it places no requirements on that API.

1.3 Definitions, Acronyms, and Abbreviations

Terms	Description
AG	Administrative Gateway
BASS	Business and Application Support System
BFS	Broadcast File System
CA	Conditional Access
DBDS	Digital Broadband Delivery System
DHCT	Digital Home Communications Terminal
DIS	Digital Interactive Service
DMS	Digital Multicast Service
DNCS	Digital Network Control System
DSM-CC	Digital Storage Module Command and Control
EMM	Entitlement Management Message
HTML	Hyper Text Markup Language

Terms	Description
IP	Internet Protocol
MAC Address	Media Access Control
MIB	Management Information Base
MPEG	Moving Pictures Experts Group
ONC	Open Network Computing
PAT	Program Association Table
PES	Packetized Elementary Stream
PIN	Personal Identification Number
PMT	Program Map Table
PSI	Program Specific Information
RPC	Remote Procedure Call
SMU	Server Management Unit
SNMP	Simple Network Management Protocol
SQL	Structured Query Language
XDR	eXternal Data Representation

1.4 References

- [1] Digital Broadband Delivery System Phase 1.0 System Description
- [2] Business and Operations Support System (BOSS) Phase 1.0 Interface Specification
- [3] RFC1050: RPC: Remote Procedure Call Protocol Specification
- [4] RFC1057: RPC: Remote Procedure Call Protocol Specification: Version 2
- [5] RFC1832: XDR: External Data Representation Standard
- [6] ISO/IEC DIS 13818-6, MPEG-2 Digital Storage Media Command & Control
- [7] Explorer Application Platform Applications Programmatic Interface Phase 1.0
- [8] WatchTv Server Software Requirements Specification
- [9] Virtual Channel Server Software Requirements Specification
- [10] Configuration Server Software Requirements Specification
- [11] Pay Per View Server Software Requirements Specification

1.5 Revision History

Table 1-1 identifies the revision history of this document.

Table 1-1: Revision History

Revision	Modifications	Date	Initials
-----------------	----------------------	-------------	-----------------

Revision	Modifications	Date	Initials
0.1	First Version – Greg Durden and Darryl DeFreese		GD, DD
0.2	Added EAS transaction, which was inadvertently omitted	10/26/1997	DD
0.2	Added Messaging transactions	10/26/1997	DD
0.2	Added Music transactions	10/26/1997	DD
0.2	Made MarketingWindow optional	11/3/1997	DD
0.2	Added an AdvertisingWindow descriptor	11/3/1997	DD
0.3	Marked WatchTv and Music transactions as apportioned to phase 1.1	11/11/1997	DD
0.3	Moved eventOrderPhoneNumber from service to event	11/11/1997	DD
0.3	Changed description of marketing window and advertising window	11/11/1997	DD
0.3	Added a description of the cost field	11/11/1997	DD
0.3	Changed all the serviceIds in the PPV transactions to shortDescriptions	11/11/1997	DD
0.4	(Revision 0.4 was inadvertently deleted. It was reconstituted by Darryl DeFreese on 3/3/1998.)	3/3/1998	DD
0.4	Changed advertisingWindowInterval to marketingWindowInterval	12/2/1997	DD
0.4	Changed WatchTv to Music on DefineMusicService transaction	12/2/1997	DD
0.4	Enforced a consistent usage of service short description	12/7/1997	DD
0.4	Added an available time to the EventOrderPhoneNumber descriptor	12/7/1997	DD
0.4	Added the OrderSource descriptor	12/7/1997	DD
0.5	Corrected types to conform to XDR	3/3/1998	DD
0.5	Corrected references to serviceId to refer to serviceName	3/3/1998	DD
0.5	Changed references to [service] short descriptions to serviceName	3/3/1998	DD
0.5	Replaced ServiceId descriptor with ServiceName descriptor	3/3/1998	DD
0.5	Made AdvertisingWindow descriptor optional on DefinePpvEvent transaction, and added a DefaultAdvertisingWindow descriptor to the DefinePpvService transaction.	3/16/1998	DD
0.5	Made MarketingWindow descriptor optional on DefinePpvEvent transaction, and added a DefaultMarketingWindow descriptor to the DefinePpvService transaction.	3/16/1998	DD
0.5	Made EventOrderPhoneNumber descriptor optional on DefinePpvEvent transaction, and added a Default EventOrderPhoneNumber descriptor to the DefinePpvService transaction.	3/16/1998	DD
0.5	Updated result codes.	3/16/1998	DD

Revision	Modifications	Date	Initials
0.6	Replaced EAS transactions with MultiMediaMessage transactions	6/22/1998	DD
0.6	Added a detailed per-transaction release plan	6/22/1998	DD
0.6	Added initial VOD billing support transactions.	6/22/1998	DD
0.7	Added the MultiLingualTitle descriptor and associated result codes.	8/10/1998	DD
0.7	Removed incorrect range information on the eventCost field of the EventInfo descriptor.	8/17/1998	DD
0.7	Changed the eventCost field to 8 characters.	8/17/1998	DD
0.7	Added a default event cost to the DefinePpvService transaction.	8/17/1998	DD
0.8	Removed the DefineVodEvent transaction (this billing interface model is no longer to be supported.)	12/11/1998	DD
0.8	Added InitializeVodUpload, GetVodRecords, and VodUploadComplete transactions	12/11/1998	DD
0.8	Added the ResetClientNvm transaction	12/11/1998	DD
0.8	Eliminated the result code numbers from this document. Result codes must be checked against the enumerated name, not actual numbers.	3/10/1999	DD
0.8	Add the proposed VOD purchase record format.	3/10/1999	DD
0.8	Added several descriptors to support the VOD transactions.	3/10/1999	DD
0.9	Corrected the AddRppvAuthorization transaction to use the PackageAuthorization descriptor.	4/8/1999	DD
1.0	Corrected response to GetVodRecords transaction	3/14/2000	GSD
1.0	Corrected VOD purchase record format	3/14/2000	GSD
1.0	Added assetOrganization, assetId to VodPurchase descriptor	3/14/2000	GSD
1.0	Added descriptors to Section 4.2 as required	3/14/2000	GSD
1.0	Added transactions to Section 5.2 as required	3/14/2000	GSD

2. BASS Interface Principles of Operations

2.1 Protocols

2.1.1 Message Protocol

The BASS interface utilizes the messaging protocol described in [2].

2.1.2 Transport Protocol

The BASS interface utilizes the transport protocol described in [2].

2.1.3 Digital Storage Module Command and Control

Digital Storage Module Command and Control (DSM-CC) represents the single avenue by which broadband connections, or sessions, may be created and manipulated. Any client of the network, including an Application Server, requiring a session through the network must request that session by communicating with the DNCS using DSM-CC.

2.1.4 Descriptors

Many of the following transactions make use of the *Descriptor* concept. The descriptor is a structure containing fields that describe some aspect of a BASS interface managed object. A single transaction may carry several descriptors to completely describe the desired state of a single managed object.

Descriptors are passed in transactions using descriptor loops. Each loop contains some combination of descriptors to describe the desired state of a single managed object. Many BASS transactions allow multiple loops, each of which specifies the state of a separate managed object. Each descriptor carries an individually unique combination of fields. Descriptors are positionally interchangeable within a descriptor loop.

The BASS interface descriptors are defined in *Section 4 -- BASS Descriptors*

2.2 Reference Architecture

The following diagram illustrates the system-context in which the BASS interface is defined to operate. Items to the right of the dotted line are Scientific-Atlanta supplied components. Items to the left of the dotted line are non-Scientific-Atlanta products required for proper operation of the BASS interface.

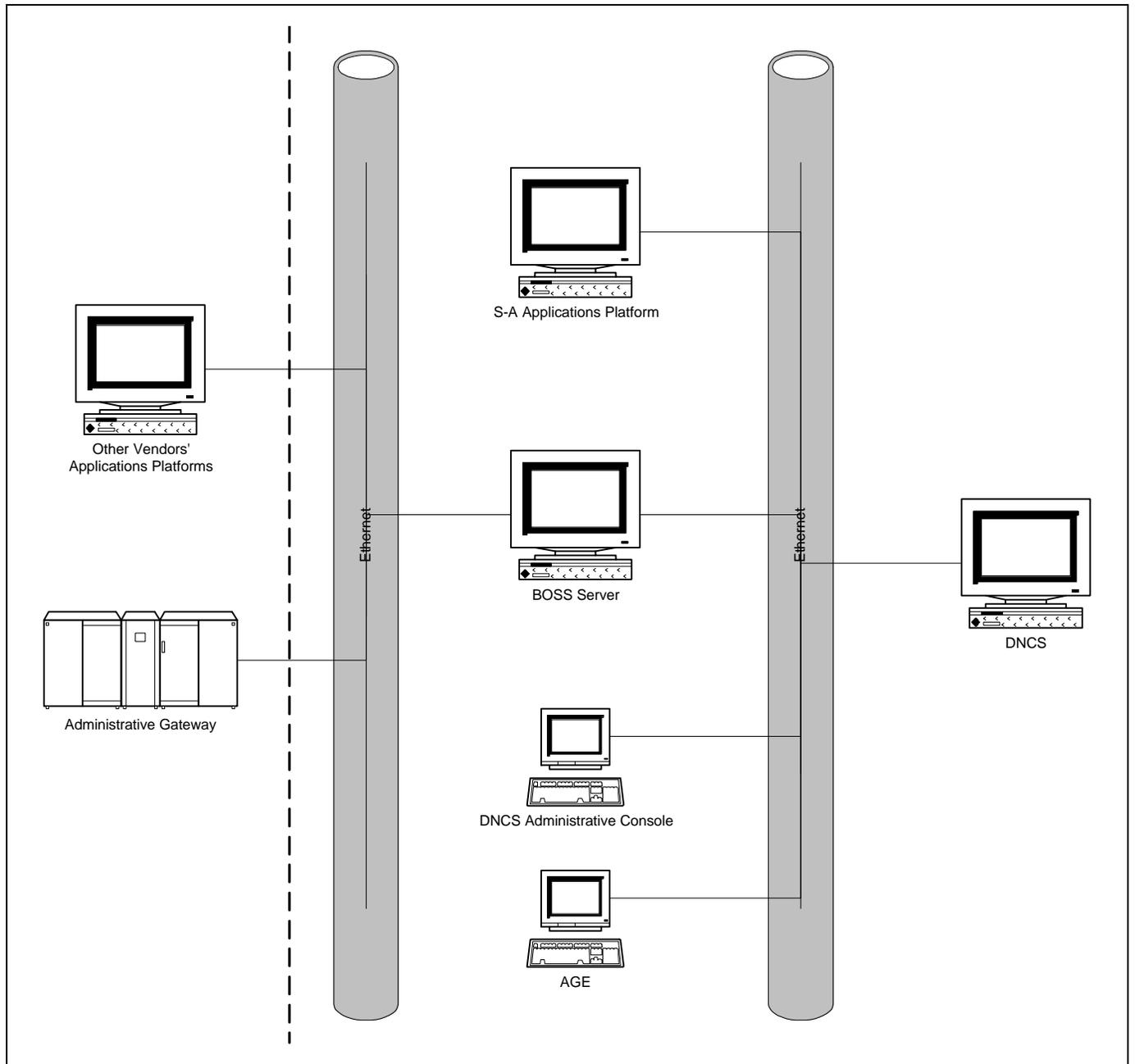


Figure 2-1 -- Reference Architecture

2.2.1 Digital Network Control System (DNCS)

The DNCS is the control and management system for the Scientific-Atlanta digital network. This section gives an overview of the basic functionality it provides.

2.2.1.1 Core Functionality

The features described in this section are controlled via the Business Operations Support System (BOSS) Interface, described in [2].

2.2.1.1.1 Broadcast Control Suite

The Broadcast Control Suite (BCS) works in conjunction with the Administrative Gateway for the definition of broadcast-based services supplied via the Scientific-Atlanta Digital Broadband Delivery System.

The Broadcast Control Suite is responsible for the definition of bandwidth segments. The bandwidth segment is an abstraction of a session over some particular time, which may be unlimited. The BCS manages the higher-level segment, but not the underlying session carrying the segment. Management of sessions is the responsibility of other DNCS components. The segment provides an interface for scheduling events, an interface for scheduling encryption status of a segment, and the basis for forming segment offerings.

2.2.1.1.2 PowerKEY™ Control Suite

The PowerKEY™ Control Suite is the component of the DNCS responsible for the encryption, key management, and other conditional access functions of the PowerKEY™ system.

The Conditional Access Manager (CAM) is a component of the PowerKEY™ Control Suite. The CAM manages end-user access to services supplied via the DBDS through the use of packages.

The package is the unit of authorization on which the CAM grants permission to use the defined segments. A given package may contain any number of segments. When a user is granted authorization (permission) for the package, that user has the right to use any of the segments contained in the package.

The CAM, in conjunction with the BCS, provides conditional access services for digital broadcast networks, and manages bandwidth, provisioning, and encryption keys for the service encryptor.

2.2.1.1.3 Inventory & Directory Manager

The Inventory & Directory Manager (IDM) stores widely useful information about entities in the Digital Network Control System (DNCS). It makes this information available to entities throughout the DNCS. As examples of its contents, the IDM will hold DHCT MAC address to IP address mappings, distinguished name to IP address mappings, and public cryptographic key certificates for Content Providers and DHCTs.

2.2.1.2 Server Application Platform

The Server Application Platform provides features useful to developers of applications running on the digital network.

2.2.1.2.1 Broadcast File System (BFS)

The Broadcast File System (BFS) provides a mechanism for delivering various types of data from a group of servers to a client attached to a network. This data may contain any type of information. Both the server and client view the data as a file system similar to those found on disk operating systems. The BFS is implemented to appear to the servers and clients as a standard hierarchical file system, which is common in computer operating systems. The underlying mechanism for transporting files from the servers to a client relies on a broadcast data carousel mechanism, which is defined in [6].

The interface between the Server, Client and BFS is defined in [7].

2.2.1.2.2 Service Application Manager (SAM)

The Service Application Manager (SAM) is the component of the applications platform that is tasked with presenting a framework for the execution of services. In this system, a service is defined as the combination of an Application Component and a Parameter Component. An example of an Application Component would be “ViewVideo,” while an example of a service would be “ViewVideo + HBO.” The SAM provides the connection between each service’s Application Component and its Parameter Component.

The interface to the SAM is provided by BOSS transactions, defined in [6].

2.2.1.2.3 Messaging

The Messaging component of the Applications Platform provides the ability to send messages to one or more DHCTs. The messages may take the form of a simple pass-through message, or may activate a previously defined service on the DHCTs.

The interface to the Messaging component is defined in [7].

2.2.1.3 DNCS Administrative Console

The DNCS Administrative Console provides a Graphical User Interface (GUI) allowing a DNCS operator to provision, control, and monitor DNCS components and other components of the Scientific-Atlanta Digital Broadband Delivery System (DBDS). These components may include digital headend equipment, PowerKEY™ equipment, and digital home communications terminals (DHCTs). Additionally, the Administrative Console allows for the configuration of communications paths to external equipment.

2.2.1.4 Administrator Gateway Emulator

The Administrative Gateway Emulator (AGE) provides a Graphical User Interface (GUI) allowing a DNCS operator to directly enter BASS transactions and view the results. The AGE is used in installations lacking an Administrative Gateway system. One example of such an installation would be one in which the Administrative Gateway is a legacy system and has not yet been upgraded to process the BASS interface.

2.2.2 S-A Applications Platform

One or more S-A Applications Platforms will be present in a typical network to provide an execution platform for S-A developed Application Servers. An Application Server is a software system that manages and transmits data required by Application Clients running on DHCTs. For example, an Impulse Pay-Per-View (IPPV) Application Server might be used to provide IPPV event schedule information to an Application Client that allows a user of a DHCT to select an IPPV event for purchase.

2.2.3 BOSS Server

The BOSS Server provides a mechanism for the routing of a BASS request to the appropriate Application Server and BASS response to the BASS client that initiated the request.

2.2.4 Administrative Gateway (AG)

The Administrative Gateway controls access to and modifies data held by Application Servers. The AG implements the access network's policies concerning service and authorization provisioning. When the AG has validated policy compliance, it transmits a BASS request to the appropriate Application Server to execute the request.

Application Servers will execute operations received from the Administrative Gateway without further access checks.

Note: Administrative Gateways are provided by other manufacturers.

2.3 BASS Interface Managed Objects

The BASS Interface provides remote access to several categories of objects managed by S-A supplied Application Servers.

2.3.1 WatchTv Service

A WatchTv service is the equivalent in the digital network to a normal analog television channel. The WatchTv application server can manage multiple WatchTv services, each of which equates to a single television channel. When the WatchTv application client is activated at the DHCT, it tunes the appropriate content stream, either an MPEG program or an analog channel, and, if authorized, presents the content stream to the display.

2.3.2 Virtual Channel Service

A Virtual Channel service is an HTML-based presentation of information. The Virtual Channel application server can manage multiple Virtual Channel services, each of which is presented on a single channel. The Virtual Channel server transports the content for each of its managed services to the DHCT in a BFS file. When the Virtual Channel application client is activated, the BFS file is retrieved and presented to an HTML engine for rendering to the display.

2.3.3 Emergency Alert

The Emergency Alert server provides a message to its client that an Emergency Alert System (EAS) message has been received and is available for the user. A text message is shown to the user, who is given the opportunity to tune to another service for additional information.

2.3.4 DHCT Configuration

The DHCT and S-A applications have several parameters that may be configured on a per-DHCT basis.

2.3.5 PPV Service, PPV Event, PPV Authorization

A Pay Per View (PPV) service provides the capability to purchase time-limited portions of another service (called PPV events,) on a per-use basis. Impulse PPV events may be purchased and authorized directly via the DHCT user interface. Reservation PPV events must be authorized by the AG.

3. BASS Interface Coding Standards

The BASS interface utilizes the coding standards described in [2].

4. BASS Descriptors

4.1 BASS Interface Descriptor Summary

The table in this section (Table 4-1) provides a summary of all valid BASS descriptors. It includes all fields in each descriptor, the types of the fields, an indication of the valid range of the field, a length for character strings, and, in some cases, an explanatory comment.

Note that the null terminator is NOT included in the lengths specified in this table.

A cross-index of valid descriptors per BASS transaction can be found in Table 5-1.

Table 4-1: BASS Interface Descriptor Summary

AdvertisingWindow		Sent in the DefinePpvEvent transaction to determine how long a IPPV event should be advertised.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
advertisingWindowStartTime	unsigned integer			Time to begin advertising the event, for advance purchase -- UNIX time
advertisingWindowInterval	unsigned integer			Seconds
AlertType		Sent in the MMMConfig transaction to determine information on the notification methods to the user.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
messageAlert	enum	ma_None ma_Text ma_Light ma_Both		This field determines how a subscriber is notified when an MMM is arrived. When an MMM is arrived: ma_None means no alert is given; ma_Text means text defined in the alertText field is displayed; ma_Light means that message LED will be turned; ma_Both means display both text and message LED;
alertText	string<>		100	This field determines a 2-line text if the messageAlert field is either ma_Text or ma_Both. This field is NULL terminated.
alertLEDRate	unsigned int			This field determines the rate of LED blinking in cycles per second. This field is only applicable for the options of ma_Light and ma_Both in the messageAlert field.
alertLEDDuration	unsigned int			This field determines the length in seconds of the interval during which blinking occurs. This field is only applicable for the options of ma_Light and ma_Both in the messageAlert field.
alertLEDPeriod	unsigned int			This field determines how often in seconds the blinking interval is repeated. This field is only applicable for the options of ma_Light and ma_Both in the messageAlert field.
BillingId		Used to restrict an InitializeVodUpload transaction to the DHCTs owned by the identified SMS.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>

billingId	unsigned integer	0..2 ¹⁶ -1		
BlockingPin		Sent to specify the new value of the channel blocking PIN.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
BlockingPinEnable	enum	BPE_Enabled BPE_Disabled		
BlockingPin	string[]		4	The pin may be any 4 numeric
DatabaseOperation		Sent in the definitional BASS transactions to indicate whether a strict insertion or a strict update is desired. If this descriptor is omitted from a transaction, the appropriate database operation will be determined by the server from the state of the server's database.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
databaseOperation	enum	do_InsertOnly do_UpdateOnly		When this field is set to do_UpdateOnly, all other descriptors in the transaction are optional, except the key fields for the transaction.
DefaultAdvertisingWindow		Sent in the DefinePpvService transaction to provide a default interval over which IPPV events should be advertised.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Comments</i>	
startAnchor	enum	a_eventStart a_eventEnd a_marketingWindowStart a_marketingWindowEnd a_buyWindowStart a_buyWindowEnd a_advertisingWindowStart a_advertisingWindowEnd a_previousEventStart a_previousEventEnd a_previousMarketingWindowStart a_previousMarketingWindowEnd a_previousBuyWindowStart a_previousBuyWindowEnd a_previousAdvertisingWindowStart a_previousAdvertisingWindowEnd	The event to be used as an anchor for the interval specified in the next field. It is invalid to specify a_advertisingWindowStart or a_advertisingWindowEnd for this field in this descriptor.	
startInterval	integer		An interval, in seconds, that specifies an offset from the event indicated in the startAnchor field. The startInterval added to the time of that event indicates the start time for the advertising window.	

endAnchor	enum	a_eventStart a_eventEnd a_marketingWindowStart a_marketingWindowEnd a_buyWindowStart a_buyWindowEnd a_advertisingWindowStart a_advertisingWindowEnd a_previousEventStart a_previousEventEnd a_previousMarketingWindowStart a_previousMarketingWindowEnd a_previousBuyWindowStart a_previousBuyWindowEnd a_previousAdvertisingWindowStart a_previousAdvertisingWindowEnd	The event to be used as an anchor for the interval specified in the next field. It is invalid to specify a_advertisingWindowStart or a_advertisingWindowEnd for this field in this descriptor.	
endInterval	integer		An interval, in seconds, that specifies an offset from the event indicated in the endAnchor field. The endInterval added to the time of that event indicates the end time for the advertising window.	
DefaultEventCost		Sent in the DefinePpvService transaction to provide a default cost for events defined against the service.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
eventCost	string[]		8	This field is sent to provide a default cost for events. It is only used when the eventCost field in the EventInfo descriptor is set to the ASCII null character ('\0').
DefaultEventOrderPhoneNumber		Sent in the DefinePpvService transaction to provide a default ordering telephone number for events defined against the PPV service.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
eventOrderPhoneNumber	string<>		20	
startAnchor	enum	a_eventStart a_eventEnd a_marketingWindowStart a_marketingWindowEnd a_buyWindowStart a_buyWindowEnd a_advertisingWindowStart a_advertisingWindowEnd a_previousEventStart a_previousEventEnd a_previousMarketingWindowStart a_previousMarketingWindowEnd a_previousBuyWindowStart a_previousBuyWindowEnd a_previousAdvertisingWindowStart a_previousAdvertisingWindowEnd	The event to be used as an anchor for the interval specified in the next field.	
startInterval	integer		An interval, in seconds, that specifies an offset from the event indicated in the startAnchor field. The startInterval added to the time of that event indicates the start time for the advertising window.	

DefaultMarketingWindow		Sent in the DefinePpvService transaction to provide a default interval over which IPPV events should be advance-purchasable.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Comments</i>	
startAnchor	enum	a_eventStart a_eventEnd a_marketingWindowStart a_marketingWindowEnd a_buyWindowStart a_buyWindowEnd a_advertisingWindowStart a_advertisingWindowEnd a_previousEventStart a_previousEventEnd a_previousMarketingWindowStart a_previousMarketingWindowEnd a_previousBuyWindowStart a_previousBuyWindowEnd a_previousAdvertisingWindowStart a_previousAdvertisingWindowEnd	The event to be used as an anchor for the interval specified in the next field. It is invalid to specify a_marketingWindowStart or a_marketingWindowEnd for this field in this descriptor.	
startInterval	integer		An interval, in seconds, that specifies an offset from the event indicated in the startAnchor field. The startInterval added to the time of that event indicates the start time for the marketing window.	
endAnchor	enum	a_eventStart a_eventEnd a_marketingWindowStart a_marketingWindowEnd a_buyWindowStart a_buyWindowEnd a_advertisingWindowStart a_advertisingWindowEnd a_previousEventStart a_previousEventEnd a_previousMarketingWindowStart a_previousMarketingWindowEnd a_previousBuyWindowStart a_previousBuyWindowEnd a_previousAdvertisingWindowStart a_previousAdvertisingWindowEnd	The event to be used as an anchor for the interval specified in the next field. It is invalid to specify a_marketingWindowStart or a_marketingWindowEnd for this field in this descriptor.	
endInterval	integer		An interval, in seconds, that specifies an offset from the event indicated in the endAnchor field. The endInterval added to the time of that event indicates the end time for the marketing window.	
DhctMacAddr		Sent to identify the DHCT being described by the current descriptor loop.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
dhctMacAddr	string[]		18	
DhctMacAddrList		Sent to identify the set of DHCTs being described by the current descriptor loop.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
dhctMacAddrCount	unsigned			

	integer			
dhctMacAddr <>	string[]		18	
DhctMessage		Sent to specify a DSM-CC pass-through message to be sent to a set of DHCTs.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
dsmccMessageType	unsigned integer			
messageLength	unsigned integer			
messageBytes	opaque<>			
DhctParameter		Sent to set a configuration parameter on a DHCT.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
parameterName	string[]		20	
parameterType	enum	pt_Int pt_String		
intParameter	integer			This field is to be used only if parameterType == pt_Int.
stringParameter	string<>		1000	This field is to be used only if parameterType == pt_String.
DisplayType		Sent in the MMMConfig transaction to determine how to display the message.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
enablePageKeys	enum	epk_No epk_Yes		This field determines whether a page key can be used to page up and down through a MMM. epk_No means disallow use of the page keys. epk_Yes means allow use of page keys.
enablePage	enum	ep_No ep_Yes		This field determines whether a page - key can be used to go to the previous page. ep_No means disallow use of the page - key. ep_Yes means allow page - key
rollOver	enum	roll_None roll_HiToLow roll_LowToHi roll_Both		This field determines whether the pages should be rolled over from hi to low and low to hi with the + key. roll_None means do not roll from high to low or low to high. roll_HiToLow means allow rollover from hi to low but not from low to hi. roll_LowToHi means allow rollover from low to hi but not from hi to low. roll_Both means allow rollover from hi to low and low to hi.
displaySize	enum	ds_FullScreen ds_TopHalfScreen ds_BottomHalfScreen ds_2LineBanner ds_3LineBanner ds_Custom		This field specifies the size of display. ds_FullScreen means full screen. ds_TopHalfScreen means top half screen. ds_BottomHalfScreen means bottom half screen. ds_2LineBanner means two line banner. ds_3LineBanner means three line banner. ds_Custom allows the user to specify their coordinates in the displayCoords fields.
displayMotion	enum	dm_Fixed dm_ContinuousScroll dm_DelayedScroll		This field specifies the motion of the text. dm_Fixed means no motion. dm_ContinuousScroll means scroll the entire textual message, using the motionDirection and motionSpeed fields.

				dm_DelayedScroll means scroll the textual message in pieces, using the motionDirection, motionSpeed, and motionDelay fields.
motionDirection	enum	md_ScrollUp md_ScrollDown md_ScrollLeft md_ScrollRight		This field specifies direction of scrolling.
motionSpeed	enum	ms_Instant ms_Slow ms_Medium ms_Fast		This field specifies speed of scrolling. ms_Instant doesn't actually scroll but rather replaces (only valid w/ dm_DelayedScroll)
pageKeyRepeatPeriod	int			This field determines time in milliseconds between page changes with a page key held down .
displayCoordsXMin	Unsigned int			x min display window on screen.
displayCoordsYMin	Unsigned int			y min display window on screen.
displayCoordsXMax	Unsigned int			x max. display window on screen.
displayCoordsYMax	Unsigned int			y max. display window on screen.
motionDelay	int			delay in seconds between scrolling of message pieces during dm_DelayedScroll.
EventInfo		Sent in the DefinePpvEvent transaction to provide user-presentable descriptive information for IPPV events.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
eventTitle	string[]		24	Event title to be displayed by the DHCT during the IPPV purchase process. The language of this field is always assumed as English. When the MultiLingualTitle descriptor is included, an English-language title specified in that descriptor takes precedence over this title.
eventCost	string[]		8	Event cost to be displayed by the DHCT during the IPPV purchase process. This string contains a text string that is displayed verbatim on the DHCT PPV GUI. If a currency sign is desired on that display, it must be included in this field. If no cost is being specified for the event, this field should be set to the ASCII null character ('\0').
EventList		Sent in the DefinePpvEvent transaction to identify a list of previously defined PPV events that are to be collectively authorized as a unit.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
eventCount	unsigned integer			Number of events in the following list
packageNameList	PackageName<>			An array containing the packageNames of previously-defined PPV events that are to be authorized as a unit.
EventOrderPhoneNumber		Sent in the DefinePpvService transaction to define the telephone number to for a user to call to order a reservation PPV event.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
eventOrderPhoneNumber	string<>		20	
numberAvailableTime	unsigned integer			Time RPPV telephone order number becomes available – UNIX time

EventUseServiceName		Sent in the DefinePpvService transaction to indicate a service to utilize to access an event.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
serviceName	string<>		20	
ForceTuneType		Sent in the MMMConfig transaction to determine the force tune of a MMM configuration.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
acOutlet	enum	ao_Ignore ao_On		This field determines whether to turn on the DHCTs when an MMM is arrived. ao_Ignore causes no changes to the AC outlet. ao_On forces the AC outlet of DHCTs.
forceTune	enum	ft_None ft_ServiceId		This field determines whether to activate a service Id defined in the forceTuneServiceId field. ft_None means no changes in the current service and the forceTuneServiceId field is not applicable. ft_ServiceId means activates the service specified in the forceTuneServiceId field.
immediacy	enum	imd_KeyPress imd_Immediate		This field determines when a MMM should be activated/displayed. imd_KeyPress means that do not activate service Id until the user presses the trigger key defined by the triggerKey field. imd_Immediate means activate the service Id immediately upon arrival. If this option is selected, the triggerKey field is not applicable.
triggerKey	enum	0x00-0x2E & 0x40		This field determines the trigger key. 0x40 means any key.
forceTuneServiceId	ServiceId		16	This field determines the service ID to be activated when an MMM is arrived, if the option ft_ServiceId is selected in the forceTune field.
GroupName		Sent to identify the message group being described by the current descriptor loop.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
groupName	string<>		20	
ImpulsePayPerView		Sent in the DefinePpvEvent transaction to provide additional definition for packages that are to be purchasable via the DHCT on a per-use basis.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
freeCaStartTime	unsigned integer			UNIX time
freeCaInterval	unsigned integer			Seconds
buyWindowStartTime	unsigned integer			UNIX time
buyWindowInterval	unsigned integer			Seconds
cancelWindowInterval	unsigned integer			Seconds from start of package.
allowImmediateCancel	enum	AIC_True		

		AIC_False		
allowWindowedCancel	enum	AIC_True AIC_False		
modeCount	unsigned integer	1..7		Number of repetitions of modeLength, modeCost, and modeRightToCopy
modeLength	unsigned integer			Minutes
modeCost	unsigned integer			Units are determined during installation configuration. These units must be consistent with the creditLimit in the DhctState descriptor.
modeRightToCopy	enum	MRTC_CopyPermitted MRTC_CopyProhibited		
InterstitialServiceName		Sent in the DefinePpvService transaction to indicate a service to utilize during the interstitial period between PPV events.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
serviceName	string<>		20	
LimitedDuration		Sent in the DefinePpvEvent transaction to specify the interval over which a PPV or IPPV event is active.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
startTime	unsigned integer			UNIX time.
duration	unsigned integer			Minutes.
Logo		Sent in the service-defining BASS transactions to provide a graphical icon for the service being defined.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
logo	opaque<>		1024	
LongDescription		Sent in the service-defining BASS transactions to identify the service being defined.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
longDescription	string<>		100	
MarketingWindow		Sent in the DefinePpvEvent transaction to determine how long a IPPV event should be advance purchasable.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
marketingWindowStartTime	unsigned integer			Time to begin offering advance purchase of the event, for advance purchase -- UNIX time
marketingWindowInterval	unsigned integer			Seconds
MessageTime		Sent in the MMMConfig transaction to determine information on the time/length of the message.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
duration	int			This field determines time in seconds which an MMM is activated for each repetition.
repeats	int			This field determines number of times which an MMM is repeated.

delayBetweenRepeats	int			This field determines time in seconds between each MMM repetition.
expirationTime	int	.		This field determines expiration date/time when an MMM expires.
Message Type		Sent in the SendMMM transaction to determine the content of a MMM.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
displayFlag	enum	df_None df_Black df_HTML df_URL		This field specifies the nature of displayContent field. df_None means that there is no text message, and the content of the displayContent field is not applicable. This is used when there is only an audio message. df_Black means display a black screen, and the content of the displayContent field is not applicable. df_HTML means the content of the displayContent field is an HTML file. This option can be used if the size of the HTML file is not greater than 100 bytes. df_URL means that the content of the displayContent field is a URL location of the HTML file on the BFS server.
audioFlag	enum	af_None af_Silence af_URL		This field specifies the nature of the audioContent field. af_None means do not override the audio of the current service and the content of audioContent is not applicable. af_Silence means play silence over the current application and the content of audioContent is not applicable. af_URL means that the content of the audioContent field is the URL location of an audio file.
displayContent	string<>		100	This field specifies either the actual HTML MMM or the URL location of the MMM depending on the option given in the displayFlag field.
audioContent	string<>		100	This field specifies the URL location of the audio file if the option in audioFlag field is af_URL.
MultiLingualTitle		Sent in the DefinePpvEvent transaction to provide multiple title strings, each encoded in a separate language.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
multiLingualTitleCount	unsigned integer			This field provides the number of iterations of the following fields.
iso639LanguageCode	string<>		3	A given system is configurable to a specific set of supported languages. If this field is set to a non-supported language, the transaction will be rejected.
eventTitle	string<>		24	
OrderSource		Sent in the AddRppvAuthorization transaction to identify the originator of the authorization.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
orderSource	unsigned integer			A numerical identifier which will be returned with the PowerKEY PPV report.
PackageAuthorization		Used to inform the CAM of authorizations for given packages.		

<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
packageListType	enum	PLT_AllPackages PLT_SubscriptionOnly PLT_PpvOnly		
packageCount	unsigned integer			
packageNameList	string<>		20	This is an XDR array of packageNames.
PackageName		Sent to identify the subscription package for service being described by the current descriptor loop, or to identify a PPV event.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
packageName	string<>		20	
PayPerView		Sent in the DefinePpvEvent transaction to provide PPV event parameters that are common to both RPPV and IPPV events.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
rightToCopy	enum	RTC_CopyPermitted RTC_CopyProhibited		
PurchasePin		Sent to specify the new value of the IPPV purchase PIN.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
PurchasePinEnable	enum	PPE_Enabled PPE_Disabled		
PurchasePin	string[]		4	The pin may be any 4 numeric digits
RecordCount		Sent in the GetVodRecords transaction to identify the number of records to be retrieved.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
recordCount	unsigned integer	1..50		
Result		Sent to encapsulate the result of the corresponding descriptor loop in the transaction.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
resultCode	enum	See text		
ServiceName		Sent to identify the service being referenced by the current descriptorLoop.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
serviceName	string<>		20	
ShortDescription		Sent in service definition transactions to identify the service being defined.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
shortDescription	string<>		5	
SourceId		Sent in the DefineWatchTvService transaction to identify the ATSC-SI source carrying the content for the service being defined.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>

sourceId	integer	1..2 ¹⁶	20	
SourceUrl		Sent in the DefineVirtualChannelService to indicate the location of the data to be used with this Virtual Channel Service.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
sourceUrl	string[]		100	
StartRecordNumber		Sent in the GetVodRecords transaction to identify the first VOD purchase record to be retrieved		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
startRecordNumber	unsigned integer			
SubscriptionServiceName		Sent in the DefinePpvService transaction to indicate a service to utilize to access an event by subscription.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
serviceName	string<>		20	
Text		Sent to provide the text for a short HTML message to be used with the emergency alert system.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
text	string<>		1000	
UrgencyType		Sent in the MMMConfig transaction to determine the urgency of a MMM configuration.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
priority	unsigned short	1..10		This field determines the priority of MMM.
UploadFileSize		Sent in response to an InitializeVodUpload transaction to specify the number of records in the upload file created.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
uploadFileSize	unsigned integer			number of records
VodArchivePath		Sent in response to a VodUploadComplete transaction to specify the full path name of the VOD archive file created.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
vodArchivePath	string<>		256	Full path name of VOD Archive file.
VodPurchase		Sent in the response to a GetVodRecords transaction to describe the characteristics of a VOD purchase.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
assetOrganization	string<>		20	
assetId	string<>		20	
title	string<>		256	
purchaseTime	unsigned integer			UNIX time
purchaseLength	unsigned integer			seconds

listPrice	unsigned integer			Units for the listPrice should be worked out between the VOD app vendor and the SMS.
purchaseRightToCopy	enum	PRTC_CopyPermitted PRTC_CopyProhibited		
VodUploadFile		Sent in an InitializeVodUpload transaction to specify the name of the VOD upload file to be created.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
vodUploadFile	string<>		80	VOD Upload file name.
VodUploadPath		Sent in response to an InitializeVodUpload transaction to specify the full path name of the VOD upload file created.		
<i>Field</i>	<i>Type</i>	<i>Range</i>	<i>Length</i>	<i>Comments</i>
vodUploadPath	string<>		256	VOD Upload file name.

4.2 BASS Interface Descriptor Definitions

This section provides a more detailed explanation of the descriptors in Table 4-1.

4.2.1 AdvertisingWindow

The AdvertisingWindow descriptor controls the interval over which an IPPV event is advertised for purchase at the PPV client.

4.2.2 AlertType

The AlertType descriptor is set in the DefineMMMConfiguration transaction to define the type of notification to be sent to the user.

4.2.3 BillingId

The BillingId is used to identify a DHCT's "owning" billing system in a configuration where multiple billing systems are being used with a single DNCS. The BillingId can be used to limit the contents of an VOD upload file to only purchases that were made by DHCTs "owned" by a particular billing system.

4.2.4 BlockingPin

The Blocking Pin is used by the home-user of the S-A Resident Application to defeat any parental control applied to objectionable channels. The BlockingPin descriptor provides a method for AGs to control whether a Blocking PIN is required at a particular DHCT, and, if required, to set its value.

4.2.5 DatabaseOperation

The DatabaseOperation descriptor is sent in definitional BASS transactions to provide a finer granularity of control over the storage of the definition. When the DatabaseOperation descriptor is omitted from a definitional transaction, the Application Server that processes the transaction will insert a new definition if a matching definition cannot be found, or update an existing, matching definition.

When this descriptor is included in a transaction, a transaction marked `do_InsertOnly` will fail if an existing, matching definition is found. A transaction marked `do_UpdateOnly` will fail if no existing definition is found.

4.2.6 DefaultAdvertisingWindow

The `DefaultAdvertisingWindow` descriptor is sent in the `DefinePpvService` transaction to provide a rule by which the PPV application server may calculate start and end times for this per-event window. It operates by specifying offset intervals from other events. If the optional `AdvertisingWindow` descriptor is included on the `DefinePpvEvent` transaction, the values specified in that descriptor will be used in preference to those calculated from this descriptor, thus providing an override mechanism. The intervals in this descriptor are signed quantities, with negative numbers representing an offset *prior to* the event anchor.

4.2.7 DefaultEventCost

The `DefaultEventCost` descriptor is sent in the `DefinePpvService` transaction to provide a default cost events defined against the service.

4.2.8 DefaultEventOrderPhoneNumber

The `DefaultEventOrderPhoneNumber` descriptor is sent in the `DefinePpvService` transaction to provide a default ordering telephone number for events defined against the PPV service, and a rule by which the PPV application server may calculate start and end times for availability of that number on a per-event basis. It operates by specifying offset intervals from other events. If the optional `EventOrderPhoneNumber` descriptor is included on the `DefinePpvEvent` transaction, the values specified in that descriptor will be used in preference to those calculated from this descriptor, thus providing an override mechanism. The interval in this descriptor is a signed quantity, with negative numbers representing an offset *prior to* the event anchor.

4.2.9 DefaultMarketingWindow

The `DefaultMarketingWindow` descriptor is sent in the `DefinePpvService` transaction to provide a rule by which the PPV application server may calculate start and end times for this per-event window. It operates by specifying offset intervals from other events. If the optional `MarketingWindow` descriptor is included on the `DefinePpvEvent` transaction, the values specified in that descriptor will be used in preference to those calculated from this descriptor, thus providing an override mechanism. The intervals in this descriptor are signed quantities, with negative numbers representing an offset *prior to* the event anchor.

4.2.10 DhctMacAddr

The `DhctMacAddr` descriptor is sent to identify the DHCT being described by the current descriptor loop.

4.2.11 DhctMacAddrList

The DhctMacAddrList descriptor is sent to identify the set of DHCTs being described by the current descriptor loop.

4.2.12 DhctMessage

The DhctMessage descriptor is sent to specify a DSM-CC pass through message to be sent to a collection of DHCTs, specified as either a list of individual DHCTs or as a message group.

4.2.13 DhctParameter

The DhctParameter descriptor is sent to set a configuration parameter for the DHCT platform or the S-A Resident Application. Due to the large number of such configuration parameters, a generic encoding mechanism is used in this descriptor. A complete list of parameters can be found in [10].

4.2.14 DisplayType

The DisplayType descriptor is sent in the DefineMMMConfiguration transaction to determine how to display a message.

4.2.15 EventInfo

The EventInfo descriptor provides user consumable information to the PPV server, on a per-event basis. This information includes the title of the event, and a text string containing the price of the event.

4.2.16 EventList

The EventList descriptor provides the capability to collect a set of previously defined PPV events into a single event, which is authorizable or purchasable as a unit. Each of the constituent events remains individually authorizable or purchasable.

4.2.17 EventOrderPhoneNumber

The EventOrderPhoneNumber descriptor provides a user-presentable telephone number for the home-user to call to purchase a Reservation PPV event. It also includes an indication of the time at which the phone number will become active.

4.2.18 EventUseService

The EventUseService descriptor is sent in the DefinePpvService transaction to indicate a service to utilize to access an event.

4.2.19 ForceTypeType

The ForceTuneType descriptor is sent in the DefineMMMConfiguration transaction to determine the force tune behavior of a message.

4.2.20 GroupName

The GroupName descriptor is sent to provide an identifier for a DHCT message group.

4.2.21 ImpulsePayPerView

The ImpulsePayPerView descriptor is sent to define the PowerKEY control parameters for an Impulse PPV event. It is passed unaltered to the DNCS PowerKEY Control Suite. The ImpulsePayPerView descriptor is completely defined in [2].

4.2.22 InterstitialServiceName

The InterstitialServiceName descriptor is sent in the DefinePpvService transaction to determine a service to utilize during the interstitial period between PPV events.

4.2.23 LimitedDuration

The LimitedDuration descriptor is sent in the DefinePpvEvent transaction to define the interval over which the event is to remain active. It is distinct from the modeLength fields in the ImpulsePayPerView descriptor in the following ways:

- The LimitedDuration descriptor controls the lifetime of the event in the system, while the modeLength controls the lifetime of an authorization of an IPPV event in the DHCT security element.
- The LimitedDuration descriptor controls authorization lifetime for Reservation PPV events.

4.2.24 Logo

The Logo descriptor is sent in service defining BASS transactions to provide a graphical icon for the service being defined.

4.2.25 LongDescription

The LongDescription descriptor is sent in service defining BASS transactions to provide a detailed textual description of the service being defined.

4.2.26 MarketingWindow

The MarketingWindow descriptor is sent in the DefinePpvEvent transaction to control the interval over which the event may be advance purchased. The MarketingWindow is distinct from the buyWindowInterval in the ImpulsePayPerView descriptor in that the buyWindow controls when the PowerKEY data is present in the network to allow an IPPV purchase to occur.

4.2.27 MessageTime

The MessageTime descriptor is sent in the DefineMMMConfiguration transaction to determine the time/length for a message to be displayed.

4.2.28 MessageType

The MessageType descriptor is sent in the DefineMMMConfiguration transaction to determine the content of a message.

4.2.29 MultiLingualTitle

The MultiLingualTitle descriptor is sent in the DefinePpvEvent transaction to provide multiple title strings, each in a separate language.

4.2.30 OrderSource

The OrderSource descriptor is sent in the AddRppvEventAuthorization transaction to identify the source of such authorizations. It is returned from the PowerKEY IppvManager with records of the PPV authorization.

4.2.31 PackageAuthorization

The PackageAuthorization descriptor is used to inform the PPV Server of authorizations for given PPV events. When used for this purpose, the PackageAuthorization descriptor specifies a replacement string of authorizations.

The packageListType field is used to indicate whether the PackageAuthorization contains only subscription packages, only PPV packages, or both. For this use, the packageListType field should indicate PPV packages.

4.2.32 PackageName

The PackageName descriptor is sent to provide an authorization name for services sold on a subscription basis or a PPV basis. The presence of this descriptor typically also controls whether the content for a service is encrypted.

4.2.33 PayPerView

The PayPerView descriptor is sent in the DefinePpvEvent transaction to provide parameters that are common to both RPPV and IPPV events.

4.2.34 PurchasePin

The PurchasePin descriptor provides AGs with the ability to control whether entry of a PIN is required to purchase IPPV events, and, if a PIN is required, to control its value.

4.2.35 RecordCount

The RecordCount descriptor is sent in the *GetVODRecords* transaction to identify the number of to be retrieved.

4.2.36 Result

The Result descriptor is sent to encapsulate the result of the corresponding descriptor loop in the transaction.

The following table is an exhaustive list of valid resultCodes:

Description	Applicable Transactions
No Error	All transactions
DhctMacAddr not defined	SetPin, SetDhctParameter, AddRppvAuthorization, RemoveRppvAuthorization, AddDhctToGroup, RemoveDhctFromGroup, SendSamMessage, SendMessage, SendEmergencyAlert
EventUseService not defined	DefinePpvService
Existing Service of Wrong Type	DefineWatchTvService, DefineMusicService, DefineVirtualChannelService, DefinePpvService
Group has no members	SendSamMessage, SendMessage, SendEmergencyAlert
Group not Defined	AddDhctToGroup, RemoveDhctFromGroup, RetireMessageGroup, SendSamMessage, SendMessage, SendEmergencyAlert
InterstitialService not defined	DefinePpvService
Invalid AdvertisingWindow	DefinePpvEvent
Invalid BlockingPin	SetPin
Invalid DatabaseOperation	DefineWatchTvService, DefineMusicService, DefineVirtualChannelService, DefinePpvService, DefinePpvEvent
Invalid DefaultAdvertisingWindow	DefinePpvService
Invalid DefaultEventOrderPhoneNumber	DefinePpvService
Invalid DefaultMarketingWindow	DefinePpvService

Description	Applicable Transactions
Invalid DhctMacAddr	SetPin, SetDhctParameter, AddRppvAuthorization, RemoveRppvAuthorization, AddDhctToGroup, RemoveDhctFromGroup, SendSamMessage, SendMessage, SendEmergencyAlert
Invalid EventInfo	DefinePpvEvent
Invalid EventList	DefinePpvEvent
Invalid EventOrderPhoneNumber	DefinePpvEvent
Invalid EventUseServiceName	DefinePpvEvent
Invalid ImpulsePayPerView	DefinePpvEvent
Invalid InterstitialServiceName	DefinePpvService
Invalid LimitedDuration	DefinePpvEvent
Invalid Logo	DefineWatchTvService, DefineMusicService, DefineVirtualChannelService, DefinePpvService
Invalid MarketingWindow	DefinePpvEvent
Invalid PackageAuthorization	AddRppvAuthorization
Invalid PackageName	DefineWatchTvService, DefineMusicService, DefineVirtualChannelService, DefinePpvEvent, RetirePpvEvent, AddRppvAuthorization, RemoveRppvAuthorization
Invalid PayPerView	DefinePpvEvent
Invalid PurchasePin	SetPin
Invalid ServiceId	DefineWatchTvService, RetireWatchTvService, DefineVirtualChannelService, RetireVirtualChannelService
Invalid ShortDescription	RetirePpvService, DefinePpvEvent
Invalid SubscriptionServiceName	DefinePpvService
Package not defined	RetirePpvEvent, AddRppvAuthorization, RemoveRppvAuthorization

Description	Applicable Transactions
Service not defined	RetireWatchTvService, RetireMusicService, DefineVirtualChannelService, RetireVirtualChannelService, RetirePpvService, DefinePpvEvent, SendSamMessage
Service Not PPV	DefinePpvEvent
SourceUrl not accessible	DefineVirtualChannelService
Start Time has passed	DefinePpvEvent
Subscription Service not defined	DefinePpvService
Invalid MultLingualTitle	DefinePpvEvent
Unsupported Language	DefinePpvEvent
Missing required descriptor	All transactions
Unspecified error, request parsed correctly	All transactions
Unspecified error	All transactions

4.2.37 ServiceName

The ServiceName descriptor is sent in service defining transactions to identify the service being referenced by the current descriptorLoop

4.2.38 ShortDescription

The ShortDescription descriptor is sent in service defining transactions to provide a short, user presentable description of the service. This field is required to be unique throughout the namespace of applications.

4.2.39 SourceId

The SourceId descriptor is sent to connect a source-managing service, such as a WatchTv service to a content stream.

4.2.40 SourceUrl

The SourceUrl descriptor is sent in the DefineVirtualChannelService transaction to connect the service with the file containing the HTML text it processes.

4.2.41 StartRecordNumber

The StartRecordNumber descriptor is sent in the GetVodRecords transaction to identify the first VOD purchase record to be retrieved.

4.2.42 SubscriptionServiceName

The SubscriptionServiceName descriptor is sent in the DefinePpvService transaction to indicate a service to utilize to access an event by subscription.

4.2.43 Text

The Text descriptor is sent in the SendEmergencyAlert transaction to provide the textual message displayed to the user.

4.2.44 UrgencyType

The UrgencyType descriptor is sent in the DefineMMMConfiguratio transaction to determine the urgency of a MMM message.

4.2.45 UploadFileSize

The UploadFileSize is sent in response to an *InitializeVodUpload* transaction to specify the number of records in the upload file created.

4.2.46 VodArchivePath

The VodArchivePath is sent in response to an VodUploadComplete transaction to specify the full path name of the VOD Archive file created.

4.2.47 VodPurchase

The VodPurchase descriptor is sent in the response to a GetVodRecords transaction to describe the characteristics of a VOD purchase.

4.2.48 VodUploadFile

The VodUploadFile is sent in an InitializeVodUpload transaction to specify the name of the VOD Upload file to be created.

4.2.49 VodUploadPath

The VodUploadPath is sent in response to an InitializeVodUpload transaction to specify the full path name of the VOD Upload file created.

5. BASS Interface Transactions

5.1 BASS Transaction Summary

Table 5-1 provides a listing of BASS transactions with a short description of each transaction. The table includes a list of valid descriptors for each transaction. Some transactions do not make use of descriptors and are so noted.

Table 4-1 provides the structural details of all BASS descriptors.

Table 5-1: BASS Transaction Summary

WatchTv Service Transactions			
DefineWatchTvService (Apportioned)		This transaction is used to define normal TV-channel style services.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	ServiceName	M	The name for the new service. This is the SAM key for the service, and so must be unique within the system. If a watchTv service with this name already exists, it will be updated to match this transaction. If a non-watchTv service exists with this name, the transaction will be rejected.
	ShortDescription	M	A short, user-presentable description of the new service.
	LongDescription	M	A detailed description of the new service.
	Logo	O	A graphical representation of the new service.
	SourceId	M	This descriptor contains the id of the ATSC source to be managed by the service being defined.
	PackageName	M/O	The name of the subscription package for the WatchTvService being defined. If this descriptor is omitted, the WatchTv service will be sent in the clear.
	DatabaseOperation	O	This descriptor is used by user interfaces that need to strictly control the appearance of a human presentable representation of the server's state. Administrative Gateways (such as cable billing systems) are not expected to use this descriptor.
Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • Existing Service of Wrong Type • Invalid PackageName • Invalid Logo • Invalid DatabaseOperation • Missing required descriptor • Unspecified error, request parsed correctly • Unspecified error (Need to define more of these.)

RetireWatchTvService (Apportioned)		This transaction is used to retire previously defined watchTv services.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	ServiceName	M	The name of the service being retired.
Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • Service not defined • Missing required descriptor • Unspecified error, request parsed correctly • Unspecified error

Music Service Transactions

DefineMusicService (Apportioned)		This transaction is used to define a service that carries music programming.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	ServiceName	M	The name for the new service. This is the SAM key for the service, and so must be unique within the system. If a music service with this name already exists, it will be updated to match this transaction. If a non-music service exists with this name, the transaction will be rejected.
	ShortDescription	M	A short, user-presentable description of the new service.
	LongDescription	M	A detailed description of the new service.
	Logo	O	A graphical representation of the new service.
	SourceId	M	This descriptor contains the id of the ATSC source to be managed by the service being defined.
	PackageName	M/O	The name of the subscription package for the Music service being defined. If this descriptor is omitted, the WatchTv service will be sent in the clear.
	DatabaseOperation	O	This descriptor is used by user interfaces that need to strictly control the appearance of a human presentable representation of the server's state. Administrative Gateways (such as cable billing systems) are not expected to use this descriptor.
Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • Invalid PackageName • Existing Service of Wrong Type • Invalid Logo • Invalid DatabaseOperation • Missing required descriptor • Unspecified error, request parsed correctly • Unspecified error (Need to define more of these.)

RetireMusicService (Apportioned)		This transaction is used to retire previously defined Music services.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	ServiceName	M	The name of the service being retired.
Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • Service not defined • Missing required descriptor • Unspecified error, request parsed correctly • Unspecified error

Virtual Channel Service Transactions

DefineVirtualChannelService		This transaction is used to define HTML-based Virtual Channel services.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	ServiceName	M	The name for the new service. This is the SAM key for the service, and so must be unique within the system. If a virtualChannel service with this name already exists, it will be updated to match this transaction. If a non-virtualChannel service exists with this name, the transaction will be rejected.
	ShortDescription	M	This descriptor provides a short, user-presentable description of the service being defined.
	LongDescription	M	This descriptor provides a long description of the service being defined.
	Logo	O	Logo for the service being defined.
	SourceUrl	M	This descriptor provides a pointer to the resource location of the Virtual Channel data to be placed on the BFS.
	PackageName	O	The name of the entitlement for the service being defined. If this descriptor is not included, the new Virtual Channel service will be sent in the clear.
	DatabaseOperation	O	This descriptor is used by user interfaces that need to strictly control the appearance of a human presentable representation of the server's state. Administrative Gateways (such as cable billing systems) are not expected to use this descriptor.
Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • Existing Service of Wrong Type • SourceUrl not accessible • Invalid Logo • Invalid PackageName • Invalid DatabaseOperation • Missing required descriptor • Unspecified error, request parsed correctly

			<ul style="list-style-type: none"> • Unspecified error
RetireVirtualChannelService		This transaction is used to retire previously defined Virtual Channel services.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	ServiceName	M	The name of the service being retired.
Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • Service not defined • Missing required descriptor • Unspecified error, request parsed correctly • Unspecified error
DHCT Configuration Transactions			
SetPin		This transaction is used to define new values for the channel blocking PIN or IPPV purchase PIN.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	DhctMacAddr	M	The DHCT whose PINs are being set.
	BlockingPin	O	
	PurchasePin	O	
Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • Invalid DhctMacAddr • DhctMacAddr not defined • Invalid BlockingPin • Invalid PurchasePin • Missing required descriptor • Unspecified error, request parsed correctly • Unspecified error
ResetClientNvm		This transaction is used to command application clients resident on the indicated DHCTs to reset any allocated non-volatile memory to its factory default values.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	DhctMacAddr	M	
Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • Invalid DhctMacAddr • DhctMacAddr not defined • Missing required descriptor • Unspecified error, request parsed correctly • Unspecified error
SetDhctParameter		This transaction is used to set the value of a DHCT configuration parameter.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	DhctMacAddr	M	The DHCT to be configured.

	DhctParameter	M	The configuration parameter to set.
Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • Invalid DhctMacAddr • DhctMacAddr not defined • Missing required descriptor • Unspecified error, request parsed correctly • Unspecified error

Pay Per View Transactions

DefinePpvService		This transaction is used to request a new PPV service.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	ServiceName	M	The name for the new PPV service. This is the SAM key for the service, and so must be unique within the system. If a PPV service with this name already exists, it will be updated to match this transaction. If a non-PPV service exists with this name, the transaction will be rejected.
	ShortDescription	M	This descriptor provides a short, user-presentable description of the service being defined
	LongDescription	M	This descriptor provides a long description of the service defined.
	Logo	O	Logo for the service being defined.
	InterstitialServiceName	O	
	SubscriptionServiceName	O	
	EventUseServiceName	M	The name of the service to be used to consume the PPV event.
	DefaultAdvertisingWindow	M	This descriptor is used to specify a rule by which the PPV application server can calculate values for the start and end time of the advertising window for an event when the AdvertisingWindow descriptor is not included on the DefinePpvEvent transaction.
	DefaultMarketingWindow	M	This descriptor is used to specify a rule by which the PPV application server can calculate values for the start and end time of the marketing window for an event when the MarketingWindow descriptor is not included on the DefinePpvEvent transaction.
	DefaultEventOrderPhoneNu mber	M	This descriptor is used to specify a default ordering telephone number for events defined against the service.
	DefaultEventCost	O	This descriptor is used to specify a default cost string for events defined against this service. If specified, it is used only when the eventCost string in the EventInfo descriptor is set to the ASCII null character ('\0').
	DatabaseOperation	O	This descriptor is used by user interfaces that need to

			strictly control the appearance of a human presentable representation of the server's state. Administrative Gateways (such as cable billing systems) are not expected to use this descriptor.
Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • Existing Service of Wrong Type • InterstitialService not defined • SubscriptionService not defined • EventuseService not defined • Invalid DefaultAdvertisingWindow • Invalid DefaultMarketingWindow • Invalid DefaultEventOrderPhoneNumber • Invalid Logo • Invalid DatabaseOperation • Missing required descriptor • Unspecified error, request parsed correctly • Unspecified error
RetirePpvService		This transaction is used to retire previously defined PPV services.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	ServiceName	M	The name of the service to be retired.
Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • Service not defined • Missing required descriptor • Unspecified error, request parsed correctly • Unspecified error
DefinePpvEvent		This transaction is used to define PPV events to the PPV application server.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	PackageName	M	The name of the event being defined
	ServiceName	M	The name of the PPV service providing the event being defined.
	EventList	O	PPV events may be defined in terms of other, previously defined PPV events. Each constituent event of a compound event may still be individually authorized, or the compound event may be authorized as a unit. Individual constituent events may continue to be extended (by a new DefinePpvEvent transaction). If such an extension exceeds the life of the compound event, the compound event will be extended as well. It is never necessary to extend a compound event.
	PayPerView	M/O	Omission of this descriptor indicates that the event is authorized as an IPPV package only. Presence of this descriptor indicates that authorization for an event may be purchased as a reservation PPV event. This descriptor is

			mandatory in the absence of the ImpulsePayPerView descriptor, and is optional but allowed in its presence.
	ImpulsePayPerView	M/O	Omission of this descriptor indicates that the event is not purchasable via the DHCT. This descriptor is mandatory in the absence of the PayPerView descriptor, and is optional but allowed in its presence.
	AdvertisingWindow	O	Used to control when a PPV event should be advertised on the host service for purchase to the users. This descriptor is optional, such that if it is omitted, the advertising window for the event will be calculated based on the mandatory DefaultAdvertisingWindow descriptor defined with the event's specified PPV service.
	MarketingWindow	O	Used to control the interval over which a PPV event is advance purchasable. This descriptor is optional, such that if it is omitted, the marketing window for the event will be calculated based on the mandatory DefaultMarketingWindow descriptor defined with the event's specified PPV service.
	LimitedDuration	M	Used to indicate the lifetime of the event.
	EventInfo	M	User-presentable PPV event information.
	MultiLanguageTitle	O	This descriptor is optional, such that if it is not included only English-language titles can be provided.
	EventOrderPhoneNumber	O	This descriptor is optional, such that if it is not included, the information specified in the DefaultEventOrderPhoneNumber descriptor will be used.
	DatabaseOperation	O	This descriptor is used by user interfaces that need to strictly control the appearance of a human presentable representation of the server's state. Administrative Gateways (such as cable billing systems) are not expected to use this descriptor.
Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • Invalid PackageName • Invalid Service Name • Invalid ServiceName • Service not defined • Service not PPV • Invalid EventList • Invalid PayPerView • Invalid ImpulsePayPerView • PPV Event Window Error • Invalid AdvertisingWindow • Invalid MarketingWindow • Invalid LimitedDuration • Start Time has passed • Invalid EventInfo • Invalid EventOrderPhoneNumber • Invalid DatabaseOperation

			<ul style="list-style-type: none"> • Missing required descriptor • Invalid MultiLingualTitle • Unsupported Language • Unspecified error, request parsed correctly • Unspecified error
RetirePpvEvent		This transaction is used to retire previously defined events.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	PackageName	M	
Response	Result	M	<p>Can produce the following result codes:</p> <ul style="list-style-type: none"> • No error • Package not defined • Missing required descriptor • Unspecified error, request parsed correctly • Unspecified error
AddRppvAuthorization		This transaction is used to authorize a reservation PPV event on a specific DHCT.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	DhctMacAddr	M	The DHCT to be authorized.
	PackageAuthorization	M	The packages (RPPV events) to authorize.
	OrderSource	O	This descriptor identifies the source of the authorization. If this descriptor is included, the RPPV authorization will be reported to the DNCS IPPV manager for reporting to the billing system. If this descriptor is not included, the authorization will not be reported.
Response	Result	M	<p>Can produce the following result codes:</p> <ul style="list-style-type: none"> • No error • Invalid DhctMacAddr • Invalid PackageAuthorization • DhctMacAddr not defined • Package not defined • Missing required descriptor • Unspecified error, request parsed correctly • Unspecified error
RemoveRppvAuthorization		This transaction is used to remove an authorization for a reservation PPV event on a specific DHCT.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	DhctMacAddr	M	The DHCT to be deauthorized.
	PackageName	M	The package to deauthorize.
Response	Result	M	<p>Can produce the following result codes:</p> <ul style="list-style-type: none"> • No error • Invalid DhctMacAddr • DhctMacAddr not defined • Package not defined • Missing required descriptor

			<ul style="list-style-type: none"> • Unspecified error, request parsed correctly • Unspecified error
--	--	--	--

Messaging Transactions

DefineMessageGroup (Apportioned)		This transaction is used to define a group of DHCTs that can be addressed collectively by a single message transmission.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	GroupName	M	The name of the group being defined.
	DhctMacAddrList	O	An optional list of DHCTs to define the membership of the group. If a DatabaseOperation descriptor is included and set to update, this list will be treated as an add to the existing membership list.
	DatabaseOperation	O	If this descriptor is omitted, the group will be inserted if not previously defined, or updated if previously defined.
Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • Group not defined • Invalid DhctMacAddr • DhctMacAddr not defined • Invalid DatabaseOperation • Missing required descriptor • Unspecified error, request parsed correctly • Unspecified error
AddDhctToGroup (Apportioned)		This transaction is used to add a list of DHCTs to the specified group's membership.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	GroupName	M	The name of the group being modified.
	DhctMacAddrList	M	The list of DHCTs to be added to the group.
Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • Group not defined • Invalid DhctMacAddr • DhctMacAddr not defined • Missing required descriptor • Unspecified error, request parsed correctly • Unspecified error
RemoveDhctFromGroup (Apportioned)		This transaction is used to remove a list of DHCTs from the specified group's membership.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	GroupName	M	The name of the group being modified.
	DhctMacAddrList	M	The list of DHCTs to be removed from the group.

Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • Group not defined • Invalid DhctMacAddr • DhctMacAddr not defined • Missing required descriptor • Unspecified error, request parsed correctly • Unspecified error
RetireMessageGroup (Apportioned)		This transaction is used to eliminate a messaging.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	GroupName	M	The name of the group being retired.
Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • Group not defined • Missing required descriptor • Unspecified error, request parsed correctly • Unspecified error
SendSamMessage (Apportioned)		This transaction is used to cause the activation of the specified service on the addressed DHCTs.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	DhctMacAddrList	M/O	The individual DHCTs to receive the message. If this descriptor is omitted, the GroupName descriptor must be included.
	GroupName	M/O	The pre-defined group of DHCTs to receive the message. If this descriptor is omitted, the DhctMacAddrList descriptor must be included.
	ServiceName	M	The service to activate on the DHCTs.
Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • Group not defined • Group has no members • Invalid DhctMacAddr • DhctMacAddr not defined • Service not defined • Missing required descriptor • Unspecified error, request parsed correctly • Unspecified error
SendMessage (Apportioned)		This transaction is used to send a message to one or more DHCTs, or a DHCT group.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	DhctMacAddrList	M/O	The individual DHCTs to receive the message. If this descriptor is omitted, the GroupName descriptor must be included.

	GroupName	M/O	The pre-defined group of DHCTs to receive the message. If this descriptor is omitted, the DhctMacAddrList descriptor must be included.
	DhctMessage	M	The message to be sent to the DHCTs.
Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • Group not defined • Group has no members • Invalid DhctMacAddr • DhctMacAddr not defined • Missing required descriptor • Unspecified error, request parsed correctly • Unspecified error

Multi-Media Messaging Transactions

SendMultiMediaMessage (Apportioned)		This transaction is used to send a MMM to one or more DHCTs, to a group of DHCTs, or to the entire network.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	DhctMacAddrList	M/O	The individual DHCTs to receive the message. If this descriptor is omitted, the GroupName descriptor may be included. If both are omitted, the alert will be broadcast throughout the network.
	GroupName	M/O	The pre-defined group of DHCTs to receive the message. If this descriptor is omitted, the DhctMacAddrList descriptor may be included. If both are omitted, the alert will be broadcast throughout the network.
	MMMConfigName	M	This field specifies to the EAS server which configuration should be send with this on screen message.
	MessageType	M	This field specifies the detail information on the message content, both visual and audio.
Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • Invalid DhctMacAddr • DhctMacAddr not defined • GroupName not defined • Group has no members • Service not defined • Missing required descriptor • Unspecified error, request parsed correctly • Unspecified error
DefineMultiMediaMessageConfiguration (Apportioned)		This transaction is used to define configurations used with MMMs.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	ShortDescription	M	This descriptor determines short name of this

			configuration.
	UrgencyType	M	This descriptor determines the priority of a message when using this configuration.
	AlertType	M	This descriptor determines the notification methods to the users when an on screen message is arrived.
	DisplayType	M	This descriptor determines how the message is displayed and navigated.
	MessageTime	M	This descriptor determines detail information on the time/length of message.
Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • Missing required descriptor • Unspecified error, request parsed correctly • Unspecified error

RetireMultiMediaMessageConfiguration (Apportioned)

This transaction is used to retire a MMM configuration.

Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	ShortDescription	M	This descriptor determines short name of this configuration.
Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • No such MMM Configuration • Named MMM Configuration cannot be retired • Missing required descriptor • Unspecified error, request parsed correctly • Unspecified error

VOD Service Transactions

DefineVodService (Apportioned)

This transaction is used to request a new VOD service.

Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	ServiceName	M	The name for the new PPV service. This is the SAM key for the service, and so must be unique within the system. If a VOD service with this name already exists, it will be updated to match this transaction. If a non-VOD service exists with this name, the transaction will be rejected.
	ShortDescription	M	This descriptor provides a short, user-presentable description of the service being defined
	LongDescription	M	This descriptor provides a long description of the service defined.
	Logo	O	Logo for the service being defined.
	DatabaseOperation	O	This descriptor is used by user interfaces that need to strictly control the appearance of a human presentable

			representation of the server's state. Administrative Gateways (such as cable billing systems) are not expected to use this descriptor.
Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • Existing Service of Wrong Type • Invalid Logo • Invalid DatabaseOperation • Missing required descriptor • Unspecified error, request parsed correctly • Unspecified error
RetireVodService (Apportioned)		This transaction is used to retire previously defined VOD services.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	ServiceName	M	The name of the service to be retired.
Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • Service not defined • Missing required descriptor • Unspecified error, request parsed correctly • Unspecified error
VOD Upload Transactions			
InitializeVodUpload		This transaction is used to request that a VOD Upload file be created.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	VodUploadFile	M	
	BillingId	O	If this descriptor is omitted, all VOD purchase records will be included in the VOD upload file.
Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • Invalid BillingId • File already exists • Missing required descriptor • Extraneous Descriptor present • Unspecified error, request parsed correctly • Unspecified error
	VodUploadPath	O	This descriptor will not be produced if the initialize Vod upload request fails.
	UploadFileSize	O	This descriptor will not be produced if the initialize VOD upload request fails.
GetVodRecords		This transaction is used to retrieve VOD purchase records from the VOD Application Server.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	VodUploadPath	M	

	StartRecordNumber	M	
	RecordCount	M	
Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • File not found • File read failure • Missing required descriptor • Extraneous Descriptor present • Unspecified error, request parsed correctly • Unspecified error
	DhctMacAddr	O	This descriptor will not be produced if the GetVodRecords request fails.
	BillingId	O	This descriptor will not be produced if the GetVodRecords request fails.
	VodPurchase	O	This descriptor will not be produced if the GetVodRecords request fails.
VodUploadComplete		This transaction is used to notify the VOD Application Server that a VOD Upload file has been successfully processed by the AG.	
Request	<i>Descriptor</i>	<i>M/O</i>	<i>Comments</i>
	VodUploadPath	M	
Response	Result	M	Can produce the following result codes: <ul style="list-style-type: none"> • No error • File not found • Missing required descriptor • Extraneous Descriptor present • Unspecified error, request parsed correctly • Unspecified error
	VodArchivePath	O	This descriptor will not be produced if the VOD upload complete request fails.

5.2 General Transaction Format

5.2.1 Request Format

BASS transaction requests use the request format specified in [2].

5.2.2 Simple Response Format

Simple BASS transaction responses use the simple response format specified in [2].

5.2.3 Complex Response Format

Complex BASS transaction responses use the complex response format specified in [2].

5.3 WatchTv Service Transactions

5.3.1 DefineWatchTvService

This transaction allows the creation of a basic television service, as described in [8].

The sources to be used by WatchTv services must be defined using the appropriate BOSS and DSM-CC transactions.

5.3.1.1 Request Structure

Multiple services can be specified, one per descriptorLoop.

5.3.1.2 Response Structure

Each *DefineWatchTvService* response will include:

- The messageId of the associated request
- An errorCount that represents the number of WatchTv services that were not processed successfully
- A result for each descriptor loop included in the request.

5.3.2 RetireWatchTvService

This transaction is used to retire previously defined WatchTv Services.

5.3.2.1 Request Structure

Multiple WatchTv services can be specified. The descriptorLoopCount identifies the number of WatchTv services to be retired.

5.3.2.2 Response Structure

In the response structure, the returned descriptor loops match positionally with those in the request structure.

5.4 Music Service Transactions

5.4.1 DefineMusicService

This transaction allows the creation of a service carrying music audio programming.

The sources to be used by Music services must be defined using the appropriate BOSS and DSM-CC transactions.

5.4.1.1 Request Structure

Multiple services can be specified, one per descriptorLoop.

5.4.1.2 Response Structure

Each *DefineMusicService* response will include:

- The messageId of the associated request
- An errorCount that represents the number of Music services that were not processed successfully
- A result for each descriptor loop included in the request.

5.4.2 RetireMusicService

This transaction is used to retire previously defined Music Services.

5.4.2.1 Request Structure

Multiple Music services can be specified. The descriptorLoopCount identifies the number of Music services to be retired.

5.4.2.2 Response Structure

In the response structure, the returned descriptor loops match positionally with those in the request structure.

5.5 Virtual Channel Service Transactions

5.5.1 DefineVirtualChannelService

This transaction is used to provision a Virtual Channel service, as described in [9].

The SourceUrl descriptor provides a pointer to the resource location of the HTML file to be presented to the user. The VirtualChannel server makes a BFS link to this resource, enabling the AG to alter the file at will.

The VirtualChannel server is capable of managing subscription-based conditional access for the services it manages. If it is desired to offer the service on a subscription basis, then the PackageName descriptor must be included. If the PackageName descriptor is not included, the content for the service will be transmitted in the clear.

5.5.1.1 Request Structure

Multiple VirtualChannel services can be specified. The descriptorLoopCount identifies the number of VirtualChannel services to be defined.

5.5.1.2 Response Structure

Each *DefineVirtualChannelService* response will include:

- The messageId of the associated request
- An errorCount that represents the number of VirtualChannel services that were not processed successfully
- A result for each descriptor loop included in the request.

5.5.2 RetireVirtualChannelService

This transaction is used to retire previously defined VirtualChannel Services.

5.5.2.1 Request Structure

Multiple VirtualChannel services can be specified. The descriptorLoopCount identifies the number of VirtualChannel services to be retired.

5.5.2.2 Response Structure

In the response structure, the returned descriptor loops match positionally with those in the request structure.

5.6 DHCT Configuration Transactions

The transactions in this section support functionality described in [10].

5.6.1 SetPin

The S-A Resident Application provides two Personal Identification Numbers (PINs) to provide a level of user identification. The Blocking Pin releases channels from parental control, and the Purchase Pin, used to provide a non-secure user authentication for Pay Per View purchases. Each of these PINs may be changed by the user at the DHCT, but this transaction provides the AG with the capability to change the PINs on the user's behalf.

5.6.1.1 Request Structure

Each descriptorLoop can set either PIN or both PINs for one DHCT.

5.6.1.2 Response Structure

In the response structure, the returned descriptor loops match positionally with those in the request structure.

5.6.2 ResetClientNvm

This transaction is used to command application clients resident on the indicated DHCTs to reset any allocated non-volatile memory to its factory default values.

5.6.2.1 Request Structure

Multiple DHCTs can be specified. The descriptorLoopCount identifies the number of DHCTs to be authorized.

5.6.2.2 Response Structure

Each *ResetClientNvm* response will include the messageId of the associated request, an errorCount that represents the number of DHCTs that were not processed successfully, and a result for each descriptor loop included in the request.

5.6.3 SetDhctParameter

This transaction provides a configuration capability for the DHCT platform and the S-A Resident Application.

5.6.3.1 Request Structure

Each descriptorLoop in the SetDhctParameter transaction can set one parameter for a single DHCT. This means that each loop may contain a single DhctMacAddr descriptor and a single DhctParameter descriptor.

5.6.3.2 Response Structure

In the response structure, the returned descriptor loops match positionally with those in the request structure.

5.7 Pay Per View Transactions

The transactions in this section support functionality described in [11].

5.7.1 DefinePpvService

The Phase 1.0 PPV server manages two classes of objects: PPV Services and PPV Events. A PPV service provides a framework for individual, time-limited events. When a PPV service is activated at the DHCT, the PPV Client determines which event is current from the PpvEvent definition.

The DefinePpvService transaction defines information common to all events on a single PPV Service. This information includes:

- An optional service to provide service during the interstitial period between events.
- A service to provide subscription-based access to the service.
- The SAM service identity information:
 - Service Name
 - Short Description
 - Long Description
 - An optional Logo

5.7.1.1 Request Structure

Multiple PPV Services may be defined, one per descriptorLoop.

5.7.1.2 Response Structure

In the response structure, the returned descriptor loops match positionally with those in the request structure. Each descriptor loop will contain a Result descriptor.

5.7.2 RetirePpvService

This transaction is used to retire previously defined PPV Services.

5.7.2.1 Request Structure

Multiple PPV services can be specified. The descriptorLoopCount identifies the number of PPV services to be retired.

5.7.2.2 Response Structure

In the response structure, the returned descriptor loops match positionally with those in the request structure. A RetirePpvService transaction will be failed if there are any active or pending events for the service. All such events must be retired prior retiring the service.

5.7.3 DefinePpvEvent

PPV Events are defined against PPV Services. Each event is provisioned with the following information:

- The service name of the PPV service against which the event is being defined.
- An EventUseServiceName, to allow the user to consume the event, when purchased.
- A telephone number to call to place a reservation order for the event.
- A PackageName, to provide a PowerKEY authorization basis for the event.
- PowerKEY control descriptors (described in [2]):

- A `PackageName`, to provide a `PowerKEY` authorization basis for the event.
- `PayPerView`
- `ImpulsePayPerView`
- A `Marketing Window`, to control the interval over which the event is offered for purchase.
- A `LimitedDuration` descriptor, to control the lifespan of the event
- An `EventInfo` descriptor, which provides user-presentable information about the event.

A second type of event can be defined that collects the singular events described above into a single authorizable event. This is done using the `EventList` descriptor.

5.7.3.1 Request Structure

Each `descriptorLoop` defines one PPV Event. The `descriptorLoopCount` identifies the number of PPV Events to be defined.

5.7.3.2 Response Structure

In the response structure, the returned descriptor loops match positionally with those in the request structure.

5.7.4 RetirePpvEvent

This transaction is used to retire previously defined PPV Events.

5.7.4.1 Request Structure

Multiple PPV Events can be specified. The `descriptorLoopCount` identifies the number of PPV Events to be retired.

5.7.4.2 Response Structure

In the response structure, the returned descriptor loops match positionally with those in the request structure.

5.7.5 AddRppvAuthorization

The *AddRppvAuthorization* transaction instructs the PPV Application Server to authorize a Reservation PPV event it manages on a DHCT.

5.7.5.1 Request Structure

Multiple DHCTs can be specified. The `descriptorLoopCount` identifies the number of DHCTs to be authorized.

5.7.5.2 Response Structure

Each *AddRppvAuthorization* response will include the `messageId` of the associated request, an `errorCount` that represents the number of DHCTs that were not processed successfully, and a result for each descriptor loop included in the request.

5.7.6 RemoveRppvAuthorization

The *RemoveRppvAuthorization* transaction instructs the PPV Application Server to remove an authorization from a DHCT for a Reservation PPV event it manages.

5.7.6.1 Request Structure

Each descriptorLoop can remove one authorization from one DHCT. This means that each descriptorLoop can contain one DhctMacAddr descriptor and one PackageName descriptor.

5.7.6.2 Response Structure

Each *RemoveRppvAuthorization* response will include the messageId of the associated request, an errorCount that represents the number of DHCTs that were not processed successfully, and a result for each descriptor loop included in the request.

5.8 Messaging Transactions

The transactions in this section provide the capability to send messages, both to individual DHCTs and to groups of DHCTs. They also provide the means to manage DHCT messaging groups.

5.8.1 DefineMessageGroup

This transaction is used to provision a DHCT messaging group. A messaging group provides the capability to send a single message addressed to several DHCTs.

The GroupName descriptor provides a name for the group. The DhctMacAddrList descriptor provides an optional means to populate the membership of a message group at the time of definition. If the *DefineMessageGroup* transaction is used with the DatabaseOperation descriptor in update mode, the DhctMacAddrList is treated in the same fashion as in the *AddDhctToGroup* transaction.

5.8.1.1 Request Structure

Multiple message groups can be specified. The descriptorLoopCount identifies the number of message groups to be defined.

5.8.1.2 Response Structure

Each *DefineMessageGroup* response will include the messageId of the associated request, an errorCount that represents the number of message groups that were not processed successfully, and a result for each descriptor loop included in the request.

5.8.2 AddDhctToGroup

This transaction is used to add DHCTs to the membership of a message group.

The GroupName descriptor specifies the group whose membership is being modified. The DhctMacAddrList descriptor specifies a list of DHCTs to add to the group. No error occurs if a DHCT in the DhctMacAddrList is already a member of the specified group.

5.8.2.1 Request Structure

Multiple message groups can be modified. The descriptorLoopCount identifies the number of message groups to be modified.

5.8.2.2 Response Structure

Each *AddDhctToGroup* response will include the *messageId* of the associated request, an *errorCount* that represents the number of message groups that were not modified successfully, and a result for each descriptor loop included in the request.

5.8.3 RemoveDhctFromGroup

This transaction is used to remove DHCTs from the membership of a message group.

The *GroupName* descriptor specifies the group whose membership is being modified. The *DhctMacAddrList* descriptor specifies a list of DHCTs to add to the group. No error occurs if a DHCT in the *DhctMacAddrList* is not already a member of the specified group.

5.8.3.1 Request Structure

Multiple message groups can be modified. The *descriptorLoopCount* identifies the number of message groups to be modified.

5.8.3.2 Response Structure

Each *RemoveDhctFromGroup* response will include the *messageId* of the associated request, an *errorCount* that represents the number of message groups that were not modified successfully, and a result for each descriptor loop included in the request.

5.8.4 RetireMessageGroup

This transaction is used to remove DHCTs from the membership of a message group.

The *GroupName* descriptor specifies the group whose membership is being modified. The *DhctMacAddrList* descriptor specifies a list of DHCTs to add to the group. No error occurs if a DHCT in the *DhctMacAddrList* is not already a member of the specified group.

5.8.4.1 Request Structure

Multiple message groups can be retired. The *descriptorLoopCount* identifies the number of message groups to be retired.

5.8.4.2 Response Structure

Each *RetireMessageGroup* response will include the *messageId* of the associated request, an *errorCount* that represents the number of message groups that were not retired successfully, and a result for each descriptor loop included in the request.

5.8.5 SendSamMessage

This transaction provides a remote activation facility for services, similar to the *rexec* command on UNIX systems. On receipt of this message the SAM client running on each addressed DHCT activates the service specified in the *ServiceName* descriptor.

5.8.5.1 Request Structure

Multiple services can be activated, one per descriptor loop. The service activated in each descriptor loop can be addressed to a different list of DHCTs or a different group.

5.8.5.2 Response Structure

Each *SendSamMessage* response will include the `messageId` of the associated request, an `errorCount` that represents the number of service activation messages that were not sent successfully, and a result for each descriptor loop included in the request.

To facilitate operation of this facility in a one-way network, the sent messages are not acknowledged. It is therefore possible that a properly sent message will not be received by a given addressed DHCT.

5.8.6 SendMessage

This transaction is used to send a DSM-CC passthrough message to a list of one or more DHCTs, or a previously defined message group of DHCTs.

The `GroupName` descriptor specifies the group to receive the message. The `DhctMacAddrList` descriptor specifies a list of individual DHCTs to receive the message. The `DhctMessage` descriptor provides the DSM-CC message type, the length, and the content bytes of the message. The *SendMessage* transaction imposes no requirements on the content of the message.

5.8.6.1 Request Structure

Multiple messages can be sent, one per descriptor loop. The message sent in each descriptor loop can be addressed to a different list of DHCTs or a different group.

5.8.6.2 Response Structure

Each *SendMessage* response will include the `messageId` of the associated request, an `errorCount` that represents the number of pass-through messages that were not sent successfully, and a result for each descriptor loop included in the request.

To facilitate operation of this facility in a one-way network, the sent messages are not acknowledged. It is therefore possible that a properly sent message will not be received by a given addressed DHCT.

5.9 Multi-Media Messaging Transactions

5.9.1 SendMultiMediaMessage

This transaction is used to send a multi-media message to one or more DHCTs, a group of DHCTs, or to the entire network.

5.9.1.1 Request Structure

Multiple messages can be sent, one per descriptor loop. The message sent in each descriptor loop can be addressed to a different list of DHCTs or a different group.

5.9.1.2 Response Structure

Each *SendMultiMediaMessage* response will include the `messageId` of the associated request, an `errorCount` that represents the number of multi-media messages that were not sent successfully, and a result for each descriptor loop included in the request.

To facilitate operation of this facility in a one-way network, the sent messages are not acknowledged. It is therefore possible that a properly sent message will not be received by a given addressed DHCT.

5.9.2 DefineMultiMediaMessageConfiguration

This transaction is used to define configurations used with multi-media messages.

5.9.2.1 Request Structure

Multiple message configurations can be specified. The descriptorLoopCount identifies the number of configurations to be defined.

5.9.2.2 Response Structure

Each *DefineMultiMediaMessageConfiguration* response will include the messageId of the associated request, an errorCount that represents the number of multi-media messages that were not processed successfully, and a result for each descriptor loop included in the request.

5.9.3 RetireMultiMediaMessageConfiguration

This transaction is used to retire a multi-media message configuration.

5.9.3.1 Request Structure

Multiple message configurations can be specified. The descriptorLoopCount identifies the number of configurations to be retired.

5.9.3.2 Response Structure

Each *RetireMultiMediaMessageConfiguration* response will include the messageId of the associated request, an errorCount that represents the number of multi-media messages that were not retired successfully, and a result for each descriptor loop included in the request.

5.10 VOD Service Transactions

5.10.1 DefineVodService

This transaction is used to provision a VOD service.

The sources to be used by VOD services must be defined using the appropriate BOSS and DSM-CC transactions.

5.10.1.1 Request Structure

Multiple VOD services can be specified. The descriptorLoopCount identifies the number of VOD services to be defined.

5.10.1.2 Response Structure

Each *DefineVodService* response will include:

- The messageId of the associated request
- An errorCount that represents the number of VOD services that were not processed successfully
- A result for each descriptor loop included in the request.

5.10.2 RetireVodService

This transaction is used to retire previously defined VOD Services.

5.10.2.1 Request Structure

Multiple VOD services can be specified. The descriptorLoopCount identifies the number of VOD services to be retired.

5.10.2.2 Response Structure

In the response structure, the returned descriptor loops match positionally with those in the request structure.

5.11 VOD Upload Transactions

5.11.1 InitializeVodUpload

This transaction is used to request that a VOD Upload file be created.

5.11.1.1 Request Structure

This transaction is constrained to accept only a single VOD file creation request.

5.11.1.2 Response Structure

Each *InitializeVodUpload* response will include a result code, an VOD Upload file path, and the number of VOD purchase records contained in the file. If the transaction fails, only the result code will be included.

5.11.2 VodUploadComplete

This transaction is used to notify the VOD Application Server that a VOD Upload file has been successfully processed by the AG.

5.11.2.1 Request Structure

This transaction is constrained to accept only a single VOD upload complete request.

5.11.2.2 Response Structure

The *VodUploadComplete* response is constrained to include only a single result descriptor.

5.11.3 GetVodRecords

This transaction is used to retrieve VOD purchase records from the VOD Application Server.

5.11.3.1 Request Structure

The *GetVodRecords* request contains starting record number and number of records to be retrieved.

5.11.3.2 Response Structure

In the response structure, the returned descriptor loops contain the records retrieved. If the end of file is encountered before the number of records requested is satisfied, the value of the descriptorLoopCount will indicate the number of records included in the response. If the transaction fails, only the result code will be included.

5.12 Usage Data Interface

Several application components can produce information describing the usage of some system resource. This section describes the retrieval interface that will apply to usage data supplied by any application component. Subsequent sections describe specific characteristics of each usage data interface, such as record formats.

In general, each application component that produces usage data produces a data file, according to its specific file format, which is retrieved by the AG using either File Transfer Protocol (FTP) or a BOSS "GetRecords" transaction. For a limited period, the file is retained by the application component for protection against accidental file corruption.

Each application component maintains two directories for the purposes of storing the data files and communicating with the AG. When the usage data file is produced, it will be placed into the primary. The AG is provided read-permission on the primary directory. When the AG acknowledges it has successfully retrieved the file, it is moved to an archive directory, and the new path to the file is provided to the AG. The file is maintained in this directory for a user-configurable period. The AG has read permission on the archive directory, and may retrieve files from that directory via either retrieval method.

All data will be represented using the eXternal Data Representation (XDR) standard. UNIX time will be used for all date/time data. Elapsed time will be provided in units of seconds. Bandwidth data will be provided in units of bits per second.

5.12.1.1 VOD Record Format

Data Element	Type	Range/Units	Length
dhctMacAddress	string[]		17
billingId	unsigned integer		
AssetOrganization	string<>		20
AssetId	string<>		20
title	string<>		256
purchaseTime	unsigned integer	UNIX time	
purchaseLength	unsigned integer	seconds	
listPrice	unsigned integer	Units determined between SMS and VOD app	
rightToCopy	enum	RTC_CopyPermitted RTC_CopyProhibited	