



INTERFACE CONTROL DOCUMENT FOR BROADSWORD VERSION 3.1



Prepared for:

AFC2ISRC/A26
Langley Air Force Base, VA 23665

Prepared by:

Air Force Research Laboratory, Rome Research Site
AFRL/IFEB
32 Brooks Road
Rome, NY 13441-4114

June 2002

CHAPTER 1 EXECUTIVE SUMMARY.....	4
1.1 Identification.....	4
1.2 System Overview	4
1.3 Document Organization	4
1.4 Definitions	5
1.5 Limitations and Restrictions	5
CHAPTER 2 APPLICABLE DOCUMENTS.....	6
2.1 Government documents	6
CHAPTER 3 SEGMENT INTERFACES	8
CHAPTER 4 BROADSWORD CLIENT GATEKEEPER API SPECIFICATION	10
4.1 Client to Gatekeeper Application Program Interface	10
4.1.1 Interface Identification	10
4.1.2 Interface Description.....	10
4.1.3 Functional Interface Specification.....	10
4.1.3.1 User Client to Gatekeeper Messages.....	11
4.1.3.2 Broadsword Gatekeeper to Client Messages.....	12
4.1.4 Physical Interface Specification.....	12
CHAPTER 5 BROADSWORD GATEKEEPER PLUG-IN API SPECIFICATION.....	14
5.1 Gatekeeper to Plug-in Application Program Interface.....	14
5.1.1 Interface Identification	14
5.1.2 Interface Description.....	14
5.1.3 Functional Interface Specification.....	14
5.1.3.1 Gatekeeper to Plug-in Messages.....	15
5.1.3.2 Plug-in to Gatekeeper Messages.....	15
5.1.4 Physical Interface Specification.....	16
CHAPTER 6 BROADSWORD ISSO TO GATEKEEPER API SPECIFICATION.....	18
6.1 ISSO to Gatekeeper Application Program Interface.....	18
6.1.1 Interface Identification	18
6.1.2 Interface Description.....	18
6.1.3 Functional Interface Specification.....	18
6.1.3.1 ISSO to Gatekeeper Messages.....	18
6.1.3.2 Gatekeeper to ISSO messages.....	19
6.1.4 Physical Interface Specification.....	19
CHAPTER 7 BROADSWORD SYS ADMIN TO GATEKEEPER API SPECIFICATION.....	20
7.1 SYS ADMIN to Gatekeeper Application Program Interface.....	20
7.1.1 Interface Identification	20
7.1.2 Interface Description.....	20
7.1.3 Functional Interface Specification.....	20
7.1.3.1 SYS ADMIN to Gatekeeper Messages.....	22
7.1.4 Physical Interface Specification.....	24
CHAPTER 8 BROADSWORD VARIABLE LENGTH MESSAGE FORMAT (BVLMF).....	28
8.1 Broadsword Variable Length Message Format Overview.....	28
8.2 BVLMF Basic Structure.....	28

8.3 BVLMF Message Construction and Parsing 29

APPENDIX I BROADSWORD GATEKEEPER API

APPENDIX II API FUNCTION DEFINITIONS, AND ERRORS

APPENDIX III BROADSWORD GATEKEEPER TO PLUG-IN API

APPENDIX IV BROADSWORD ISSO TO GATEKEEPER API

APPENDIX V API FUNCTION DEFINITIONS, ISSO AND ADMIN ERRORS

APPENDIX VI BROADSWORD ADMIN TO GATEKEEPER API

APPENDIX VII ADMIN API FUNCTIN DEFINITIONS

APPENDIX VIII MISCELLANEOUS

Chapter 1

Executive Summary

1.1 Identification

This Interface Control Document (ICD) establishes the requirements for the interface between the Broadsword Gatekeeper version 3.1 and client applications. This document was developed by AFRL/IFEB.

1.2 System Overview

The basic mission of the Gatekeeper is to act both as a broker to various backside sources (including IPL, 5D, MIDB, AMHS, and IESS among others) and the sole location for all security audits created by this access. To achieve this mission the Gatekeeper performs the following activities:

- a. Accepts a single login for all backside sources.
- b. Audits the login and logout of every client access.
- c. Audits all query requests plus every backside source response including all hits received.
- d. Audits all product requests including any alternate delivery path.
- e. Performs image product conversion, compression and interpolation.
- f. Performs all backside routing including remote Gatekeepers

1.3 Document Organization

- a. Section 1 Identifies Scope and General Overview.
- b. Section 2 Identifies the Applicable Documents.
- c. Section 3 Identifies the Interfaces covered by the ICD.
- d. Section 4 Provides the Segment Details for the Client to Gatekeeper Interface.
- e. Section 5 Provides a detailed description of Broadsword Variable Length Message Format (BVLMF).
- f. Appendix I Describes Broadsword Client to Gatekeeper Application Program Interface.
- g. Appendix II Describes the "C" Client to Gatekeeper API Library Functions.
- h. Appendix III Describes the Broadsword Gatekeeper Plug-in API
- i. Appendix IV Describes Broadsword Gatekeeper ISSO API
- j. Appendix V Describes the "C" ISSO to Gatekeeper API Library Functions
- k. Appendix VI Describes the Broadsword Gatekeeper Sys Admin API
- l. Appendix VII Describes the "C" Sys Admin to Gatekeeper API Library Functions

m. Appendix VIII Describes Default, Parameter Values and Definitions

1.4 Definitions

There are no special definitions applicable to this document.

1.5 Limitations and Restrictions

There are no special limitations or restrictions applicable to this document.

Chapter 2

Applicable Documents

2.1 Government documents

The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement. Copies of specifications, standards, drawings, and publications required by suppliers in connection with specified procurement functions should be obtained from the contracting agency, or as directed by the contracting officer.

SPECIFICATIONS

Document ID	Description
RFC N01-0044 N0101-B	Geospatial and Imagery Access Services Specification
	<i>5 February 1998</i>
	Broadsword INK document
	<i>June 2000</i>

STANDARDS

Document ID	Description
FIPS PUB 4-1	Federal Information Processing Standards Publication
	<i>27 January 1988</i>
OT-32796	Data Item Description System/System Segment
	<i>19 February 1988</i>
	Broadsword Interface Control Document (ICD)
	<i>June 2000</i>
MIL-STD-1777	Internet Protocol (IP)
	<i>12 August 1983</i>
	<i>26 October 1984 Notice 1</i>
MIL-STD-1778	Transmission Control Protocol (TCP)
	<i>12 August 1983</i>
	<i>26 October 1984 Notice 1</i>
MIL-STD-1780	File Transfer Protocol (FTP)
	<i>10 May 1984</i>
MIL-STD-2500A	National Imagery Transmission Format (Version 2.0)
	<i>12 October 1994</i>

This page left intentionally blank.

Chapter 3

Segment Interfaces

There are four interfaces controlled by this document.

- 1.) Client to Gatekeeper and Gatekeeper to Client interface
- 2.) Gatekeeper to Plug-in and Plug-in to Gatekeeper interface
- 3.) ISSO to Gatekeeper audit interface
- 4.) SYS ADMIN to Gatekeeper interface

This page left intentionally blank

Chapter 4

Broadsword Client Gatekeeper API Specification

4.1 Client to Gatekeeper Application Program Interface

The following subparagraphs describe the interface, which allows user client applications to communicate with the Gatekeeper.

4.1.1 Interface Identification

The interface is identified as the Broadsword Gatekeeper Application Program Interface.

4.1.2 Interface Description

This interface allows the users of a Broadsword Gatekeeper to write their own applications to interface with the Broadsword Gatekeeper.

4.1.3 Functional Interface Specification

This interface provides for the following capabilities for a user-written client application in interfacing with the Broadsword Gatekeeper. Appendix II describes the API functions.

- a. Perform Login to the Gatekeeper.
- b. Submit a query, geo search or keyword search to one or more backside sources.
- c. Request products, reports, messages from the backside sources.
- d. Inquire on the status of out-standing product requests.
- e. Catalog new products on selected backside sources.
- f. Inquire on the status of out-standing catalogs.
- g. Inquire on the most often queried sources or products accessed.
- h. Inquire on the status of the connection to the Gatekeeper.
- i. Initiate a stream or change filter parameters
- j. Terminate a stream
- k. Change a password
- l. Get site catalog elements
- m. Log off from the Gatekeeper.

4.1.3.1 User Client to Gatekeeper Messages

The following messages, as specified in Appendix I, will flow from the client to the Gatekeeper.

- a **Broadsword Gatekeeper Client Connect Message.** This message (BGCCM) allows the client to establish a TCP/IP socket connection and then Login to the Gatekeeper.
- b **Broadsword Gatekeeper Client Query Message.** This message (BGCQM) allows the client to query, geo search, and perform keyword searches to one or more backside sources.
- c **Broadsword Gatekeeper Client Product Request Message.** This message (BGCPRM) allows the client to request products, reports from a backside source.
- d **Broadsword Client Request for Product Delivery Status.** This message (BGCRFS) allows the client to inquire on the status of a requested product.
- e **Broadsword Client Catalog New Product.** This message (BGCCNP) allows the client to catalog a new product on the selected backside source.
- f **Broadsword Client Request for Product Catalog Status.** This message (BGCRFCS) allows the client to inquire on the status of a cataloged product.
- g **Broadsword Client Request for Top Ten Requests.** This message (BGCTTR) allows the client to inquire on the top number of requests initiated at this gatekeeper.
- h **Broadsword Client Verify Gatekeeper Connection.** This message BGKAM allows the client to check on the status of its connection to the gatekeeper.
- i **Broadsword Client Initiate Stream Request.** The (BGCISR) provides the client the capability to either initiate a stream or change a stream filter.
- j **Broadsword Client Terminate Stream Request.** The (BGCTSR) is used to terminate the connection to the stream server.
- k **Broadsword Client Change Password.** The (BGCPM) is used to change the password of current user.
- l **Broadsword Client Get Site Catalog Elements.** The (BGGSC) is used to retrieve the catalog elements and values from a source that supports cataloging.

4.1.3.2 Broadsword Gatekeeper to Client Messages.

The following message will flow from the Gatekeeper to the client.

- a. **Broadsword Gatekeeper Connect Response Message.** The (BGRM) message provides status and backside connectivity information to the client.
- b. **Broadsword Gatekeeper Query Response Message.** The (BGQRM) provides both status and query/search results generated by the BGCQM.
- c. **Broadsword Gatekeeper Product Request Acknowledgment.** The (BGPPRA) provides the status of either a product request or a request for status.
- d. **Broadsword Gatekeeper Catalog New Product Acknowledgment.** The (BGCNPA) provides the status of either a catalog product or a catalog product request for status.
- e. **Broadsword Gatekeeper Top Ten Requests Response Message.** The (BGCTTRM) provides both status and the top requests initiated at this gatekeeper to the client.
- f. **Broadsword Gatekeeper Column Attributes Response Message.** The (BGCARM) contains the column attribute information requested.
- g. **Broadsword Gatekeeper Initiate Stream Request Response.** The (BGISRR) contains the status of the initiate or change stream filter requested.
- h. **Broadsword Gatekeeper Terminate Stream Request Response.** The (BGTSRR) contains the status of the terminate stream request.
- i. **Broadsword Gatekeeper Change Password Response.** The (BGCPSM) contains the status of the change user password request.
- j. **Broadsword Gatekeeper Get Site Elements Response.** The (BGGSC) contains the source catalog information requested.

4.1.4 Physical Interface Specification.

The physical interface for all Client and Gatekeeper activity is any network supporting TCP/IP.

This page left intentionally blank

Chapter 5

Broadsword Gatekeeper Plug-in API Specification

5.1 Gatekeeper to Plug-in Application Program Interface

The following subparagraphs describe the interface between the Gatekeeper and the Plug-ins.

5.1.1 Interface Identification

The interface is identified as the Broadsword Gatekeeper to Plug-in Interface.

5.1.2 Interface Description

The Plug-in interface consists of two parts, a program “shell” complete with a makefile which will open a TCP/IP port, wait for a connection from the Gatekeeper and receive messages from the Gatekeeper, and the message definition between the Gatekeeper and the Plug-in.

The message interface is outlined in Appendix III to show the differences between the client interface and the plug-in interface. In most cases the Gatekeeper to plug-in message contains the same information as the client message but has additional tokens.

5.1.3 Functional Interface Specification

This interface provides for the following capabilities for the plug-ins:

- a) Open a port and listen for connections from the Gatekeeper.
- b) Receive a client query message (BGCQM) or a client request for product message (BGCPRM).
- c) Send an acknowledgement to the Gatekeeper when a valid BGCQM is received and processed.
- d) Send a query response message to the Gatekeeper (BGQRM).
- e) Send a Set Request Status message to the Gatekeeper FTP/STATUS daemon.
- f) Disconnect from the Gatekeeper.

This interface will not create the BGQRM or perform the necessary steps to extract the

product from the applicable backside source. These functions will be the responsibility of the developer. The Broadsword development library can be used to construct BVLMF messages according to this ICD. Sample plug-in sources will be delivered as examples and guides.

5.1.3.1 Gatekeeper to Plug-in Messages

The following messages, as specified in Appendix III, will flow from the Gatekeeper to the Plug-in.

- a. **Broadsword Gatekeeper Client Query Message.** This message (BGCQM) forwards the client query message parts particular to this backside source to query, geo search, and perform keyword searches on behalf of the client.
- b. **Broadsword Gatekeeper Client Product Request Message.** This message (BGCPRM) forwards the client request for products message parts particular for this backside source to perform the product delivery on behalf of the client.
- c. **Broadsword Gatekeeper Client Catalog New Product.** This message (BGCCNP) forwards the client catalog product request message to the applicable plug-in.
- d. **Broadsword Gatekeeper Client Column Attributes Message.** This message (BGCCAM) forwards the client column attributes message to the applicable plug-in.
- e. **Broadsword Gatekeeper Client Initiate Stream Message.** This message (BGCISM) forwards the client initiate stream message to the plug-in that supports data streaming.
- f. **Broadsword Gatekeeper Terminate Stream Message.** This message (BGCTSM) forwards the client terminate stream message to the plug-in that supports data streaming.

5.1.3.2 Plug-in to Gatekeeper Messages.

The following messages will flow from the Plug-in to the Gatekeeper.

- a. **Broadsword Gatekeeper Query Response Message.** The (BGQRM) provides both status and query/search results generated by the BGCQM for this particular source.
- b. **Plug-in to Gatekeeper Acknowledgement.** The (ACK) message provides the Gatekeeper with acknowledgement that the Plug-in received and processed the BGCPRM message.

- c. **Plug-in to Gatekeeper FTP/STATUS Daemon Status Update.** The (STS) message allows the Plug-in to both update product delivery status and to initiate product delivery. Product delivery can consist of image file conversion as well as FTP to the client destination(s). The Gatekeeper FTP/STATUS daemon will perform the status updates as well as FTP delivery.
- d. **Plug-in to Gatekeeper Column Attributes Response Message.** The (BGCARM) message returns the plug-in column attributes response message to the gatekeeper. The gatekeeper in turn forwards this message to the client.
- e. **Plug-in to Gatekeeper Generic Response Message.** The (BGGSR) message is used to return status and other source specific information. The content of the message is the STATUS sub-message plus other source specific and configurable elements. This message is currently used as the response to both the BGCISM and BGCTSM messages. The gatekeeper forwards the BGGSR message to the client.

5.1.4 Physical Interface Specification.

The physical interface for all Gatekeeper and Plug-in activity is any network supporting TCP/IP.

This page left intentionally blank

Chapter 6

Broadsword ISSO to Gatekeeper API Specification

6.1 ISSO to Gatekeeper Application Program Interface

The following subparagraphs describe the interface between the ISSO and the Gatekeeper.

6.1.1 Interface Identification

The interface is identified as the Broadsword ISSO to Gatekeeper Interface.

6.1.2 Interface Description

This interface is a message-based interface. The ISSO must first connect to the Gatekeeper using the BGCCM message outlined in the Client to Gatekeeper interface. The user must have ISSO privileges to use this interface. After a successful connect and lock to the Gatekeeper, the ISSO may get a audit report, extract audit records for archive, update the audit by deleting audit records and finally log out from the Gatekeeper.

6.1.3 Functional Interface Specification

This interface provides for the following capabilities for the ISSO: Appendix V describes the API functions.

- a. Create an audit report based on a date-time window, user-id, or event.
- b. Extract audit records into a flat file for archive.
- c. Update the audit database by removing archived records.
- d. Retrieve a list of previously created audit archives.
- e. Create an audit report from an audit archive file or files.

6.1.3.1 ISSO to Gatekeeper Messages

The following message, as specified in Appendix IV, will flow from the ISSO to the Gatekeeper.

- a. **Broadsword Gatekeeper Audit Detail Record Request.** This message (BAMSG), with function code AUDITR, will contain all the information so the Gatekeeper can create the Audit Report desired by the ISSO.
- b. **Broadsword Gatekeeper Audit Extract Data Request.** This message (BAMSG), with function code AUDITE, will contain all the information so the Gatekeeper can create the Audit Extract file desired by the ISSO.
- c. **Broadsword Gatekeeper Audit Update Data Request.** This message (BAMSG), with function code AUDITU, will contain all the information so the Gatekeeper can remove those audit records desired by the ISSO.
- d. **Broadsword Gatekeeper Audit Detail Record from Archive Request.** This message (BAMSG), with function code AUDITRA, will contain all the information so the Gatekeeper can create the Audit Report desired by the ISSO from the audit archive.
- e. **Broadsword Gatekeeper List Audit Archive.** This message (BAMSG), with function code ARCHIVE, will contain all the information so the Gatekeeper can create the List Audit Archive report for the ISSO.

6.1.3.2 Gatekeeper to ISSO messages

The following messages, as specified in Appendix IV, will flow from the Gatekeeper to the ISSO.

- a. **Broadsword Gatekeeper Audit Detail Record.** This message (BGADR) will contain the audit information for a audit report in date, user and session order.
- b. **Broadsword Gatekeeper Audit Extract Response.** This message (BGER) will contain information describing the Audit Extract request.
- c. **Broadsword Gatekeeper Audit Update Response.** This message (BGAUR) will contain information describing the Audit Update request.
- d. **Broadsword Gatekeeper Audit Archive Response.** This message (BGARSP) will contain information describing the Audit Archive List request.

6.1.4 Physical Interface Specification.

The physical interface for all Gatekeeper and ISSO activity is any network supporting TCP/IP.

Chapter 7

Broadsword SYS ADMIN to Gatekeeper API Specification

7.1 SYS ADMIN to Gatekeeper Application Program Interface

The following subparagraphs describe the interface between the SYS ADMIN and the Gatekeeper.

7.1.1 Interface Identification

The interface is identified as the Broadsword SYS ADMIN to Gatekeeper interface.

7.1.2 Interface Description

This interface is a message-based interface. The SYS ADMIN must first connect to the Gatekeeper using the BGCCM message outlined in the Client to Gatekeeper interface. The user must have SYS ADMIN privileges to use this interface. After a successful connect to the Gatekeeper, the SYS ADMIN must LOCK the Gatekeeper for SYS ADMIN usage. One and only One SYS ADMIN can LOCK the Gatekeeper in order to modify the Gatekeeper configuration. After a successful lock, the SYS ADMIN may inquire about the current configuration and modify the configuration. After the SYS ADMIN is finished, it will UNLOCK the Gatekeeper and finally log out from the Gatekeeper.

7.1.3 Functional Interface Specification

This interface provides for the following capabilities for the SYS ADMIN: Appendix VII describes the API functions.

- a. Lock the Gatekeeper for SYS ADMIN use.
- b. Unlock the Gatekeeper so other SYS ADMIN users may view or modify configuration.
- c. Get Plug-in names of all installed plug-ins.
- d. Get either the Gatekeeper template or Plug-in template.
- e. Get a list of all sources currently conTabled.

- f. Get configuration information for the Gatekeeper or local Source.
- g. Create a new backside source.
- h. Delete a backside source.
- i. Set or modify a current backside source parameter.
- j. Retrieve all current conTabled group names.
- k. Retrieve all members of a group.
- l. Add a new member to an existing group.
- m. Delete a member from an existing group.
- n. Delete a group.
- o. Add a new group.
- p. Retrieve the discretionary access control permissions for a particular gatekeeper or backside source.
- q. Add/Set discretionary access control permissions for a particular gatekeeper or backside source.
- r. Delete discretionary access control permissions for a particular gatekeeper or backside source.
- s. Retrieve the user discretionary access control permissions.
- t. Retrieve the valid users, producers, admin users or isso users for the local gatekeeper or a particular backside source.
- u. Add users as a valid producer to a particular backside source or add users as valid sys admins or issos to the local Gatekeeper.
- v. Delete users as valid producers to a particular backside source or from valid sys admins or issos to the local Gatekeeper.
- w. Modify a conTabled DATA_NAME's parameters
- x. Initialize a gatekeeper registration.
- y. Retrieve gatekeeper registration status.
- z. Register a gatekeeper with the Keymaster.
- aa. Retrieve the list of Keymasters.
- bb. Retrieve the list of Auditable Events.
- cc. Unregister a gatekeeper from a Keymaster
- dd. Get all user information from the LDAP server.
- ee. Set or Update user information in a LDAP server.
- ff. Modify a group's description
- gg. Set a user's Discretionary Control for a gatekeeper or source
- hh. Clear a gatekeeper's statistics.
- ii. Retrieve the current connection response.
- jj. Translate an admin api error code to a test message.
- kk. Allow updates of remote gatekeepers user information.

7.1.3.1 SYS ADMIN to Gatekeeper Messages

The following messages, as specified in Appendix VI, will flow from the Sys Admin to the Gatekeeper. All Administrator to Gatekeeper messages are of type BAMSG. A FUNCTION code determines the Administrator function to be performed.

- a. **Broadsword Gatekeeper Admin Lock Message.** This message with function code LOCK will lock the Gatekeeper to this SYS ADMIN user.
- b. **Broadsword Gatekeeper Admin Unlock Message.** This message with function code UNLOCK will unlock the Gatekeeper from this SYS ADMIN user.
- c. **Broadsword Gatekeeper Admin Get Plug-in Names.** This message with function code PLNAMES will retrieve the Plug-in names from the Gatekeeper.
- d. **Broadsword Gatekeeper Admin Get Template.** This message with function code GETTEM will retrieve either the Gatekeeper template or a backside source template from the Gatekeeper.
- e. **Broadsword Gatekeeper Admin Get Sources.** This message with function code GETSRCS will retrieve the list of conTabled backside sources from the Gatekeeper.
- f. **Broadsword Gatekeeper Admin Get Configuration Information.** This message with function code GETINFO will retrieve the configuration information of a backside source or local gatekeeper from the Gatekeeper.
- g. **Broadsword Gatekeeper Admin Create Backside Source.** This message with function code CREATESRC will create a new backside source.
- h. **Broadsword Gatekeeper Admin Delete Backside Source.** This message with function code DELETESRC will delete a conTabled backside source.
- i. **Broadsword Gatekeeper Admin Set a Source Parameter.** This message with function code SETSRCP is used to change an existing source parameter.
- j. **Broadsword Gatekeeper Admin Get Groups.** This message with function code GETGRPS is used to obtain a list of all known group names.
- k. **Broadsword Gatekeeper Admin Get Group Members.** This message with function code GETGRPM obtains a list of all member names in a particular group.
- l. **Broadsword Gatekeeper Admin Add a Group Member.** This message with function code ADDGRPM allows a new member name to be added to an existing group.
- m. **Broadsword Gatekeeper Admin Delete a Group Member.** This message with function code DELGRPM allows a current member name to be removed from an existing group.
- n. **Broadsword Gatekeeper Admin Delete a Group.** This message with function code DELGRP allows a complete group to be removed.
- o. **Broadsword Gatekeeper Admin Add a Group.** This message with function

code ADDGRP allows a new group name to be created.

- p. Broadsword Gatekeeper Admin Get Discretionary Access Control.** This message with function code GETDAC obtains the user, group or all permissions for a Gatekeeper or backside source.
- q. Broadsword Gatekeeper Admin Add Discretionary Access Control.** This message with function code ADDDAC sets access control for a group, user or all to a particular Gatekeeper or backside Source.
- r. Broadsword Gatekeeper Admin Delete Discretionary Access Control.** This message with function code DELDAC removes user, group or all access to a particular Gatekeeper or backside Source.
- s. Retrieve the User Discretionary Access Control.** This message with the function code GETUDAC retrieves the User Discretionary Access Control.
- t. Broadsword Gatekeeper Admin Get Users.** This message with function code GETUSERS obtains a list of user names which have particular access permissions for a Gatekeeper or Source.
- u. Broadsword Gatekeeper Admin Add User.** This message with function code ADDUSER sets either System Admin, ISSO or Producer capabilities to a user or group for a Gatekeeper or backside Source.
- v. Broadsword Gatekeeper Admin Delete User.** This message with function code DELUSER removes System Admin, ISSO or Producer capabilities from a user or group for a Gatekeeper or backside Source.
- w. Broadsword Gatekeeper Admin Modify an Element Name.** This message with function code MODEL permits an existing DATA_NAME's help text, display name, data range or data list to be changed or set.
- x. Broadsword Gatekeeper Admin Initialize Gatekeeper Registration.** This message with function code INITREG initializes the Gatekeeper registration process with a Key-master.
- y. Broadsword Gatekeeper Retrieve Gatekeeper Registration Status.** This function allows the administrator to either wait for a registration status response from the Gatekeeper or send a message with the function code ABORTREG to abort the registration currently in progress.
- z. Broadsword Gatekeeper Admin Register Our Gatekeeper.** This message with function code REGOURGKPR processes the registration with the Key-master.
- aa. Broadsword Gatekeeper Admin Get Key-masters.** This message with function code KEYMASTERS obtains a list of Gatekeepers and their registration status known as Key-masters by this Gatekeeper.
- bb. Broadsword Gatekeeper Admin Get Audit Events.** This message with function code AUDITEVENTS retrieves a list of audit event names currently being generated by the Gatekeeper.
- cc. Broadsword Gatekeeper Admin Unregister Gatekeeper.** This message with the

function code UNREGGKPR removes a Gatekeeper reference from a Keymaster.

- dd. Broadsword Gatekeeper Admin Get All User Information from the LDAP Server.** This message with function code GETUSERINFO retrieves all user information stored in the LDAP server.
- ee. Broadsword Gatekeeper Admin Set All User Information in the LDAP Server.** This message with the function code SETUSERINFO sets or updates all the user information stored in the LDAP server.
- ff. Broadsword Gatekeeper Admin Modify a Group's Description.** This message with the function code MODGRP allows the changing of a group's description.
- gg. Broadsword Gatekeeper Admin Set User Discretionary Access Control.** This message with function code SETUDAC sets or removes access to a Gatekeeper or backside Source for a user or group.
- hh. Broadsword Gatekeeper Admin Clear a Gatekeeper's Statistics.** This message with the function code CLEARSTATS clears the statistics from the database.
- ii. Broadsword Gatekeeper Admin Retrieve Connection Response Message.** This message with the function code GETCRM retrieves the current connection response message as if the administrator had just logged in.
- jj. Broadsword Gatekeeper Admin Translate API Error Code.** This function translates an Admin API error code to a test message.
- kk. Broadsword Gatekeeper Admin Set Remote Gatekeeper's User Information.** This function with the function code SETREMGKRP allows the administrator at a keymaster to update user information of a registered gatekeeper.

7.1.4 Physical Interface Specification.

The physical interface for all Gatekeeper and System Administrator activity is any network supporting TCP/IP.

Chapter 8

Broadsword CATALOG MANAGER to Gatekeeper API Specification

8.1 CATALOG MANAGER to Gatekeeper Application Program Interface

8.1.2 Interface Description

This interface is a message-based interface. The Catalog Manager must first connect to the Gatekeeper using the BGCCM message outlined in the Client to Gatekeeper interface. The user must have Catalog manager privileges for one or more sources to use this interface. One and only One CATALOG_MANAGER should manage a particular queue at a time. The gatekeeper will LOCK the queue entries to prevent over-write by other managers.

8.1.3 Functional Interface Specification

This interface provides for the following capabilities for the Catalog Manager: Appendix VII describes the API functions.

- a. **Broadsword Catalog Manager Review Queue.** This function will Review a managed queue
- b. **Broadsword Catalog Manager Delete Queue.** This function will delete a managed queue entry.
- c. **Broadsword Catalog Manager Transfer Queue.** This function will transfer a managed queue entry to a source.
- d. **Broadsword Catalog Manager Lock Queue.** This function will lock a managed queue entry.
- e. **Broadsword Catalog Manager Unlock Queue.** This function will unlock a managed queue entry.
- f. **Broadsword Catalog Manager Set Metadata.** This function will set or update the queue entry's metadata

8.1.4 Physical Interface Specification.

The physical interface for all Gatekeeper and System Administrator activity is any network supporting TCP/IP.

This page left intentionally blank

Chapter 9

Broadsword Variable Length Message Format (BVLMF).

9.1 Broadsword Variable Length Message Format Overview.

As new backside data sources become available (with their corresponding new data elements and information), it is crucial that the interface be flexible. Along with flexibility, performance, simplicity, and functionality are major requirements. BVLMF is a flexible message format that will never need a translator as elements are added or removed from future software releases. Along with being flexible for the future, BVLMF offers simplicity and a compact structure, which will enhance performance and all the functionality required for current plus future releases of Broadsword.

9.2 BVLMF Basic Structure.

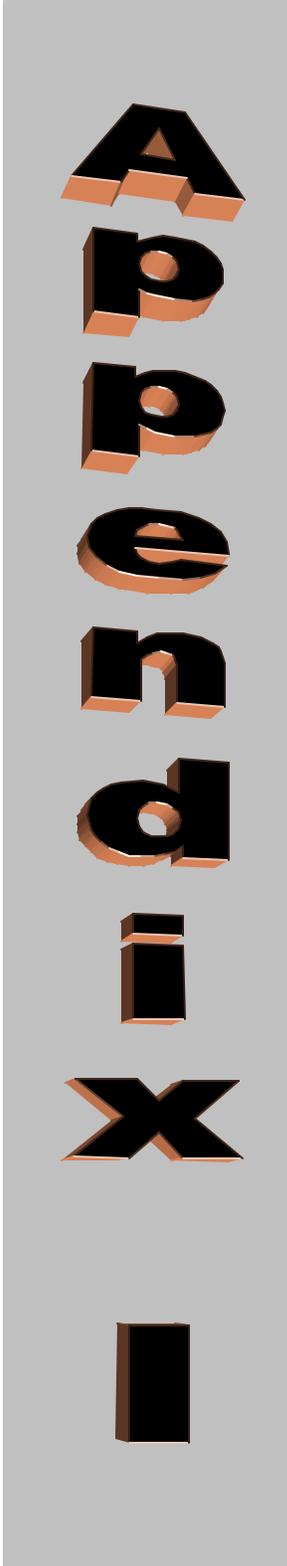
The message structure contains just three basic parts; the MESSAGE LENGTH, MESSAGE TYPE, and MESSAGE TOKENS. The MESSAGE LENGTH is a variable length numeric field, which is the total overall length of the message including the MESSAGE LENGTH. The MESSAGE TYPE is a variable length A/N field that can be equated to a function or action. The MESSAGE TOKENS contain the FIELDS of the message. These MESSAGE TOKENS are made up of three parts; The TOKEN LENGTH, TOKEN TYPE, and TOKEN DATA. The TOKEN LENGTH is a variable length numeric field, which is the overall length of the TOKEN, which includes the TOKEN LENGTH. The TOKEN TYPE is a variable length A/N field that identifies the TOKEN to follow. The TOKEN DATA is a variable length field containing the TOKEN information. The TOKEN DATA can contain any character value 0x00 - 0xFF. Its length is implied by the TOKEN LENGTH. It should be noted that tokens can appear anywhere in the message. Token order is only important if the application defines a particular order. It is also important to note that MESSAGE TOKENS can be in themselves BVLMF messages. This message is called a complex BVLMF message. The BGQRM is an example of a complex BVLMF message.

9.3 BVLMF Message Construction and Parsing

The Broadsword API provides support for both message construction and parsing as outlined in Appendix II.

(STDOUT

Broadword Gatekeeper API



The following tables describe the API. Any TOKEN_NAME with an asterisk (*) is currently not supported but planned for a future release. Note: All messages to and from the gatekeeper have a variable length numeric field delimited with a new line. This field is generated by the socket send routine and is used by the socket receive routine to allocate memory for the received message. This field is not returned to the caller of BswdReceiveMsg().

1.1 Broadsword Gatekeeper to Client Connect Message (BGCCM)

The BGCCM is used to initiate a TCP/IP socket connection to the assigned Gatekeeper. A client must make a connection to the Gatekeeper on the port specified in the server_env_vars file constructed during installation. The message format is outlined in Table 1.1-1 below.

Message Name **BGCCM**

Token Name	Meaning or Use	Format	Max. Length	Required
LOGIN_ACCOUNT	Login user ID or identity of user at the assigned Gatekeeper	A/N/S	8	Yes
LOGIN_HOST_IP	IP address of users workstation	A/N/S	15	Yes
LOGIN_PASSWORD	Login user password at the assigned Gatekeeper	A/N/S	8	Yes
ORIGINATOR_WS	User workstation user ID	A/N/S	8	Yes
CLIENT_HELP	Yes, No flag to request client help information. Default is NO.	A	1	No

Table 1.1-1

1.2 Broadsword Gatekeeper Connect Response Message (BGCRM)

The Gatekeeper responds to the client with the Gatekeeper Connect Response Message. The BGRM informs the client of the status of the connect, reason for failure, and Gatekeeper configuration information. Table 1.2-1 outlines the message format for the BGCRM. Please note that not all tokens are returned. Also note that the BGCRM is composed of multiple BVLMF sub-messages.

Parent Message Name **BGCRM**

Token Name	Meaning or Use	Format	Max Length	Required
STATUS	A Boolean status returned indicating success(0) or failure(1) Of the connect	N	1	Yes
MESSAGE	Gatekeeper reason for connection failure.	A/N	Variable	No
CLIENT_ROLE	Indicates the user's role: 0 = General user, 1 = System Admin. 2 = SSO. This token may repeat.	N/S	Variable	Yes
GATEKEEPER	This token begins the GATEKEEPER sub-message. The message defines the applicable gatekeeper and connected backside sources.	A	N/A	Yes
TYPE	Denotes the gatekeeper as (A) Assigned, (R) Remote and (K) for Keymaster	A	1	Yes
GKPR_REF	World unique gatekeeper ID	A/N/S	25	Yes
GKPR_DESC	Site gatekeeper description	A/N	Variable	Yes
GKPR_POC	Site gatekeeper point of contact	A/N/S	Variable	No
GKPR_POC_EMAIL	Site gatekeeper point of contact's email address	A/N/S	Variable	No
GKPR_POC_PHONE	Site gatekeeper point of contact's phone number.	A/N/S	Variable	No
GKPRKMFLAG	Denotes the gatekeeper as keymaster (Y)es or (N)o. Default is (N)o.	A	1	No
SOURCE	This token begins the SOURCE sub-message. This message defines a backside source. There will be at least one source.	A	N/A	Yes
SOURCE_REF	World unique source reference ID	A/N/S	47	Yes
SOURCE_NAME	A source type abbreviation for the backside source	A/N	6	Yes
SOURCE_DESC	Description of the source	A/N	Variable	Yes

Broadsword-3.1-ICD-17 JUN 02
FINAL

Token Name	Meaning or Use	Format	Max Length	Required
SOURCE_TYPE_INFO	Describes the type or kind(s) of information this source provides.	A/N	Variable	Yes
FIND	Source supports keyword search as well as BQS (Y/N). Default is N.	A	1	No
TARGET_GEO_PNT	A valid geo attribute name indicates the source supports geographic target searches by point. Default is not supported.	A	Variable	No
TARGET_GEO_CIR	A valid geo attribute name indicates the source supports geographic target searches by circle. Default is not supported.	A	Variable	No
TARGET_GEO_POLY	A valid geo attribute name indicates the source supports geographic target searches by polygon. Default is not supported.	N	Variable	No
TARGET_GEO_POLY_VERTEX	Numeric value to indicate maximum number of vertices supported for TARGET_GEO_POLY.	N	2	No
IMAGE_GEO_PNT	A valid geo attribute name indicates the source supports geographic image searches by point. Default is not supported.	A	Variable	No
IMAGE_GEO_CIR	A valid geo attribute name indicates the source supports geographic image searches by circle. Default is not supported.	A	Variable	No
IMAGE_GEO_POLY	A valid geo attribute name indicates the source supports geographic image searches by polygon. Default is not supported.	N	Variable	No
IMAGE_GEO_POLY_VERTEX	Numeric value to indicate maximum number of vertices supported for IMAGE_GEO_POLY.	N	2	No
CATALOG	Source has production capabilities (Y/N). Default is (N).	A	1	No
PRODUCER	User has producer permissions (Y/N). Default is N	A	1	No
CHIP_MODE	Method of image chipping. Geo, Faf or None. Default is None	A	1	No
QUERYABLE_FLDS	This token identifies the queryable fields sub-message. There will be one QUERYABLE_FLDS per source. This sub-message is part of the SOURCE sub-message.	A	N/A	Yes
ELEMENT_SECTION	This token identifies the element section sub-message. This sub-message is part of the QUERYABLE_FLDS sub-	A	N/A	

Broadsword-3.1-ICD-17 JUN 02
FINAL

Token Name	Meaning or Use	Format	Max Length	Required
	message. There may be more than one ELEMENT_SECTION			
SECTION_NAME	Section name	A	Variable	Yes
ELEMENT	This token begins the ELEMENT sub-message. This message defines each queryable field per section	A	N/A	Yes
DATA_NAME	Data field name	A/N	Variable	Yes
RETURNABLE_FLDS	This token identifies the returnable fields sub-message. There will be one RETURNABLE_FLDS per source. This sub-message is part of the SOURCE sub-message.	A	N/A	Yes
ELEMENT_SECTION	This token identifies the element section sub-message. This sub-message is part of the RETURNABLE_FLDS sub-message. There may be more than one ELEMENT_SECTION	A	N/A	
SECTION_NAME	Section name	A	Variable	Yes
ELEMENT	This token begins the ELEMENT sub-message. This message defines each returnable field per section	A	N/A	Yes
DATA_NAME	Data field name	A/N	Variable	Yes
CONF_ELEMENT	This token identifies the CONF_ELEMENT sub-message. These element(s) conflict with an ELEMENT in the ELEMENTS sub-message. This sub-message is part of the SOURCE sub-message.	A	N/A	No
ELEMENT_SECTION	This token identifies the element section sub-message. There may be more than one ELEMENT_SECTION. This sub-message is returned only if CLIENT_HELP = Y	A	N/A	
SECTION_NAME	Section name	A	Variable	Yes
ELEMENT	This token begins the fourth BVLMF sub-message. This message defines all elements common to all sources returned in a query response. Sources must identify all field names returned in a query response.	A	N/A	Yes
DATA_NAME	Data field name	A/N	Variable	Yes
DATA_TYPE	Data attribute type: (C) character; (N) numeric; (S) signed numeric.	A	1	Yes
MAX_LENGTH	Maximum length of data field element	N	Variable	Yes
MIN_LENGTH	Minimum length of data field element. Default is 0	N	Variable	No
DATA_PRECISION	Numeric precision for signed DATA_TYPE	N	10	No

Broadsword-3.1-ICD-17 JUN 02
FINAL

Token Name	Meaning or Use	Format	Max Length	Required
DATA_RANGE	A range expression depicting the DATA_TYPE. This field may repeat. Examples in Appendix V.	A/N/S	Variable	No
DATA_LIST	A single data value for this DATA_TYPE. This field may repeat. Examples in Appendix V.	A/N	Variable	No
QRYP	This token identifies the parameter(s) needed to query this source. This sub-message is part of the SOURCE sub-message.	A	N/A	No
SRCTOK	This token identifies a field that is required when a query is performed on this source.	A	N/A	Yes
TOKEN_NAME	Token name to be supplied in the SRC sub-message of the BGCQM.	A/N	Variable	Yes
DISPLAY_NAME	Display name for this token if different than the TOKEN_NAME	A/N	Variable	No
DATA_HELP	Token name description or help text	A/N/S	Variable	No
DATA_TYPE	Token name type (C) character; (N) numeric; (S) signed numeric	A	1	Yes
MAX_LENGTH	Maximum length of token data.	N	Variable	Yes
MIN_LENGTH	Minimum length of token data. Default is 0.	N	Variable	No
DATA_PRECISION	Numeric precision for signed token data	N	10	No
DATA_RANGE	A range expression depicting the token data. This field may repeat. Examples in Appendix V.	A/N/S	Variable	No
DATA_LIST	A single data value for this token data. This field may repeat. Examples in Appendix V.	A/N	Variable	No
REQP	This token identifies the parameter(s) needed to request a product from this source. This sub-message is part of SOURCE sub-message.	A	N/A	No
SRCTOK	This token identifies a field that is required when a request is performed on this source	A	N/A	Yes
TOKEN_NAME	Token name to be supplied in the BGCPRM message.	A/N	Variable	Yes
DISPLAY_NAME	Display name for this token if different than the TOKEN_NAME	A/N	Variable	No
DATA_HELP	Token name description or help text	A/N/S	Variable	No
DATA_TYPE	Token name type (C) character; (N) numeric; (S) signed numeric	A	1	Yes
MAX_LENGTH	Maximum length of token data.	N	Variable	Yes
MIN_LENGTH	Minimum length of token data. Default	N	Variable	No

Broadsword-3.1-ICD-17 JUN 02
FINAL

Token Name	Meaning or Use	Format	Max Length	Required
	is 0.			
DATA_PRECISION	Numeric precision for signed token data	N	10	No
DATA_RANGE	A range expression depicting the token data. This field may repeat. Examples in Appendix V.	A/N/S	Variable	No
DATA_LIST	A single data value for this token data. This field may repeat. Examples in Appendix V.	A/N	Variable	No
ELEMENTS	This sub-message identifies the ELEMENTS sub-message. This sub-message is part of the GATEKEEPER sub-message.	A	N/A	No
ELEMENT_SECTION	This token identifies the element section sub-message. There may be more than one ELEMENT_SECTION. This sub-message is returned only if CLIENT_HELP = Y	A	N/A	Yes
SECTION_NAME	Section name	A	Variable	Yes
SECTION_DISPLAY_NAME	Section name to display if different than SECTION_NAME	A/N/S	Variable	No
ELEMENT	This token begins the fourth BVLMF sub-message. This message defines all elements common to all sources returned in a query response. Sources must identify all field names returned in a query response.	A	N/A	Yes
DATA_NAME	Data field name	A/N	Variable	Yes
DISPLAY_NAME	Data field display name if different than DATA_NAME	A/N	Variable	No
MAPPING	This token begins the fifth sub-message. This message defines a source's mapping of this element. This sub-message may repeat for every source that maps this element.	A	N/A	No
SYSTEMNAME	Native source system name	A/N	Variable	Yes
MAPSTO	Native source system data name identification or nomenclature. This element ends the MAPPING sub-message	A/N	Variable	Yes
DATA_HELP	Data field element description or help text. This and the following tokens continue the ELEMENT sub-message.	A/N/S	Variable	No
DATA_TYPE	Data attribute type: (C) character; (N) numeric; (S) signed numeric.	A	1	Yes
MAX_LENGTH	Maximum length of data field element	N	Variable	Yes
MIN_LENGTH	Minimum length of data field element.	N	Variable	No

Token Name	Meaning or Use	Format	Max Length	Required
	Default is 0			
DATA_PRECISION	Numeric precision for signed DATA_TYPE	N	10	No
DATA_RANGE	A range expression depicting the DATA_TYPE. This field may repeat. Examples in Appendix V.	A/N/S	Variable	No
DATA_LIST	A single data value for this DATA_TYPE. This field may repeat. Examples in Appendix V.	A/N	Variable	No

Table 1.2.1

1.3 Broadsword Gatekeeper to Client Query Message (BGCQM)

The BGCQM is used to initiate a query or find to a source or sources. The query may be sequential or simultaneous. Table 1.3-1 outlines the message format for the BGCQM. Note that the BGCQM is composed of multiple BVLMF sub-messages.

Parent Message Name **BGCQM**

Token Name	Meaning or Use	Format	Max Length	Required
QUERY_TYPE	Indicates either Simultaneous (0) or Sequential (1) query or search. Default is Simultaneous.	N	1	Yes
THUMBNAILS	Return thumbnails if available Y or N. Default is NO.	A	1	No
MAX_THUMBNAILS	Return maximum number of thumbnails. Default is All.	N	Variable	No
MAX_HITS	Return maximum number of total hits. Default is All.	N	Variable	No
NEXT_HITS	Return this number of hits as the next block of hits. Default is All.	N	Variable	No
BQS	This token identifies the BQS sub-message. There may be more than one BQS sub-message	A	N/A	Yes
BQS_STMT	Boolean query syntax statement for query and or geo search.	A/N/S	Variable	No
BQS_STMT_MD5	Boolean query syntax statement MD5 seal	A/N/S	Variable	No
FIND_STMT	Boolean query syntax statement for keyword search.	A/N/S	Variable	No
SRC	This token identifies the SRC sub-message. There may be more than one SRC sub-message	A	N/A	
HIT_LIST *	A DATA_NAMES to return. (This	A/N/S	Variable	No

Token Name	Meaning or Use	Format	Max Length	Required
	token may repeat)			
SOURCE_REF	Source reference to route query or search. There may be more than one source reference	A/N/S	47	Yes
SOME TOKEN NAME	A source specific token name required for this source. There may be zero or more of these TOKEN NAMES for a particular source. The name or names are returned in the BGRM.	A/N/S	Variable	No

Table 1.3-1

1.4 Broadsword Gatekeeper Query Response Message (BGQRM)

The BGQRM message is used to return the query or find results to the client. Table 1.4-1 outlines overall structure of the message format for the BGQRM. This message may contain many BVLMF sub-messages. The SECTION NAMES and TOKEN NAMES are defined in the INK document.

Parent Message Name **BGQRM**

Token Name	Meaning or Use	Format	Max Length	Required
TOTAL_HITS	Total number of hits reported.	N	Variable	Yes
STATUS	This token identifies the status sub-message	A/N	N/A	Yes
STS_SOURCE_REF	Source reference reporting the status.	A/N/S	47	Yes
QRY_STATUS	Denotes SUCCESS (0) or FAILURE (1) of this source. For sequential queries, a status of (1) indicates that the source was not accessed when this source appears after the source that reported SUCCESS with SOURCE_HITS > 0.	N	1	No
SOURCE_HITS	Total number of hits reported by this source. Default is zero.	N	Variable	No
STATUS_MSG	Text status message from this source.	A/N	Variable	No
HIT	This token identifies the hit sub-message	A	N/A	No
DUP_FLAG	Indicates that this hit is a	A	1	Yes

Token Name	Meaning or Use	Format	Max Length	Required
	duplicate(Y/N)			
SOURCE_REF	Source reference generating this hit.	A/N/S	47	Yes
SECTION NAME	This token identifies a SECTION sub-message. This token may repeat.	A	N/A	Yes
ELEMENT NAME	These tokens identify the element name. The elements will only appear if data for that element exists. This token may repeat.	A/N/S	Variable	Yes

Table 1.4-1

1.5 Broadword Gatekeeper Client Product Request Message (BGCPRM)

The BGCPRM message is used to request products from the Gatekeeper's backside sources. Products can be imagery, video, text documents, audio clips, and reports. Table 1.5-1 outlines the message BGCPRM message format. Note that the BGCPRM is composed of multiple BVLMF sub-messages

Parent Message Name **BGCPRM**

Token Name	Meaning or Use	Format	Max Length	Required
SRC	This token identifies the SRC sub-message. There may be more than one SRC sub-message.	A	N/A	Yes
SOURCE_REF	World unique source ID where product resides	A/N/S	47	Yes
SOME TOKEN NAME	A source specific token name required for this source. There may be zero or more of these TOKEN NAMES for a particular source. The name or names are returned in the BGCRM .	A/N/S	Variable	No
ACCESSID	Accessid, Imageid, which uniquely identifies the product requested. Note: This token is part of the parent BGCPRM	A/N/S	64	Yes

Broadsword-3.1-ICD-17 JUN 02
FINAL

Token Name	Meaning or Use	Format	Max Length	Required
	message.			
RRDS	Reduced resolution data set requested for image product.	N	1	No
FILE_NUMBER	A comma separated numeric list for the image file(s) requested. Default is all image files.	N/S	Variable	No
IMAGENBR	A comma separated numeric list for sub-images requested from one image file. Default is all images.	N/S	Variable	No
METADATA	A Y/N flag indicating that the backside source will supply a source-defined meta-data file which contains all the meta-data about the product. Default is "N".	A	1	No
FORMAT	For image requests, the requested image format. For reports the report name or type.	A/N	Variable	No
COMPRESSION	For image requests the compression type.	A/N	2	No
RATE	For image requests the compression rate.	A/N/S	4	No
FILENAME	Requestors filename.	A/N/S	80	Yes
BQS_STMT	Boolean query syntax statement for report generation.	A/N/S	Variable	No
BQS_STMT_MD5	Boolean query syntax statemet MD5 seal	A/N/S	Variable	No
NOTIFICATION_MODE	Identifies methods clients may receive product delivery status: 0 = Notification via a socket message sent to STATUS_LOG_ADDR on port STATUS_LOG_PORT or 1 = Notification sent via email to EMAIL_ADDR or Blank = No notification. See section 1.7 for description of the socket message.	A	1	Yes
EMAIL_ADDR	User full email address if	A/N/S	Variable	No

Broadsword-3.1-ICD-17 JUN 02
FINAL

Token Name	Meaning or Use	Format	Max Length	Required
	notification method is email.			
STATUS_LOG_ADDR	IP address to which product delivery status messages are sent.	N/S	15	No
STATUS_PORT	The port number where product delivery status message is sent.	N	6	No
IDEX_REQ	This token identifies the IDEX request specific parameters sub-message	A	N/A	No
PRODUCT_SIZE	Numeric value which identifies the image product size requested	N	1	No
LAN_CLASS	Character which identifies the LAN classification for product request.	A	1	No
REGION	Upper left and lower right geographic coordinates required for image product chip generation.	A/N/S	30	No
CLEVEL	NITF compliance level.	N	2	No
POINT	Geographic point required for image product chip generation by point and radii.	A/N/S	15	No
RADIUS_MAJOR	Major radius required for image product chip generation by point.	N	4	No
RADIUS_MINOR	Minor radius required for image product chip generation by point	N	4	No
AZIMUTH	Azimuth of ellipse in degrees.	N	3	No
AUTO_RED	Image auto reduction.	N	1	No
BIT_SIZE	Image product bit size.	N	2	No
FAF	This token identifies the FAF chip method sub-image	A	N/A	No
SUB_IMAGE_TYPE	Type of sub-image requested.	A	1	No
START_FAF_H	Valid start faf horizontal.	N	5	No
START_FAF_V	Valid start faf vertical.	N	2	No
END_FAF_H	Valid end faf horizontal.	N	5	No
END_FAF_V	Valid end faf vertical.	N	2	No
EXP_FORMAT	Expansion format (00).	N	2	No

Token Name	Meaning or Use	Format	Max Length	Required
DEST	This token identifies the delivery destination sub-message. There may be more than one DEST sub-message specified.	A	N/A	Yes
DEST_ACCOUNT	Deliver product to this account name.	A/N/S	8	Yes
DEST_PASSWORD	Deliver product using this password.	A/N/S	8	Yes
DEST_IP	Deliver product to this IP address.	A/N/S	15	Yes
DEST_PATH	Deliver product to this path.	A/N/S	255	Yes
CSIL	This token identifies the CSIL request sub-message.	A	N/A	No
MAIL_ADDRESS	Deliver the product to this mailing address	A/N/S	Variable	Yes
DELIVERY_METHOD	Use this delivery method	A/N/S	Variable	Yes
PRIORITY	Request priority	A/N/S	Variable	Yes
USERNAME	CSIL registered username	A/N/S	Variable	Yes
PASSWORD	CSIL registered password	A/N/S	Variable	Yes
ORDER	This token identifies the CSIL ORDER sub-message.	A	N/A	Yes
MEDIA	Delivery media	A/N	Variable	Yes
MEDIA_FORMAT	Deliver in this image format.	A	Variable	Yes
BAND_COMBO	Band combination, a comma separated list of band numbers.	N/S	Variable	Yes
QUANTITY	Actual number of copies	N	Variable	Yes

Table 1.5-1

1.6 Broadsword Gatekeeper Product Request Acknowledgment (BGPR)

The BGPR message is used by the Gatekeeper to both acknowledge product requests have been received by the Gatekeeper and report status of those product requests. This message will be sent by the Gatekeeper for every product request (BGCPRM) and as a response to Client request for status, (BGCRFS). The BGPR will be composed of multiple BVLMF sub-messages. The message format is outlined in Table 1.6-1.

Parent Message Name **BGPR**

Token Name	Meaning or Use	Format	Max Length	Required
REQUEST_RSP	This token identifies the REQUEST_RSP sub-message.	A	N/A	Yes
REQUEST_KEY	World unique product request identifier	A/N/S	Variable	Yes
FILENAME	Users request filename.	A/N/S	255	Yes
REQ_DATE_TIME	Product request date – time.	A/N	14	Yes
SOURCE_REF	Original source reference	A/N/S	47	Yes
TASK_NUMBER	CSIL request task number.	A/N/S	Variable	No
STATUS	This token identifies the STATUS sub-message. For synchronous requests this message will be sent on request completion. For asynchronous, email, and IPC, this message will be sent as an acknowledgement of the BGCPRM . If notification mode is IPC, this message will be sent once for every request state change. There will be one STATUS sub-message for every destination in the original request.	A	N/A	Yes
REQUEST_STATUS	Current status of request. 0 = success, 1 = Failure, 2 = Pending, 3 = request audit has been archived.	N	1	Yes
STATUSMSG	Text message describing current status.	A/N/S	Variable	Yes
DEST_ACCOUNT	Identifies the destination account name.	A/N/S	8	Yes
DEST_IP	Identifies the destination IP address.	A/N/S	15	Yes
DEST_PATH	Identifies the destination path.	A/N/S	255	Yes
RPT_DATE_TIME	The report date and time.	A/N	14	Yes

Table 1.6-1

1.7 Broadsword Gatekeeper Product Request IPC Status response. BGPR

The BGPR message is used to communicate product request status via socket inter-process communication. This message is sent to the IP address and Port number identified in the BGCPRM. This is the same message as the Product Request Acknowledgement. The only difference is that is sent to the client on completion of the product request. The message format is outlined in Table 1.6-1.

1.7 Broadsword Gatekeeper Product Status Request Message (BGCRFS)

The BGCRFS message is used to request product delivery status from the Gatekeeper. Table 1.7-1 outlines the BGCRFS message.

Parent Message Name **BGCRFS**

Token Name	Meaning or Use	Format	Max Length	Required
REQUEST_INFO	This token identifies the REQUEST_INFO sub-message.	A	N/A	Yes
REQUEST_KEY	World unique product request identifier.	A/N/S	Variable	Yes
ACCOUNT	Request status for this user or account name	A/N/S	Variable	No
PASSWORD	Request status for the user with this password	A/N/S	Variable	No
TASK_NUMBER	Request CSIL status for this task number only	A/N/S	Variable	No

Table 1.7-1

1.8 Broadsword Gatekeeper Client Catalog New Product (BGCCNP)

The BGCCNP message is used to catalog a new product on a backside source. Products can be imagery, video, text documents, audio clips, or reports. Table 1.8-1 outlines the message BGCCNP message format. Note that the BGCCNP is composed of multiple BVLMF sub-messages

Parent Message Name **BGCCNP**

Token Name	Meaning or Use	Format	Max Length	Required
CATALOG_USER	The valid catalog username on the backside source.	A/N	28	No
CATALOG_PASSWORD	The catalog user's password on the backside source.	A/N	28	No
SOURCE_REF	World unique source ID where product will be pushed.	A/N/S	47	Yes
PRODTITLE	Product title.	A/N	80	Yes
IP_ADDRESS	Ip address of where the PRODUCT_FILE(S) reside.	A/N/S	15	Yes
PATH	The file path of where the PRODUCT FILE(S) reside.	A/N/S	255	Yes
CATALOG_SOURCE_USER	User account name to be used to ftp the PRODUCT_FILE(S) to the gatekeeper if the IP_ADDRESS is not the same as the gatekeeper.	A/N	28	No
CATALOG_SOURCE_PASSWORD	User account password to be used to ftp the PRODUCT_FILE(S) to the gatekeeper if the IP_ADDRESS is not the same as the gatekeeper.	A/N	28	No
SOURCE_SPECIFIC_TOKENS	Unique token(s) specific to source required for catalog.	A/N/S	Variable	No
PRODUCT_FILES	This token identifies the PRODUCT_FILES sub message. This sub-message may repeat.	A	N/A	Yes
FILENAME	New product's file name.	A/N/S	255	Yes

Table 1.8-1

1.9 Broadsword Gatekeeper Catalog New Product Acknowledgment (BGCNPA)

The BGCNPA message is used by the Gatekeeper to both acknowledge catalog new product requests have been received by the Gatekeeper and report status of those catalog new product requests. This message will be sent by the Gatekeeper for every product request (BGCCNP) and as a response to Client request for catalog status, (BGCRCFS).

The message format is outlined in Table 1.9-1.

Parent Message Name **BGCNPA**

Token Name	Meaning or Use	Format	Max Length	Required
CATALOG_KEY	World unique product catalog identifier	A/N/S	Variable	Yes
SOURCE_REF	Source reference accepting catalog	A/N/S	47	Yes
PRODTITLE	Product title	A/N	80	Yes
CAT_DATE_TIME	Catalog accepted time	A/N	14	Yes
RPT_DATE_TIME	Current status date – time	A/N	14	Yes
CATALOG_STATU S	Numeric success, failure. 0 = success, 1 = Failure, 2 = Accepted.	N	1	Yes
STATUSMSG	Text message detailing reason(s) for product catalog failure or a successful message.	A/N/S	Variable	Yes

Table 1.9-1

1.10 Broadword Gatekeeper Client Request for Catalog Status (BGCRFCS)

The BGCRFCS message is used to request product catalog status from the Gatekeeper. Table 1.10-1 outlines the BGCRFCS message.

Parent Message Name **BGCRFCS**

Token Name	Meaning or Use	Format	Max Length	Required
CATALOG_KEY	World unique catalog product identifier	A/N/S	Variable	Yes

Table 1.10-1

1.11 Broadword Gatekeeper Client Request for Top Ten Requests (BGCTTR)

The BGCTTR message is used to request a list of the top product request from the Gatekeeper. Table 1.11-1 outlines the BGCTTR message.

Parent Message Name **BGCTTR**

Token Name	Meaning or Use	Format	Max Length	Required
REQUESTS	Return the top REQUESTS where REQUESTS is a number ≥ 0 .	N	Variable	Yes
USER_NAME	Return top REQUEST for this user only.	A/N	Variable	No
SOURCE_REF	Return top REQUESTS for this source reference only.	A/N/S	Variable	No

Table 1.11-1

1.12 Broadsword Gatekeeper Top Ten Requests Response Message (BGCTTRM)

The BGCTTRM message is used to return a list of the top product requests from the Gatekeeper. Table 1.12-1 outlines the BGCTTRM message. The **REQUEST** with the highest count will appear first in the BGCTTRM. There will be zero or more **REQUEST** sub-messages returned. The actual number depends on how many were requested and how many rows are currently in the database table.

Parent Message Name **BGCTTRM**

Token Name	Meaning or Use	Format	Max Length	Required
STATUS	Status of the request, 0 = SUCCESS, 1 = FAILURE	N	Variable	Yes
REQUEST	This token identifies the REQUEST sub-message.	A	N/A	No
SOURCE_REF	This is the source reference that provided this product.	A/N/S	Variable	Yes
GKPR_REF	This is the gatekeeper reference that provided this product	A/N/S	Variable	Yes
ACCESSID	This is the source accessid for the product.	A/N/S	64	Yes
USERID	The user id that last requested the product.	A/N	28	Yes
REQUESTDTM	This is the last time the product was requested.	N	14	Yes
COUNT	This is the count for this request.	N	Variable	Yes

Table 1.12-1

1.13 Broadsword Gatekeeper Verify Connection Message.

The (BGKAM) provides both status of the connection and resets the gatekeeper timeout if the connection is valid.

Table 1.13-1 outlines the BGKAM message.

Parent Message Name **BGKAM**

Token Name	Meaning or Use	Format	Max Length	Required
SOCKET	Identifies the TCP/IP socket number.	N	Variable	Yes

Table 1.13-1

1.14 Broadsword Gatekeeper Verify Connection Response Message (BGKAMR)

The BGKAMR message is used to return connection status from the Gatekeeper. Table 1.14-1 outlines the BGKAMR message. Note: This message is not sent to the client. This message is sent by the gatekeeper to the API if the connection is valid.

Parent Message Name **BGKAMR**

Token Name	Meaning or Use	Format	Max Length	Required
STATUS	Status of the request, 0 = SUCCESS.	N	Variable	Yes

1.15 Broadsword Gatekeeper Client Column Attributes Message (BGCCAM)

The BGCCAM message is used to request column specific information from the Gatekeeper. Table 1.15-1 outlines the BGCTTR message.

Parent Message Name **BGCCAM**

Token Name	Meaning or Use	Format	Max Length	Required
SOURCE_REF	Column attributes for this source reference only.	A/N/S	Variable	Yes
SQL_STMT	Use this SQL statement for sources that support ODBC	A/N/S	Variable	No
SOME_TOKEN_NAME	Source defined token name.	A/N/S	Variable	No

Table 1.15-1

1.16 Broadsword Gatekeeper Column Attributes Response Message (BGCARM)

The BGCARM message is used to return column attribute information from the Gatekeeper. Table 1.16-1 outlines the BGCARM message.

Parent Message Name **BGCARM**

Token Name	Meaning or Use	Format	Max Length	Required
CA_STATUS	Status of the request, 0 = SUCCESS, 1 = FAILURE	N	1	Yes
CA_STATUS_MSG	Text error message if the function failed.	A/N	Variable	No
SOURCE_REF	Reference of source responding	A/N/S	47	Yes
COLUMN	This token identifies the COLUMN sub-message.	A	6	Yes
SECTION_NAME	Section name where column resides	A	Variable	Yes
DATA_NAME	Table and column name being described	A/N/S	Variable	Yes
DATA_TYPE	Data attribute type: (C) character, (N) numeric, (S) signed numeric, (D) date type	C	1	Yes
REQUIRED	DATA_NAME is required, Y for Yes, default is N	C	1	No
LIST_MODE	The number of times a column repeats. 0 denotes no limit else numeric count.	N	Variable	Yes
EXCLUDE	Exclude this character from (C) type fields.	C	1	No
MAX_LENGTH	Maximum column length	N	Variable	No
MIN_LENGTH	Minimum column length	N	Variable	No
DATA_PRECISION	Numeric precision for signed DATA_TYPE	N	10	No
DATA_RANGE	A range expression depicting the DATA_TYPE. This token may repeat.	A/N/S	Variable	No
DATA_LIST	A single valid data value for this column. This token may repeat.	A/N/S	Variable	No
DATA_LIST_HELP	Help text which helps describe the DATA_LIST item. This token may repeat.	A/N/S	Variable	No

Table 1.16-1

1.17 Broadsword Gatekeeper Client Initiate Stream Request Message (BGCISM)

The BGCISM message is used to either initiate a stream request or modify the stream filter from the Gatekeeper. Table 1.17-1 outlines the BGCISM message.

Parent Message Name **BGCISM**

Token Name	Meaning or Use	Format	Max Length	Required
SRC	This token identifies the SRC sub-message. There may be more than one SRC sub-message.	A	N/A	Yes
SOURCE_REF	World unique source ID supplying the stream.	A/N/S	47	Yes
BQS_STMT	Filter Boolean Query Syntax statement	A/N/S	Variable	Yes
STREAM_ID	Unique stream identifier. Supply this token to change a filter on a existing stream.	A/N/S	Variable	No

Table 1.17-1

1.18 Broadsword Gatekeeper Client Initiate Stream Request Response Message (BGGSR)

The BGGSR message is used to return status and the STREAM_ID for the initiate stream request or change filter request. Table 1.18-1 outlines the BGGSR message.

Parent Message Name **BGGSR**

Token Name	Meaning or Use	Format	Max Length	Required
STATUS	This token identifies the STATUS sub-message. There may be more than one STATUS sub-message.	A	N/A	Yes
SOURCE_REF	World unique source ID supplying the stream.	A/N/S	47	Yes
STATUS_CODE	Status of initiate stream, 0 = SUCCESS, 1 = FAILURE.	N	1	Yes
STATUS_MSG	Informational message if the stream request failed.	A/N	Variable	No
RSPTOMSG	Responding to message BGCISM . This token will contain the value BGCISM .	A	6	Yes

STREAM_ID	A unique identifier returned if function was successful.	A/N/S	Variable	No
STREAM_PORT	Stream Server TCP/IP Port number.	N	Variable	No
STREAM_IP	IP Address of the Stream Server.	N/S	15	No

Table 1.18-1

1.19 Broadsword Gatekeeper Client Terminate Stream Request Message (BGCTSM)

The BGCISM message is used to terminate a stream request from the Gatekeeper. Table 1.19-1 outlines the BGCTSM message.

Parent Message Name **BGCTSM**

Token Name	Meaning or Use	Format	Max Length	Required
SRC	This token identifies the SRC sub-message. There may be more than one SRC sub-message.	A	N/A	Yes
SOURCE_REF	World unique source ID supplying the stream.	A/N/S	47	Yes
STREAM_ID	Unique stream identifier.	A/N/S	Variable	Yes

Table 1.19-1

1.20 Broadsword Gatekeeper Client Terminate Stream Request Response Message (BGGSR)

The BGGSR message is used to return status for a terminate stream request. Table 1.20-1 outlines the BGGSR message.

Parent Message Name **BGGSR**

Token Name	Meaning or Use	Format	Max Length	Required
STATUS	This token identifies the STATUS sub-message. There may be more than one STATUS sub-message.	A	N/A	Yes
SOURCE_REF	World unique source ID supplying the stream.	A/N/S	47	Yes
STATUS_CODE	Status of terminate stream, 0 = SUCCESS, 1 = FAILURE.	N	1	Yes
STATUS_MSG	Informational message if the stream	A/N	Variable	No

	request failed.			
RSPTOMSG	Responding to message BGCTSM . This token will contain the value BGCTSM .	A	6	Yes
STREAM_ID	A unique identifier returned if function was successful.	A/N/S	Variable	No

Table 1.20-1

1.21 Broadsword Client Start Stream Request Message (BCSSM)

The BCSSM message is used to tell the plug-in supplying the stream to start sending data to the client. Table 1.21-1 outlines the BCSSM message.

Parent Message Name BCSSM

Token Name	Meaning or Use	Format	Max Length	Required
SRC	This token identifies the SRC sub-message. There may be more than one SRC sub-message.	A	N/A	Yes
SOURCE_REF	World unique source ID supplying the stream.	A/N/S	47	Yes
STREAM_ID	Unique stream identifier	A/N/S	Variable	Yes

Table 1.21-1

1.22 Broadsword Client Start Stream Response Message (BCSSRM)

The BCSSRM message is used to send the stream data to the client. Table 1.22-1 outlines the BCSSRM message.

Parent Message Name **BCSSRM**

Token Name	Meaning or Use	Format	Max Length	Required
STATUS	This token identifies the STATUS sub-message.	A	N/A	Yes
SOURCE_REF	World unique source ID supplying the stream.	A/N/S	47	Yes
STATUS_CODE	Status of the stream request. 0 = SUCCESS, 1 = FAILURE	N	1	Yes
STATUS_MSG	Informational message if the stream failed.	A/N	Variable	No
RSPTOMSG	Identifies the client message this	A	Variable	Yes

Broadsword-3.1-ICD-17 JUN 02
FINAL

	response is for.			
EQP	This token identifies the Equipment sub-message	A	1	No
ELNOT	Electronic notification	A/N	Variable	No
CONF	ELINT confidence	A/N	Variable	No
ELNOT2	Secondary ELINT notification	A/N	Variable	No
CONF2	Secondary ELINT notification confidence	A/N	Variable	No
QTY	This token defines the Quantity Sub-elements	A	3	No
ON_HAND	Quantity on hand	N	Variable	No

Broadsword-3.1-ICD-17 JUN 02
FINAL

TRK	This token identifies the Track sub-message	A	3	No
SPEED	Defines the speed of the ELNOT	N	Variable	No
HEAD	Defines the heading of the ELNOT	A/N	Variable	No
ALT	Defines the altitude of the ELNOT	N	Variable	No
GEO_COORDS	The geographic coordinate of the ELNOT	A/N	15	No
IW	Indications and Warnings flag	A/N	Variable	No
SCN	Sequential Contract Number	A/N	Variable	No
COR_INDX	Correlation Index	A/N	Variable	No
LOC_TYPE	Type of location	A/N	Variable	No
ID	Elint Identifier	A/N	Variable	No
SENSOR_SET_TY PE	Sensor set type	A/N	Variable	No
DFLDS	Dynamic flags	A/N	Variable	No
HDFLDS	Hex Dynamic flags	A/N	Variable	No
EQP_DESC	This token identifies the Equipment Description sub-message	A	8	No
SMA	Semi-Major Axis Length	N	Variable	No
SMI	Semi-Minor Axis Length	N	Variable	No
ORIENT	Ellipse Orientation/Bearing	S	Variable	No
FREQ	Radio Frequency	S	Variable	No
REP_INT	Pulse Repetition	S	Variable	No
PW	Pulse Width	S	Variable	No
SCAN_RATE	Scan Rate	A/N/S	Variable	No
SCAN_TYPE	Scan Type	A	Variable	No
RF	This token identifies the RF sub-tag	A	2	No
STABILITY	RF Stability	N	Variable	No
AGILITY	RF Agile Flag	A/N	Variable	No
PRI	This token identifies the PRI sub-tag	A	3	No
STABILITY	Pri Stability	N	Variable	No
TYPE	Pri Type	N	Variable	No
STAG_LEGS	Number of Stager Legs	N	Variable	No
OBS	This token identifies the Observation sub-message	A	3	No
TIME_OBS	Date Time of Intercept	N	14	No
ARP	This token identifies the Associated Reports sub-message	A	3	No
ID	Report Number	N	Variable	No
ID_UPDATE	Update Number	N	Variable	No
PRD	This token identifies the Document Management sub-message	A	3	No

DATECREATED	Time of Receipt	N	14	No
PRODSNME	Message Type	A	2	No
TGT	This token identifies the Target Detail Sub-message	A	3	No
ID	Target Identifier	A/N	10	No

Table 1.22-1

1.23 Broadsword Client Change Password Request Message (BGCPM)

The BGCPM message is used to change a users password. Table 1.23-1 outlines the BGCPM message.

Parent Message Name **BGCPM**

Token Name	Meaning or Use	Format	Max Length	Required
OLD_PASSWORD	This token identifies the old password that is to be changed	A/N/S	8	Yes
NEW_PASSWORD	This token identifies the new password.	A/N/S	8	Yes

Table 1.23-1

1.24 Broadsword Client Change Password Response Message (BGCPSM)

The BGCSPM message is used to send the status of the change password function. Table 1.24-1 outlines the BGCSPM message.

Parent Message Name **BCSSRM**

Token Name	Meaning or Use	Format	Max Length	Required
CHG_PWD_STAT US	Change password status code	N	2	Yes
CHG_PWD_STAT US_MSG	Change password status message.	A/N	Variable	Yes

Table 1.24-1

1.23 Broadsword Get Site Catalog Elements Message (BGGSCCE)

The BGGSCCE message is used to retrieve the catalog elements and valid values from a source that supports the catalog function. Table 1.25-1 outlines the BGGSCCE message.

Parent Message Name **BGGSCCE**

Token Name	Meaning or Use	Format	Max Length	Required
SOURCE_REF	Reference of the source	A/N/S	47	Yes

Table 1.25-1

1.24 Broadsword Gatekeeper Get Site Catalog Element Response Message (BGGSCERM)

The BGGSCERM message is used to return Site Catalog Element information from the Gatekeeper to the Client. Table 1.26-1 outlines the BGGSCERM message.

Parent Message Name **BGGSCERM**

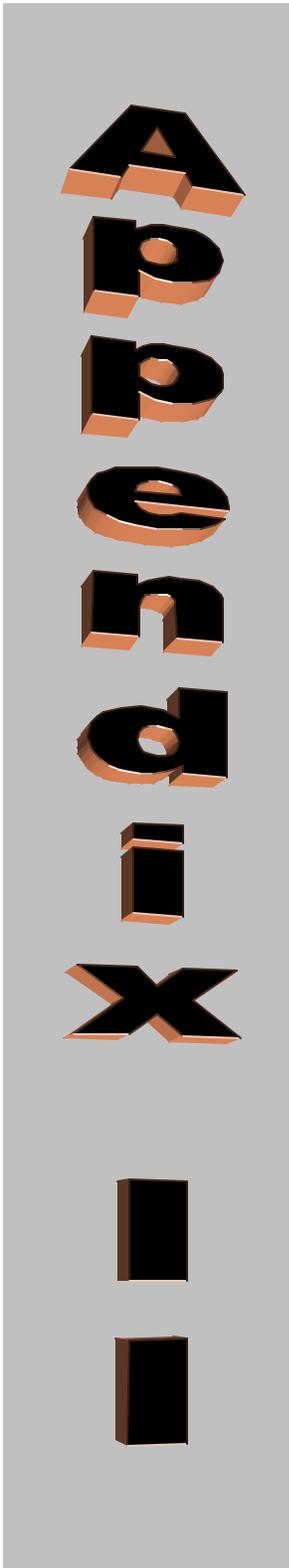
Token Name	Meaning or Use	Format	Max Length	Required
STATUS	Status of the request, 0 = SUCCESS, 1 = FAILURE	N	1	Yes
STATUS_MSG	Test error message if the function failed	A/N	Variable	No
SOURCE_REF	Reference of source responding	A/N/S	47	Yes

Broadsword-3.1-ICD-17 JUN 02
FINAL

ELEMENT	This token identifies the ELEMENT sub-message. This token may repeat.	A	7	Yes
SECTION_NAME	Section name where column resides	A	Variable	Yes
SECTION_DISPLAY_NAME	Section display name	A/N/S	Variable	Yes
DATA_NAME	Table and column name being described	A/N/S	Variable	Yes
DISPLAY_NAME	Data field display name	A/N/S	Variable	Yes
DATA_TYPE	Data attribute type: (C) character, (N) Numeric, (S) Signed numeric, (D) Date type, (Z) Zulu date	A	1	Yes
REQUIRED	Is this Element required for cataloging? Y/N/M. Default is N.	A	1	Yes
DATA_SOURCE	Specified by (S)ource. Default is Site	A	1	Yes
MAX_LENGTH	Maximum column length.	N	Variable	Yes
MIN_LENGTH	Minimum column length. Default is 0.	N	Variable	No
DATA_PRECISION	Numeric precision for signed DATA_TYPE	N	10	No
DATA_HELP	Data field description or help text	A/N/S	Variable	No
DATA_RANGE_HELP	Help text which describes the DATA_RANGE. This token may repeat.	A/N/S	Variable	No
DATA_LIST	A single valid data value for this column. This token may repeat	A/N/S	Variable	No
DATA_LIST_HELP	Help test, which describes the DATA_LIST item. This token may repeat.	A/N/S	Variable	No

Table 1.26-1

API Function Definitions, and Errors



2.1. API Function Definitions

The Broadsword Client to Gatekeeper API defines the "C" functions to generate the various messages defined in the ICD. This API is comprised of the following functions:

- a. BswdClientConnect() ... Establish connection to gatekeeper and return status, source information.
- b. BswdClientQuery() ... Send a query, find or geo search to the source(s) selected and return with query status and any query results.
- c. BswdClientProductReq() ... Send a product request and return status of that product request.
- d. BswdClientRfs() ... Send a product request for status and return product request status.
- e. BswdClientCnp() ... Catalog a new product on a backside source.
- f. BswdClientRfcs() ... Send a request for catalog status and return catalog status.
- g. BswdClientRtt() ... Send a request for the top ten requests.
- h. BswdClientVgc() ... Send a verify connection message.
- i. BswdClientColumnAttr() ... Send a retrieve column attributes message.
- j. BswdClientInitateStream() ... Send a initiate stream message.
- k. BswdClientTermStream() ... Send a terminate stream message.
- l. BswdClientStartStream() ... Send a start stream message to the stream server.
- m. BswdClientChangePassword ... Send a change password message to the gatekeeper.
- n. BswdClientProductReq30 ... Send a product request message to the gatekeeper using the More flexible version 3.0 API function.
- o. BswdClientGetSiteCatElem() ... Send a request for the site catalog elements.
- p. BswdClientLogout() ... Log of the client from the Gatekeeper.

All functions return a integer status equal to SUCCESS (0) or FAILURE (non zero). Reasons for failure are detailed in the eAPI_ERRORS enumeration.

2.1.1 BswdClientConnect()

Calling Sequence:

```
int BswdClientConnect(pCONNECT_PARAMS pConnect_params,- Structure containing
                    parameters                               connection
                    int      *pSocket_id,                   - Socket ID returned
                    int      mode,                           - FILE_MODE,
                    unsigned char **hConnect_rsp)           - MEMORY_MODE
                                                            - the BGCRM
```

The caller fills out the structure pCONNECTP_PARAMS with applicable information. The mode allows the caller to accept the response either in a UNIX flat FILE or heap MEMORY. If the mode is FILE_MODE, then the caller supplies the full file path and file name in hConnect_rsp. If the mode is MEMORY_MODE, the response is returned in hConnect_rsp and the caller is responsible for freeing the HEAP memory.

Return values are API_SUCCESS for a successful login. The socket connection to the Gatekeeper is returned in pSocket_id and hConnect_rsp contains the BGCRM. For status = API_FAILURE, the reason for failure is found in the BGCRM. For any other status, hConnect_rsp is null and the reason for failure is an API failure, usually invalid parameters in tCONNECT_PARAMS. The user is required to parse the BGCRM for applicable connection information and freeing the hConnect_rsp.

The following is the tCONNECT_PARAMS structure:

```
typedef struct
{
    char *pPort; /* TCP/IP port number gatekeeper is
listening on*/
    char *pGkpr_hostname; /* Hostname or IP address where
gatekeeper is
running */
    char *pLogin_account; /* Broadsword UNIX login account */
    char *pLogin_host_ip; /* Client workstation IP address */
    char *pLogin_pwd; /* Broadsword UNIX login password */
    char *pOriginator_ws; /* Client's originator userid at his
workstation */
    char *pClient_help; /* Want help data returned Y or N */
}tCONNECT_PARAMS, *pCONNECT_PARAMS;
```

2.1.2. BswdClientQuery()

Calling Sequence:

```
int BswdClientQuery(pQUERY_PARAMS pQuery_params, /* query parameters */
                  int socket_id, /* socket id */
                  int mode; /* FILE_MODE or
                             MEMORY_MODE*/
                  char **hQuery_rsp)/* the BGQRM */
```

The following is the tQUERY_PARAMS structure:

```
typedef struct
{
    char *pQuery_type; /* Simultaneous 0 or Sequen 1 */
    char *pThumbnails; /* Want thumbnails returned Y or N
*/
    char *pMax_thumbnails; /* If thumbnails = Y then this is
                           the maximum number of
                           thumbnails
                           returned */
    char *pMax_hits; /* maximum number of hits per
source
                           to be returned. Default is ALL */
    tBQS_PARAMS *pBqs_params; /* linked list of BQS messages */
}tQUERY_PARAMS, *pQUERY_PARAMS ;

typedef struct BQS_PARAMS tBQS_PARAMS;
typedef struct BQS_PARAMS *pBQS_PARAMS;

struct BQS_PARAMS
{
    char *pBqs_stmt; /* BQS statement to be sent to 1 or
more
                           sources */
    char *pBqs_stmt_md5; /* MD5 seal for above BQS
statement */
    char *pBqs_stmt_md5_len; /* Length of pBqs_stmt_md5 seal */
    char *pFind_stmt; /* Keyword search statement to be
sent to 1
                           or more sources */
    pSOURCES pSrc_reference; /* linked list of sources to pass the
Bqs_
                           statement */
    char *pHit_list; /* comma separated hit list
(not implemented in this release)*/
    pBQS_PARAMS pNext_bqs; /* next BQS src etc, null if last */
```

```
};

typedef struct SOURCES tSOURCES;
typedef struct SOURCES *pSOURCES;

struct SOURCES
{
    char          *pSource_ref;          /*source reference of source
returned in
                                         the connect response message */
    pEXTRA_PARAMS pParams;              / Source specific parameters linked
list */
    pSOURCES      pNext_src;            / next source in list, NULL if no
more */
}

typedef struct EXTRA_PARAMS tEXTRA_PARAMS;
typedef struct EXTRA_PARAMS *pEXTRA_PARAMS;

struct EXTRA_PARAMS
{
    char          *pParam_name;          /* Parameter or token name */
    char          *pParam_data; /* Parameter data value */
    pEXTRA_PARAMS pNext_param /* next token/value pair */
}

```

The caller fills out the structure tQUERY_PARAMS with applicable information. The mode allows the caller to accept the response either in a UNIX flat FILE or heap MEMORY. As with the BswdClientConnect, if the mode is FILE_MODE, then the caller supplies the full file path and file name in hQuery_rsp. If the mode is MEMORY_MODE, the response is returned in hQuery_rsp. The caller is responsible for freeing the HEAP memory.

Return values are API_SUCCESS for a successful query response from the Gatekeeper and the hQuery_rsp contains the BGQRM. If the status is not API_SUCCESS, then the hQuery_rsp will be null and the reason for failure is an API failure, usually invalid parameters in tQUERY_PARAMS. The caller is responsible for parsing the BGQRM for applicable query status and results and freeing the hQuery_rsp.

2.1.3 BswdClientProductReq()

Calling Sequence:

```
int BswdClientProductReq(pPRODUCT_REQ      pProd_req_param, /* product
request                                     parameters */
                        int                 socket_id,         /* socket id */
                        int                 mode,              /* FILE_MODE or
                                                                MEMORY_MODE */
                        unsigned char       **hRequest_rsp)    /* the BGPRA */
```

The following is the tPRODUCT_REQ structure:

```
typedef struct
{
the      char      *pSource_ref;          /* The source reference which has
parameters */
returned in pEXTRA_PARAMS pParams;      /* linked list of source specific
0-8 */      char      *pAccessid;            /* The identifier of the product
comma      char      *pRrds;            /* the query response */
files */      char      *pFile_number;    /* Reduced resolution data set value
char      *pImagenbr;                  /* Which file of a multi-file product:
char      *pMetadata;                  separated list 1,2,3... or 0 for all
char      *pFormat;                    /* comma separated list of images to
char      *pCompression;                extract 1,2,3 or 0 for all image(s)*/
sent      char      *pNotify_mode;      /* Want source specific metadata file
with product Y or N */
=          char      *pRate;            /* destination format or report */
          char      *pFilename;        /* compression mode C3= JPEG, C2
          char      *pBqs_stmt;        ARIDPCM */
          char      *pBqs_stmt_md5;    /* compression rate based on
          char      *pNotify_mode;     pCompression */
          /* BQS for (MIDB...) reports */
          /* MD5 seal for BQS statement */
          /* notification mode: 0 for socket
```

```

based
                                notification, 1 for email or Blank
                                for
                                none. */
char *pEmail_addr; /* email address for notification*/
char *pStatus_log_addr; /* IP address to send the request
status
                                IPC messages */
char *pStatus_port; /* TCP/IP port # to send the request
status
                                IPC messages */
char *pProduct_size; /* numeric product size: 1 =
1024x1024
                                2 =
                                2048x2048
                                3 =
                                4096x4096
                                4 =
                                6144x6144
                                5 =
                                8192x8192
                                6 = Full
                                Segment */
char *pLan_class; /* LAN classification of client S =
SCI, C =
                                Collateral */
char *pRegion; /* UL and LR geo region of request
for
                                chip*/
char *pClevel; /* NITF compliance level 1 - 6*/
char *pPoint; /* geo point for elliptical chipping*/
char *pMajor_radius; /* Semi major radius in nautical
miles */
char *pMinor_radius; /* Semi minor radius in nautical
miles */
char *pAzimuth; /*Azimuth of ellipse in degrees (0-
359)*/
char *pAuto_red; /* auto reduction:1= Suitable RRDs,
2=
                                Send Image Cropped, 3= Abort
request
                                If cant satisfy client requirements
*/
char *pBit_size; /* Image bit size 8,16,24 */
char *pSub_image_type; /* IPL 2.0 requirement */
char *pStart_faf_h; /* faf # for chipping IPL 2.0 */

```

```
char      *pStart_faf_v;      /* faf # for chipping IPL 2.0 */
char      *pEnd_faf_h;       /* faf # for chipping IPL 2.0 */
char      *pEnd_faf_v;       /* faf # for chipping IPL 2.0 */
char      *pExp_format;      /* TFRD expansion format */
pDESTINATION pDests;        /* linked list of destinations */
pCSIL     pCsil_params;      /* CSIL parameters */

}tPRODUCT_REQ ;

typedef struct DESTINATION tDESTINATION;
typedef struct DESTINATION *pDESTINATION
struct DESTINATION
{
    char      *pDest_account;  /* destination ftp account name */
    char      *pDest_password; /* destination ftp account password
*/
    char      *pDest_ip;      /* destination IP address */
    char      *pDest_path;    /* destination path not including
filename*/
    pDESTINATION pNext;      /* next destination or NULL if no
more*/
};

typedef struct CSIL tCSIL;
typedef struct CSIL *pCSIL
struct CSIL
{
    char      *pMail_address;  /* Deliver product to this mailing
address */
    char      *pDel_method;    /* Use this delivery method */
    char      *pPriority;      /* At this priority */
    char      *pUsername;     /* Registered CSIL user*/
    char      *pPassword;     /* Registered CSIL password */
    pORDER    pOrder;        /* The CSIL order information */
};

typedef struct ORDER tORDER;
typedef struct ORDER *pORDER
struct CSIL
{
    char      *pMedia;        /* Delivery media type */
    char      *pMedia_format; /* Delivery media format*/
    char      *pBand_combo;   /* Band combination*/
    char      *pQuantity;     /* Order quantity*/
    char      *pHardcopy_size; /* hardcopy media size */
    pORDER    pOrder_next;    /* Next CSIL order or NULL if last
```

```
*/
};
```

The caller fills out the structure tPRODUCT_REQ with applicable information. The mode allows the caller to accept the response either in a UNIX flat FILE or heap MEMORY.

Return values are API_SUCCESS for a successful product request from the Gatekeeper and the hRequest_rsp contains the BGPRA. If the status is not API_SUCCESS, then the hRequest_rsp will be null and the reason for failure is an API failure. The caller is required to parse the BGPRA for applicable product request status.

2.1.4 BswdClientRfs()

Calling Sequence:

```
int BswdClientRfs(pREQUEST_IDS *pRequest_key,      /* world unique request
key(s) */
                  const int    socket_id;          /* socket id */
                  int          mode;              /* FILE or MEMORY */
                  char         **hRfs_rsp)        /* the BGPRA */
```

The caller fills out the request identifier structure, socket_id and mode with the applicable information. The mode allows the caller to accept the response either in a UNIX flat FILE or heap MEMORY.

Return values are API_SUCCESS for a successful product status request from the Gatekeeper and the hRfs_rsp contains the BGPRA. If the status is not API_SUCCESS, then the hRfs_rsp will be null and the reason for failure is an API failure. The caller is required to parse the BGPRA for applicable product request status.

Following is the REQUEST_IDS structure.

```
typedef struct REQUEST_IDS tREQUEST_IDS;
typedef struct REQUEST_IDS *pREQUEST_IDS;
struct REQUEST_IDS
{
    char    *pRequest_id;      /* request key returned from request
*/
    char    *pAccount;        /* Csil account name */
    char    *pPassword;      /* Csil registered password */
    char    *pTask_nbr;      /* Csil task number returned from
request */
    pREQUEST_IDS    pNext_id; /* next id or NULL if last */
};
```

2.1.5 BswdClientCnp()

Calling Sequence:

```
int BswdClientCnp(pCAT_NEW_PRODUCT      pNewProductParams, /* username,
                                           password,
                                           files */
                 int                    socket_id;           /* socket id */
                 int                    mode;               /* FILE or
                                                           MEMORY
                                                           */
                 char                   **hCnp_rsp)         /* the
BGCNPA*/
```

The caller fills out the new product parameters, socket_id and mode with the applicable information. The mode allows the caller to accept the response either in a UNIX flat FILE or heap MEMORY.

Return values are API_SUCCESS for a successful product status request from the Gatekeeper and the hCnp_rsp contains the BGCNPA. If the status is not API_SUCCESS, then the hCnp_rsp will be null and the reason for failure is an API failure. The caller is required to parse the BGCNPA for applicable catalog new product status.

The following is the tCAT_NEW_PRODUCT structure:

```
typedef struct
{
    char        *pUser_name,           /* catalog username */
    char        *pUser_pwd,           /* catalog user password */
    char        *pSource_ref,         /* source reference */
    char        *pProd_title,         /* product title */
    char        *pProduct_path,       /* path where new product file exists
*/
    char        *pProduct_ip,         /* IP address where new product
exists */
    char        *pMy_name,            /* Ftp user name */
    char        *pMy_pwd,            /* Ftp user password */
    pEXTRA_PARAMS pParams;           /* linked list of source specific
parameters */
    pFILE_ID_LIST pFile_list;         /* linked list of filename(s) */
} tCAT_NEW_PRODUCT, *pCAT_NEW_PRODUCT;

typedef struct FILE_ID_LIST tFILE_ID_LIST;
typedef struct FILE_ID_LIST *pFILE_ID_LIST;
struct FILE_ID_LIST
```

```
{
    char      *pProduct_filename;    /* new product file name */
    pFILE_ID_LIST pNext;            /* next file or NULL if no more*/
};
```

2.1.6 BswdClientRfcs()

Calling Sequence:

```
int BswdClientRfcs(char      *pCatalog_key,    /* world unique catalog key
*/
                    const int  socket_id;      /* socket id */
                    int        mode;          /* FILE or MEMORY */
                    char        **hRfcs_rsp)   /* the BGCNPA */
```

The caller fills out the catalog key, socket_id and mode with the applicable information. The mode allows the caller to accept the response either in a UNIX flat FILE or heap MEMORY.

Return values are API_SUCCESS for a successful product status request from the Gatekeeper and the hRfcs_rsp contains the BGCNPA. If the status is not API_SUCCESS, then the hRfcs_rsp will be null and the reason for failure is an API failure. The caller is required to parse the BGCNPA for applicable product catalog status.

2.1.7 BswdClientRtt()

Calling Sequence:

```
int BswdClientRtt(char      *pUser_name,      /* Top Ten for this User */
                    char      *pSource_ref,    /* Top Ten from this source
*/
                    int        mode;          /* FILE or MEMORY */
                    char      *pRequests,     /* return the top REQUESTS
where
                    char      *pMode,        REQUESTS is >= 0 */
or R
                    for top ten queries
*/
                    const int  socket_id;     /* socket id */
                    char      **hRtt_rsp)     /* the BGCTTRM */
```

The caller fills out the pUser_name, pSource_ref and socket_id with the applicable information. Note: The pUser_name and pSource_ref may be NULL. The mode allows

the caller to accept the response either in a UNIX flat FILE or heap MEMORY.

Return values are API_SUCCESS for a successful top ten response from the Gatekeeper and the hRtt_rsp contains the BGCTTRM. If the status is not API_SUCCESS, then the hRtt_rsp will be null and the reason for failure is an API failure. The caller is required to parse the BGCTTRM for applicable top ten information.

2.1.8 BswdClientVgc()

Calling Sequence:

```
int BswdClientVgc(const int      socket_id;          /* socket id */
```

The caller fills out the socket_id with the applicable information. Return values are API_SUCCESS indicates the connection is still valid. Status other than API_SUCCESS indicates that the connection is no longer valid. The client must re-establish connection to the Gatekeeper via the Login Function.

2.1.9 BswdClientColumnAttr()

Calling Sequence:

```
int BswdClientColumnAttr(char      *pSource_ref,      /* of this source */
                           char      *pSql_stmt,        /* using this sql
                           statement */
                           pEXTRA_PARAMS pParams,      /* linked list of source
                           values                      defined tokens and
                                                         */
                           const int  socket_id;        /* socket id */
                           const int  mode,             /* receive response in
                           file or
                           char      **hCa_rsp)         /* caller passes
                           filename or
                                                         function returns
                           pointer to
                                                         response in heap
                           memory.
                                                         Caller responsible
                           for
                                                         Freeing memory */
```

The caller fills out the source reference, optional sql statement, mode, socket_id, and response handle with the applicable information. Return values are API_SUCCESS indicates the call was successful and the response can be found in the handle. Status other than API_SUCCESS indicates that the function failed for some reason.

2.1.10 BswdClientInitiateStream()

Calling Sequence:

```
int BswdClientInitiateStream(char      *pSource_ref,      /* of this source */
                             char      *pBqs_stmt,        /* using this bqs
                             statement */
                             char      *pStream_id,      /* stream id to change
                             the
                             filter */
                             const int  socket_id;        /* socket id */
                             const int  mode,            /* receive response in
                             file or
                             memory */
                             char      **hIs_rsp)        /* caller passes
                             function returns
                             pointer to
                             response in heap
                             memory.
                             Caller responsible
                             for
                             Freeing memory */
```

The caller fills out the source reference, bqs statement, mode, socket_id, stream id and response handle with the applicable information. Return values are API_SUCCESS indicates the call was successful and the response can be found in the handle. Status other than API_SUCCESS indicates that the function failed for some reason.

2.1.11 BswdClientTermStream()

Calling Sequence:

```
int BswdClientTermStream(char      *pSource_ref,      /* of this source */
                           char      *pStream_id,      /* stream id to
                           terminate */
                           const int  socket_id;        /* socket id */
                           const int  mode,            /* receive response in
```

file or
char filename or
pointer to
memory.
for

memory */
/* caller passes
function returns
response in heap
Caller responsible
Freeing memory */

The caller fills out the source reference, mode, socket_id, stream id and response handle with the applicable information. Return values are API_SUCCESS indicates the call was successful and the response can be found in the handle. Status other than API_SUCCESS indicates that the function failed for some reason.

2.1.12 BswdClientStartStream()

Calling Sequence:

```
int BswdClientStartStream(char *pSource_ref, /* of this source */
char *pStream_id, /* using this stream id
*/
char *pPort, /* port # of stream
plug-in */
char *pIp, /* hostname or ip
address of
char *pSocket_id) stream plug-in */
/* returns socket to
read stream data */
```

The caller fills out the source reference, stream-id, plug-in port, and plug-in port with the information received from the initiate stream request response. Return values are API_SUCCESS indicates that the stream plug-in will now initiate the data stream. The stream data will be sent on the socket referenced by pSocket_id. The client must issue a BswdReceiveMsg to obtain this stream data. Status other than API_SUCCESS indicates that the function has failed for some reason.

2.1.13 BswdClientChangePassword()

Calling Sequence:

```
int BswdClientChangePassword(int socket_id, /* socket id returned by
                                           BswdClientConnect */
                             char *pOld_pwd, /* users old password */
                             char *pNew_pwd, /* users new password */
                             char **hMsg); /* return optional text message */
```

The caller passes the connected socket id a pointer to the old password and a pointer to the new password. The API returns a status of API_SUCCESS if the function was successful. Any other value indicates some error.

2.1.14 BswdClientProductReq30()

Calling Sequence:

```
Int BswdClientProductReq30(pPRODUCT_REQ30 pProd_req_params, /* product
request
                                                                    Parameters
*/
                           int socket_id, /* socket id */
                           int mode, /* response
type */
                           char **hRequest_rsp); /* either a
filename or
                                                                    or returned
memory
                                                                    pointer. */
```

The caller completes the required and optional entries in the product request parameters, the socket identifier and the mode. If the caller requires the response in a file, the request_rsp points to a filename complete with path. A status of API_SUCCESS indicates that the function was successful and the request has been accepted by the gatekeeper. Any other value indicates the function failed.

Following is the pPRODUCT_REQ30 structure:

```
typedef struct
{
    char *pSource_ref; /* the source which has the product/report etc */
    char *pAccessid; /* the identifier of which product to request or an
identifier
                    which identifies a particular request */
    char *pFilename; /* users destination file name */
    char *pBqs_stmt; /* the optional BQS statement used as a filter for the
                    product or report */
    char *pNotify_mode; /* Notify delivery status either by
```

IPC_NOTIFICATION

```
                                or EMAIL */
char *pEmail_addr;           /* users full email address if notification mode is
EMAIL*/
char *pStatus_log_addr;     /* TCP/IP address to send the IPC_NOTIFICATION
product
                                delivery status message */
char *pSZtatus_port;       /* TCP/IP port to send the IPC_NOTIFICATION
product
                                delivery status message */
pEXTRA_PARAMS pParams /* linked list of source specific required tokens or
fields */
pDESTINATION pDests;      /* linked list of user destination information */
} tPRODUCT_REQ30, *pPRODUCT_REQ30;
```

appear in the audit.

2.1.15 BswdClientGetSiteCatElem()

Calling Sequence:

```
int BswdClientGetSiteCatElem( char *pSource_ref, /* of this source reference*/
                                int socket_id,    /* send msg on this socket */
                                int mode,         /* file or memory */
                                char **hRsp);     /* caller passes filename for response
                                                    or function returns pointer to
                                                    response in memory */
```

The caller fills out the socket_id and pSource_ref with the applicable values. A return of SUCCESS indicates that the get site elements was successful and the response is supplied.

2.1.16 ClientLogout()

Calling Sequence:

```
int ClientLogout(int socket_id); /* socket id to disconnect */
```

The caller fills out the socket_id with the applicable value. A Return value of SUCCESS indicates that the logout was successful. A value of FAILURE indicates that the socket_id supplied is not currently valid.

2.2 API ERRORS

typedef enum

```
{
    API_FAILURE = -1, /* general failure status
    API_SUCCESS, /* good status */

    /* errors unique to the client connect */

    API_CONN_NULL_PARAM, /* one of the connection
                           NULL */
    API_COULD_NOT_CONNECT, /* Socket connection failed */
    API_LOGIN_FAILURE, /* Bad login status from
                       gatekeeper */

    /* errors unique to the client query */

    API_QRY_NULL_PARAM, /* One of the query params
                           NULL*/

    /* errors unique to the client product request */

    API_PROD_REQ_NULL_PARAM, /* One of the Product Req
                               was NULL */

    /* errors unique to the client request for product status */

    API_NO_REQUEST_KEY, /* no request key supplied */

    /* errors common to all API calls */

    API_BUILD_MSG, /* Build BVLMF message
failed */
    API_PARSE_MSG, /* Parse BVLMF message failed
*/

    API_NO_MEMORY, /* couldn't malloc memory */
    API_FILE_WRITE, /* there was a file write error */
    API_FILE_OPEN, /* file open failure */
    API_IPC_FAILURE, /* network communication
failure */

    API_IPC_READ, /* network comm read error */
    API_IPC_WRITE, /* network comm write error */
    API_SOCKET, /* invalid socket id */
    API_MODE /* invalid mode specified */
    API_NO_STREAM, /* stream id missing */
    API_INVALID_STREAM, /* invalid stream id */
    API_USERUPD_NOT_ALLOWED, /* changing password not
allowed */

    API_PWD_CIR_SHIFT, /* password is a circular shift of
```

required	API_PWD_SPECIAL_CHAR,	Username */ /* password doesn't have the Number of special characters
/	API_PWD_IN_HISTORY,	/ password is in password
history */	API_PWD_TO_LONG, API_PWD_TO_SHORT, API_PWD_WHITESPACE,	/* password is too long */ /* password is too short */ /* password contains a
whitespace */	API_ACCOUNT_LOCKED, API_PWD_IN_DICT,	/* the account is locked */ /* password is invalid, its in the Password dictionary */
as the	API_PWD_EQ_UID,	/* password cannot be the same
match	API_PWD_OLD_PWD_INV	Userid */ /* given old password doesn't
	}eAPI_ERRORS;	Current password */

2.3 Support Functions

In addition to the above mentioned API functions, the following functions are supplied to parse the BVLMF response messages.

- a. ParseMsgSize() ... Returns the message size in bytes.

Calling Sequence:

```
int ParseMsgSize(char    *pMsg,    --- Pointer to the message being sized
                   char    **hFunc, --- Handle to the message function name
                   int     filename) --- TRUE indicates that pMsg is a FILE
```

Returns the integer byte size of the message. The hFunc returned is the message name. A value of (-1) is returned if the size of the message is non numeric or pMsg is NULL. The hFunc is heap memory and must be freed by the user.

- b. ParseAllMsg() ... Parses a message token by
token returning the token value and
length.

Calling Sequence:

int ParseAllMsg(int	*pOffset,	---	relative offset in hValid table
char	**hValid,	---	Null terminated char array of valid token
types.			
int	mode,	---	IGNORE or FLAG unidentified token
types			
char	*pMsg	---	Points to the beginning of the message or filename if MAP
char	**hType,	---	handle to token type
char	**hToken,	---	handle to token value
tPARSE_INFO	*pResume,	---	Reentrant information
int	filename)	---	MAP Indicates pMsg is a file

Returns the integer size of *hToken if a valid token was found. A value of PARSE_EOD indicates end of message. A status of FAILURE (-1) indicates a parse failure. The structure tPARSE_INFO allows the function to be re-entrant and thread safe. Since the BVLMF structure is recursive, recursion can be used to parse a multi-message BVLMF like the BGQRM.

c. ParseAllMsgClean() ... Cleans up memory map and file pointers when message being parsed is a flat file.

Calling Sequence:

void ParseAllMsgClean(tPARSE_INFO *pResume) --- Reentrant information

Cleans up the memory map and file id stored in tPARSE_INFO when the parser is parsing a message contained in a flat file.

d. ShowMsg() ... Displays the contents of a BVLMF message to standard out.

Calling Sequence:

void ShowMsg(char *pMsg, --- Head of message to be displayed
char fill) --- Fill character for NULL values

This function is a debug function which displays the message contents in standard out

Broadsword Gatekeeper to Plug-in Application Program Interface



The following tables describe the message differences between the original client message and the message sent from the gatekeeper to the plug-in and the plug-in to gatekeeper responses.

3.1 Broadsword Gatekeeper to Client Query Message (BGCQM)

Token Name	Meaning or Use	Format	Max Length	Required
TRACE_KEY	Unique Gatekeeper generated key	A/N/S	Variable	Yes

The TRACE_KEY will be the first token in the BGCQM message sent from the Gatekeeper to a plug-in. All other tokens will be the same as specified in the Client ICD. The TRACE_KEY is used by the WriteLogEntry (system log function) to track errors, faults and anomalies throughout the life of the BGCQM. The plug-in must call the function SetTraceKey() supplying the TRACE_KEY received to enable this tracking.

It should be noted that if the Client's original BGCQM specified additional sources, the message the plug-in receives from the Gatekeeper will only contain this plug-in's source reference.

The plug-in must respond to the Gatekeeper with the BGQRM as specified in the Client ICD. The plug-in may call the function CreateQrfResponse() if the plug-in made the decision to create the BGQRM in a UNIX flat file. This function will create the **BGQRF** message:

Token Name	Meaning or Use	Format	Max Length	Required
RSP_FILE	This token identifies the response file name (full path)	A/N/S	Variable	Yes

The BGQRF message is sent to the Gatekeeper in lieu of the BGQRM.

3.2 Broadsword Gatekeeper to Client Product Request Message (BGCPRM)

Token Name	Meaning or Use	Format	Max Length	Required
TRACE_KEY	Unique Gatekeeper generated key	A/N/S	Variable	Yes

The TRACE_KEY will be the first token in the BGCPRM message sent from the Gatekeeper to a plug-in. All other tokens will be the same as specified in the Client ICD. The TRACE_KEY serves the same purpose as in the BGCQM.

In addition to the TRACE_KEY, the following 4 tokens will be appended to the original BGCPRM message:

Token Name	Meaning or Use	Format	Max Length	Required
BSWD_KEY	Unique request key	A/N/S	54	Yes
ALT_DEST_ACCOUNT	Alternate destination account name	A/N	28	Yes
ALT_DEST_PASSWORD	Alternate destination password	A/N	28	Yes
ALT_DEST_IP	Alternate destination IP address	N/S	15	Yes
ALT_DEST_PATH	Alternate destination path	A/N/S	255	Yes

These tokens are used by the Gatekeeper's FTP daemon to perform store and forward capability for remote Gatekeepers. These tokens will usually be ignored by the plug-in. They will be used by the Remote Plug-in.

The plug-in must respond to the Gatekeeper with a request acknowledgement ACK for a successful product request initiation or a NACK if the plug-in cannot perform the request. The function AckNack() can be used to create the ACK or NACK message and send the message to the Gatekeeper. The format of the ACK and NACK messages are:

ACK

Token Name	Meaning or Use	Format	Max Length	Required
ACK	The value ACK	A	3	Yes

NACK

Token Name	Meaning or Use	Format	Max Length	Required
NACK	The value NACK	A	4	Yes
NACK_MSG	Optional error or notification message	A/N/S	Variable	No

In addition to the ACK or NACK, the plug-in must also send the Plug-in to Gatekeeper status update message. This message is sent to the Gatekeeper FTP/STATUS daemon to maintain the request security audit and perform image conversion, and ftp services for the request if the plug-in source is unable to perform these services. The plug-in will use the function SetRequestStatus() to perform this function.

3.3 Support Functions

The following functions are supplied for use by a plug-in.

a. SetTraceKey() ... Sets the unique trace key for the Gatekeeper syslog.

Calling Sequence:

```
void SetTraceKey(char *pKey, --- Pointer to the trace key to be set
                  int length --- length of the trace key
                  )
```

The trace key is the first token in any message received from the gatekeeper. After the plug-in has received a message, this function should be called to set the trace key. This will permit ease of trouble shooting should a problem arise with the gatekeeper or plug-in.

b. StartLog() ...Starts the system (syslog) to audit significant events during the processing of a message.

Calling Sequence:

```
void StartLog(char *pName) -- Process name (usually argv[0])
```

c. InitFuncLog() ...Communicates a new function name to the system audit

Calling Sequence:

```
void InitFuncLog(tFUNCTION_ERROR *pErr, --audit error structure
                 char *pFunction --Function name
                 )
```

d. WriteLogEntry() ...Writes a significant event to the system log

Calling Sequence:

```

void WriteLogEntry(tFUNCTION_ERROR *pErr, --audit error structure
                  int loc, --event location in
function
                  (__LINE__)
                  int error, --error code (errno)
                  char *pMessage, --text error message
                                describing event
                  char *pOptional, --additional message
                                may be NULL
                  int level --level of error or
event
                  )

```

e. SetRequestStatus() ... Formats and sends the STS message to the Gatekeeper

FTP/STATUS Daemon

Calling Sequence:

```

int SetRequestStatus(char *pIp, --Ftp Status daemon is running on this IP
                    int ipc_port, --and listening on this port
                    char *pStatus --set this status
                    char *pBswd_id, --Unique request id or key
                    int mrs_flag, --Update the request log True,False
                    int live, --TRUE if another SetRequestMessage will
be
                                sent shortly, else FALSE
                    char *pMessage, --Optional text message
                    int destination --Destination number of request update
                    char *pStart_time, --Not currently used
                    char *pEnd_time, --Not currently used
                    int numfiles, --Number of files in the request
                    double transize, --Byte size of the product file(s)
                    int alt_flag, --Set to 0
                    int s_mode --Set to 0
                    char *pTrace_key --Set to the trace key in original request
message
                    )

```

The function returns SUCCESS if the message was sent and acknowledged else FAILURE if the connection could not be made.

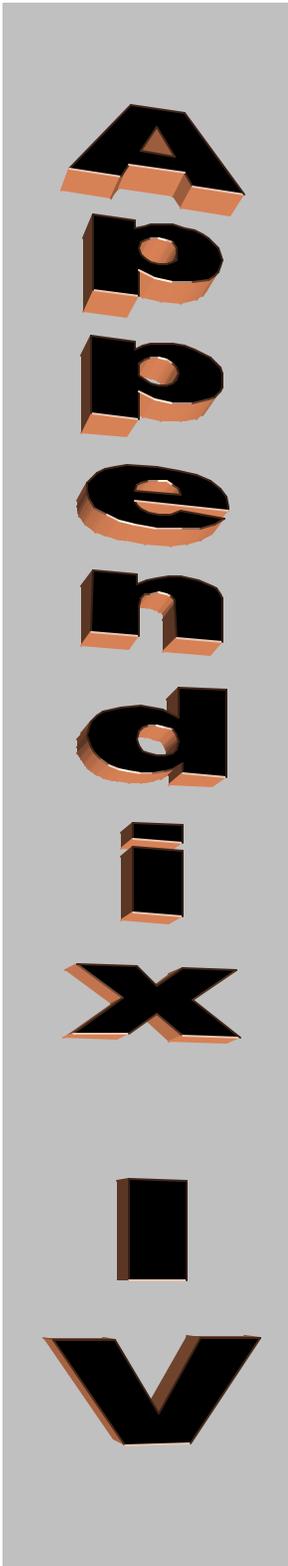
f. AckNack() ...Sends either the ACK or NACK message on the socket
supplied.

Calling Sequence:

```
int  AckNack(int      socket,      --socket number to send the message
          int      mode,      --either ACK or NACK
          char      *pMessage  --optional NACK message
      )
```

This function returns SUCCESS if the message was send successfully, else FAILURE if the BswdSendMessage failed.

Broadword ISSO to Gatekeeper API



The following tables describe the message sent between the ISSO API to the Gatekeeper to initiate ISSO functions and the Gatekeeper to ISSO API responses.

4.1 Broadsword Gatekeeper Audit Detail Record Request (BAMSG)

Token Name	Meaning or Use	Format	Max Length	Required
FUNCTION	Isso function AUDITR	A	6	Yes
USERNAME	Login id to select from audit	A/N	Variable	No
FROMDATE	Audit report start date (ZULU)	A/N/S	14	No
TODATE	Audit report end date (ZULU)	A/N/S	14	No
USEREVENT	Audit report event to select	A	Variable	No

4.2 Broadsword Gatekeeper Audit Extract Data Request (BAMSG)

Token Name	Meaning or Use	Format	Max Length	Required
FUNCTION	Isso function AUDITE	A	6	Yes
USERNAME	Login id to select from audit	A/N	Variable	No
FROMDATE	Audit report start date (ZULU)	A/N/S	14	No
TODATE	Audit report end date (ZULU)	A/N/S	14	No
USEREVENT	Audit report event to select	A	Variable	No
FILENAME	Audit extract file name	A/N/S	255	Yes

4.3 Broadsword Gatekeeper Audit Update Data Request (BAMSG)

Token Name	Meaning or Use	Format	Max Length	Required
FUNCTION	Isso function AUDITU	A	6	Yes
USER_NAME	Login id to select from audit	A/N	Variable	No
DATE_FROM	Audit update start date (ZULU)	A/N/S	14	No
DATE_TO	Audit update end date (ZULU)	A/N/S	14	No
EVENT	Audit update event to select	A	Variable	No

4.4 Broadsword Gatekeeper Audit Detail Record Request from Archive (BAMSG)

Token Name	Meaning or Use	Format	Max Length	Required
FUNCTION	Isso function AUDITRA	A	7	Yes
USERNAME	Login id to select from audit	A/N	Variable	No
FROMDATE	Audit report start date (ZULU)	A/N/S	14	No
TODATE	Audit report end date (ZULU)	A/N/S	14	No
USEREVENT	Audit report event to select	A	Variable	No

4.5 Broadsword Gatekeeper Audit Detail Response (BGADR)

This message is sent from the gatekeeper to the Audit API for the audit detail record request. The API will forward this message to the ISSO.

Token Name	Meaning or Use	Format	Max Length	Required
SESSION	Session record sub-message	A	N/A	Yes
SESSION_KEY	Unique session identifier	N	Variable	Yes
LOGIN_ACCOUNT	Gatekeeper login user name	A/N	28	Yes
LOGIN_HOST_IP	Login user host ip address	N/S	15	Yes
ORIGINATOR_WS	Originator login user name	A/N	28	Yes
ORIG_GKPR_IP	Assigned Gatekeeper ip address	N/S	15	Yes
EVENT_REC	Event record sub-message	A	N/A	Yes
EVENT_KEY	Unique event identifier	A/N/S	Variable	Yes
EVENT_DESC	Event description	A/N	Variable	Yes

Token Name	Meaning or Use	Format	Max Length	Required
EVENT_DTM	Event date time stamp (ZULU)	A/N	14	Yes
EVENT_DETAILS	Event details sub-message	A	N/A	Yes
EVENTDTM	Event detail time stamp (ZULU)	A/N	14	Yes
EVENT_INFO	Event text information	A/N/S	Variable	Yes
REFER	Gatekeeper or Source description	A/N	Variable	Yes

If there was a failure the following message will be sent from the gatekeeper to the audit API:

4.6 Broadsword Gatekeeper Status Response (BARSP)

Token Name	Meaning or Use	Format	Max Length	Required
STATUS	Status response from Gatekeeper	N	1	Yes

4.7 Broadsword Gatekeeper Status Response (BARSP)

Token Name	Meaning or Use	Format	Max Length	Required
STATUS	Status response from Gatekeeper	N	1	Yes
REC_COUNT	Extract record count	N	Variable	No
FILENAME	Audit extract filename	A/N/S	255	No

This message is sent to the admin API from the gatekeeper in response to the Audit Update Request.

4.8 Broadsword Gatekeeper Status Response (BARSP)

Token Name	Meaning or Use	Format	Max Length	Required
------------	----------------	--------	------------	----------

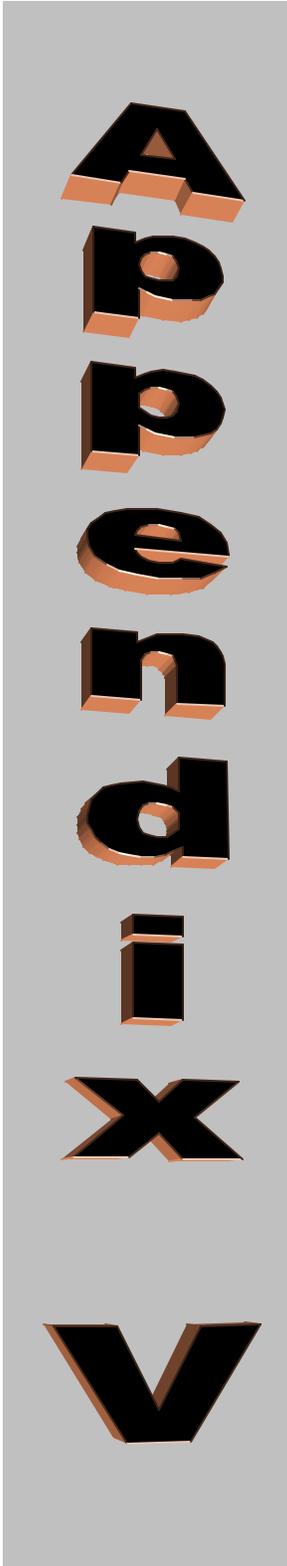
STATUS	Status response from Gatekeeper	N	1	Yes
REC_COUNT	Deleted record count	N	Variable	No

This message is sent to the admin API from the gatekeeper in response to the Audit List Archives request.

4.8 Broadsword Gatekeeper Status Response (BGARSP)

Token Name	Meaning or Use	Format	Max Length	Required
GKPR_REF	Gatekeeper reference	N/S	Variable	Yes
ARCHIVE	This token identifies the ARCHIVE sub-message. There may be 0 or more ARCHIVE sub-messages.	A	N/A	Yes
FILENAME	Archive file name	A/N/S	Variable	Yes
ARCHIVE_DATE	Zulu date when file was archived	N	14	Yes

API Function Definitions, ISSO and Admin Errors



5.1 API Function Definitions

The Broadsword ISSO to Gatekeeper API defines the “C” and “C++” functions to generate the various messages defined in this ICD. This API is comprised of the following functions:

- a. BswdAdminLock() ...This function must be called to initialize the Admin/Isso functions. The caller will require API_ADMIN_SUCCESS to access all ISSO functions. If API_READ_ONLY is returned only the Audit Detail Record Request function can be called.

- b. BswdAdminUnlock() ...This function must be called to release the admin lock. This will allow other administrators/ISSOS read and write permission to the audit records.

- c. BswdAdminGetAuditRpt() ...This function is used to get the Audit Detail Report.

- d. BswdAdminGetAuditExtract() ...This function is used to generate the Audit Extract file.

- e. BswdAdminGetAuditUpdate() ...This function is used to initiate the Audit Update

- f. BswdAdminGetAuditArchiveRpt() ...This function is used to get the Audit Detail Report from archive.

- g. BswdAdminListArchive() ...This function is used to obtain a list of audit archive file(s).

All functions return an integer status equal to SUCCESS (0) or FAILURE (non zero). Reasons for failure are detailed in the tADMIN_API_ERRORS enumeration. Note that this enumeration contains errors from the ISSO and ADMIN API.

5.1.1 BswdAdminLock()

Calling Sequence:

```
int BswdAdminLock(
    int          admin_socket,      -Socket number returned
                                         from the client connect
                                         response.
    char         *pBswd_user,      -Admin user account name
    char         *pBswd_ip,       -Admin user ip address
    tADMIN_INFO *pAdmin_info      -Pointer to structure
                                         containing administrator
                                         information.
```

```
typedef struct
{
    int lock_flag;      - Tracks the locking flag
    int socket;        - Administrators socket
}tADMIN_INFO, *pADMIN_INFO;
```

5.1.2 BswdAdminUnlock()

Calling Sequence:

```
int BswdAdminUnlock(
    pADMIN_INFO pAdmin_info- Pointer to tADMIN_INFO structure
                                         returned by
                                         BswdAdminLock().
```

5.1.3 BswdAdminGetAuditRpt()

Calling Sequence:

```
int BswdAdminGetAuditRpt(
    tADMIN_INFO *pAdmin_info,      Returned by
    BswdAdminLock().
    char *pUser_name,             NULL or Bswd login account
    char *pDate_from,             NULL or ZULU start date.
    char *pDate_to,               NULL or ZULU end date.
    char *pUserEvent,             NULL or audit event
    description. char **hAudit_rsp BVL MF
    response
```

)

5.1.4 BswdAdminGetAuditExtract()

Calling Sequence:

int	BswdAdminGetAuditExtract(tADMIN_INFO	*pAdmin_info,	Returned by
BswdAdminLock().	char	*pUser_name,	NULL or Bswd login account
	char	*pDate_from,	NULL or ZULU start date.
	char	*pDate_to,	NULL or ZULU end date.
	char	*pUserEvent,	NULL or audit event
description.	char	*pFile_name,	File name for
extract			
int		*pRec_count	# of records archived
)	

5.1.5 BswdAdminGetAuditUpdate()

Calling Sequence:

int	BswdAdminGetAuditUpdate(tADMIN_INFO	*pAdmin_info,	Returned by
BswdAdminLock().	char	*pUser_name,	NULL or Bswd login account
	char	*pDate_from,	NULL or ZULU start date.
	char	*pDate_to,	NULL or ZULU end date.
	char	*pUserEvent,	NULL or audit event
description.			
Int		*pRec_count	# of records deleted
)	

5.1.6 BswdAdminGetAuditArchiveRpt()

Calling Sequence:

int	BswdAdminGetAuditArchiveRpt(tADMIN_INFO	*pAdmin_info,	Returned by
BswdAdminLock().	char	*pUser_name,	NULL or Bswd login account

char	*pDate_from,	NULL or ZULU start date.
char	*pDate_to,	NULL or ZULU end date.
char	*pUserEvent,	NULL or audit event
description.		
PARCHIVE_FILES	pArchive_files,	NULL or linked list of
archive files		
char	**hAudit_rsp	BVLMF response
)	

5.1.7 BswdAdminListArchive()

Calling Sequence:

```
int    BswdAdminListArchive(
        tADMIN_INFO    *pAdmin_info,    Returned by
BswdAdminLock().
        char            **hArchive_rsp    BGARSP response message.
    )
```

```
typedef struct ARCHIVE_FILES    tARCHIVE_FILES;
typedef struct ARCHIVE_FILES    *pARCHIVE_FILES;
struct ARCHIVE_FILES
{
    char            *pArchive_file;
    pARCHIVE_FILES pNext_file;
}
```

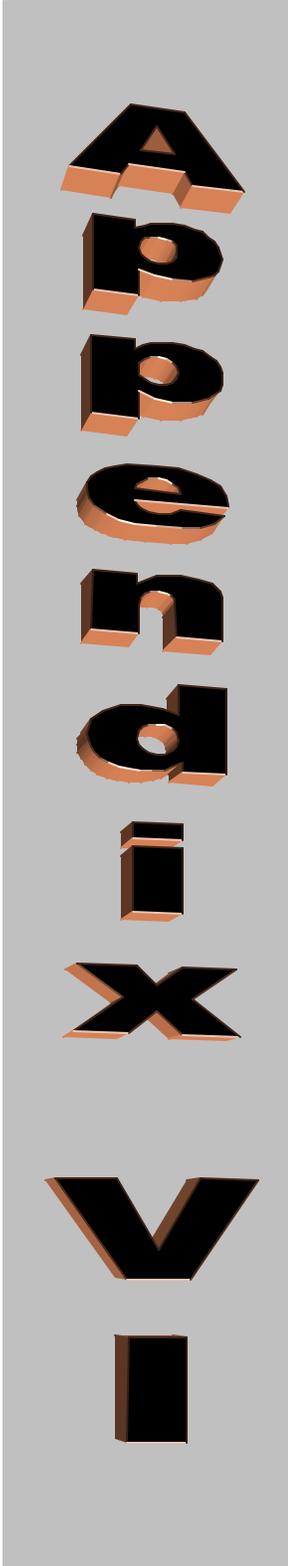
5.2 ISSO and ADMIN API ERRORS

```
typedef enum
{
    API_ADMIN_FAILURE = -1    /* any failure */
    API_ADMIN_SUCCESS,        /* function was successful */
    API_ADMIN_READ_ONLY,     /* user has READ only permissions
*/
    API_ADMIN_PERMISSION_DENIED, /* user does not have ADMIN/ISSO
Permissions.
    API_ADMIN_IPC_WRITE,     /* there was a socket write failure */
    API_ADMIN_IPC_READ,      /* there was a socket read failure */
    API_ADMIN_UNKNOWN_PLUGIN, /* unknown plug-in name */
    API_ADMIN_SRCREF_EXISTS,  /* create a new source failure */
    API_ADMIN_UNKNOWN_FUNCTION, /* unknown ADMIN/ISSO function
type */
```

API_ADMIN_INVALID_REFERENCE, /* invalid gkpr or source reverence */
API_ADMIN_FILE_EXISTS, /* Audit archive file exists */
API_ADMIN_INVALID_FILENAME, /* Audit archive file must NOT include
path */
API_ADMIN_INVALID_GRP_NAME, /* Group name not found */
API_ADMIN_INVALID_PARAM_NAME, /* Param name not found in template
for
SetSrcParam */
API_ADMIN_INVALID_MODE, /* The value for MODE is invalid */
API_ADMIN_NULL_PARAM, /* required parameter is NULL */
API_ADMIN_INVALID_USER, /* User name supplied not valid */
API_ADMIN_REG_BUSY, /* Gkpr registration active by
another
Process */
API_ADMIN_NOT_KM, /* Attempted gkpr registration on a
gkpr which is not setup as a key-master
*/
API_ADMIN_REG_PENDING, /* Initial registration still pending */
API_ADMIN_REG_ABORTED, /* Initial registration aborted by
client */
API_ADMIN_REG_GKPR_CONNECT, /* Could not connect to remote KM
*/
API_ADMIN_REG_NO_PUBLICKEY, /* Error accessing public-key file */
API_ADMIN_REG_NO_CERTREQ, /* Error accessing cert reg file */
API_ADMIN_REG_NO_CERT, /* Error accessing certificate file */
API_ADMIN_REG_PWD, /* Registration password incorrect */
API_ADMIN_REG_PWD_WARN, /* password is incorrect, may retry */
API_ADMIN_REG_REFUSED, /* Registration failed by KM */
API_ADMIN_REG_TO_ITSELF, /* Registration to own gkpr is wrong
*/
API_ADMIN_REG_MISSING_POC, /* Registration is missing poc
information */
API_ADMIN_REG_TO, /* Registration password timed out
*/
API_ADMIN_NOT_START_ALPHA, /* Registration field doesn't start
with an alpha character */
API_ADMIN_NO_SRCS_DEFINED, /* GetElements needs source(s)
defined */
API_ADMIN_USERUPD_NOT_ALLOWED, /* Solaris users not allowed to
add/ delete users */
API_ADMIN_CATALOG_NOT_SUPP, /* source doesn't support catalog */
API_ADMIN_NAME_EXISTS, /* Username already exists */

```
API_ADMIN_LDAP_ERROR,          /* Error attempting to bind to the
LDAP                               server */
API_ADMIN_KM_WRONG,            /* cannot register a bswd3.0 gkpr
with a                               bswd2.0 Keymaster. */
} tADMIN_API_ERRORS;
```

Broadword Admin to Gatekeeper API



The following tables describe the message sent between the Admin API to the Gatekeeper to initiate Admin functions and the Gatekeeper to Admin API responses.

6.1 Broadsword Admin Get Plug-in Names (BAMSG)

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, PLNAMES	A/N	7	Yes

6.2 Broadsword Admin Get Template (BAMSG)

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, GETTEM	A/N	6	Yes
PLUGIN_NAME	Plug-in name passed by API	A/N	Variable	No

6.3 Broadsword Admin Get Sources (BAMSG)

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, GETSRCS	A/N	7	Yes

6.4 Broadsword Admin Get Information (BAMSG)

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, GETINFO	A/N	7	Yes
SOURCEREF	Source reference if source	A/N	Variable	No

6.5 Broadword Admin Create Backside Source (BAMSG)

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, CREATSRC	A/N	8	Yes
PLUGIN_NAME	Name returned from GetPluginNames	A/N	Variable	Yes
SOURCEDESC	Source Description	A/N	Variable	Yes

6.6 Broadword Admin Delete Backside Source (BAMSG)

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, DELETESRC	A/N	9	Yes
SOURCEREF	Source reference of source to delete	A/N/S	Variable	Yes

6.7 Broadword Admin Set Backside Source Parameter (BAMSG)

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, SETSRCP	A/N	7	Yes
SOURCEREF	Source or gatekeeper reference	A/N/S	Variable	Yes
PARAMNAME	The parameter to update	A/N	Variable	Yes
PARAMVALUE	The parameter value	A/N/S	Variable	Yes

6.8 Broadword Admin Get Groups (BAMSG)

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, GETGRPS	A/N	7	Yes

6.9 Broadsword Admin Get Group Members (BAMSG)

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, GETGRPM	A/N	7	Yes
NAME	The group name	A/N	Variable	Yes

6.10 Broadsword Admin Add Group Member (BAMSG)

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, ADDGRPM	A/N	7	Yes
GROUPNAME	The group name	A/N	Variable	Yes
MEMBER	Group member to add	A/N	Variable	Yes

6.11 Broadsword Admin Delete Group Member (BAMSG)

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, DELGRPM	A/N	7	Yes
GROUPNAME	The group name	A/N	Variable	Yes
MEMBER	Group member to delete	A/N	Variable	Yes

6.12 Broadsword Admin Delete Group (BAMSG)

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, DELGRP	A/N	6	Yes
GROUPNAME	The group name to delete	A/N	Variable	Yes

6.13 Broadsword Admin Add New Group (BAMSG)

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code,	A/N	6	Yes

	ADDGRP			
GROUPNAME	The group name to delete	A/N	Variable	Yes

6.14 Broadsword Admin Get Discretionary Access Control (BAMSG)

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, GETDAC	A/N	6	Yes
REF	The gatekeeper or source ref	A/N/S	Variable	Yes

6.15 Broadsword Admin Add Discretionary Access Control (BAMSG)

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, ADDDAC	A/N	6	Yes
SOURCEREF	The gatekeeper or source ref	A/N/S	Variable	Yes
ALLOW	User/group or ALL	A/N	Variable	No
DENY	User/group or ALL	A/N	Variable	No

6.16 Broadsword Admin Delete Discretionary Access Control (BAMSG)

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, DELDAC	A/N	6	Yes
SOURCEREF	The gatekeeper or source ref	A/N/S	Variable	Yes
ALLOW	User/group or ALL to delete	A/N	Variable	No
DENY	User/group or ALL to delete	A/N	Variable	No

6.17 Broadsword Admin Get Users

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, GETUSERS	A/N	8	Yes
ROLE	Type of information to retrieve	N	1	Yes
SOURCEREF	The source ref for producers	A/N/S	Variable	No

6.18 Broadsword Admin Add User

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, ADDUSER	A/N	7	Yes
ROLE	Type of user to add	N	1	Yes
SOURCEREF	The source ref for producers	A/N/S	Variable	No
NAME	User name to add	A/N	Variable	Yes

6.19 Broadsword Admin Delete User

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, DELUSER	A/N	7	Yes
ROLE	Type of user to delete	N	1	Yes
SOURCEREF	The source ref for producers	A/N/S	Variable	No
NAME	User name to delete	A/N	Variable	Yes

6.20 Broadsword Admin Modify Element Name

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, MODELL	A/N	6	Yes
SOURCENAME	Bswd source name.	A/N	6	No
SECTION_NAME	The section name or component for this DATA_NAME	A/N	Variable	Yes
DATA_NAME	Data name.	A/N	Variable	Yes
DATA_HELP	Data help text.	A/N/S	Variable	No
DISPLAY_NAME	Data display name.	A/N	Variable	No
MODE	Pick-list, data-range mode.(A,D)	A	1	No
DATA_RANGE	Data range to modify.	A/N/S	Variable	No
DATA_LIST	Data list to modify.	A/N/S	Variable	No

6.21 Broadword Admin Set User Discretionary Access Control

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, SETUDAC	A/N	6	Yes
SOURCEREF	Back-side source or gatekeeper ref.	A/N	47	Yes
NAME	User or group name.	A/N	Variable	Yes
MODE	Allow (A) or Deny(D)	A	1	Yes

6.22 Broadword Admin Get User Discretionary Access Control

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, GETUDAC	A/N	6	Yes
NAME	User or group name.	A/N	Variable	Yes

6.23 Broadword Admin Initialize Registration

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, INITREG	A/N	7	Yes

6.24 Broadword Admin Initialize Registration Status

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, ABORTREG	A/N	8	Yes

6.25 Broadword Admin Register Our Gatekeeper With Keymaster

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, REGOURGKPR	A/N	10	Yes

CC	Country Code	A	2	Yes
STATE	State	A	Variable	Yes
ORGNAME	Organization name	A/N	Variable	Yes
POC	Point of contact	A/N	Variable	Yes
POC_EMAIL	Poc email address	A/N/S	Variable	Yes
GKPRIPADDR	Register with this gatekeeper IP address	N/S	15	Yes
GKPRIPCPORT	Register with this gatekeeper TCP/IP port	N	6	Yes
REG_PASSWORD	The one-time registration password	A/N/S	Variable	Yes

6.26 Broadword Admin Generate Public Key

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, GENPUBKEY	A/N	9	Yes

6.27 Broadword Admin Get Keymasters

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, KEYMASTERS	A/N	10	Yes

6.28 Broadword Admin Get Audit Events

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, AUDITEVENTS	A	11	Yes

6.29 Broadword Admin Get Connection Response Message

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Admin function code, GETCRM	A	6	Yes

Note: The response for this message is the BGCRM.

If there was a failure in any of the above messages, the following message will be sent from the gatekeeper to the Admin API:

6.30 Broadsword Gatekeeper Status Response (BARSP)

Token Name	Meaning or Use	Format	Max Length	Required
STATUS	Status response from Gatekeeper. (1) = Failure, (0) = SUCCESS	N	1	Yes

6.31 Broadsword Admin Gatekeeper Plug-in Name Response (BAGPN)

This message is sent from the gatekeeper to the Admin API for the Get Plug-in Names request. The API will forward this message to the Admin.

Token Name	Meaning or Use	Format	Length	Required
PLUGIN_NAME	Installed plug-in name	A/N	A/N	Yes

6.32 Broadsword Admin Gatekeeper Get Template Response (BAGT)

This message is sent from the gatekeeper to the Admin API for the Get Template request. The API will forward this message to the Admin.

Token Name	Meaning or Use	Format	Length	Required
SOURCENAME	Source or Gatekeeper	A	Variable	Yes
GKPRDESC	BSWD 1.0 Gatekeeper description	A/N	Variable	Yes
DISPLAYNAME	Display name of GKPRDESC	A/N	Variable	No
DATAHELP	Text to describe the GKPRDESC token	A/N	Variable	No
GKPRPUBLICKEY	Gatekeeper Public Key	A/N/S	Variable	Yes
GKPRIPADDR	Gatekeeper IP address	A/N/S	15	Yes
GKPRIPPORT	Gatekeeper IP port number	N	6	Yes
GKPRCDIMUSER	Account for CDIM	A/N/S	Variable	Yes
GKPRCDIMPWD	CDIM password	A/N/S	Variable	Yes
GKPRLOGINTIMEOUT	Gatekeeper user login timeout	N	Variable	Yes

GKPRALTACCOU NT	Gatekeepers Alternate transfer Account	A/N/S	Variable	Yes
--------------------	---	-------	----------	-----

6.33 Broadword Admin Gatekeeper Get Sources Response (BAGS)

This message is sent from the gatekeeper to the Admin API for the Get Sources request. The API will forward this message to the Admin.

Token Name	Meaning or Use	Format	Length	Required
GATEKEEPER	This token begins the Gatekeeper sub-message.	A	N/A	Yes
TYPE	Denotes the gatekeeper as (A) Assigned or (R) remote.	A	1	Yes
GKPRREF	World unique gatekeeper ID	A/N/S	25	Yes
GKPRDESC	Site gatekeeper description	A/N	Variable	Yes
ACCESS_PERMS	This token begins the DAC sub-message for this gatekeeper	A	11	Yes
DENY	This token identifies the DENY sub-message	A	N/A	No
NAME_REC	This token identifies the NAME_REC sub-message	A	N/A	Yes
NAME	Group or user name or the word "ALL"	A/N/S	Variable	Yes
TYPE	Denotes name as a user (U), (G) group or (A) all. Default is U.	A	1	No
ALLOW	This token identifies the ALLOW sub-message.	A	N/A	No
NAME_REC	This token identifies the NAME_REC sub-message	A	N/A	Yes
NAME	Group, user or "ALL" name	A/N/S	Variable	Yes
TYPE	Denotes name as a user (U), (G) group or (A) all. Default is U.	A	1	No
SOURCE	This token begins the Source sub-message	A	N/A	Yes
SOURCEREF	World unique source ID	A/N/S	47	Yes
SOURCENAME	Site source name	A/N/S	Variable	Yes
SOURCEDESC	Site source description	A/N/S	Variable	Yes
ACCESS_PERMS	This token begins the DAC sub-message for this source	A	11	Yes
DENY	This token identifies the DENY sub-message	A	N/A	No
NAME_REC	This token identifies the	A	N/A	Yes

	NAME_REC sub-message			
NAME	Group or user name or the word "ALL"	A/N/S	Variable	Yes
TYPE	Denotes name as a user (U), (G) group or (A) all. Default is U.	A	1	No
ALLOW	This token identifies the ALLOW sub-message.	A	N/A	No
NAME_REC	This token identifies the NAME_REC sub-message	A	N/A	Yes
NAME	Group, user or "ALL" name	A/N/S	Variable	Yes
TYPE	Denotes name as a user (U), (G) group or (A) all. Default is U.	A	1	No

6.34 Broadword Admin Gatekeeper Get Info Response (BAGI)

This message is sent from the gatekeeper to the Admin API for the Get Source Information request. The API will forward this message to the Admin.

Token Name	Meaning or Use	Format	Length	Required
SOURCENAME	Source name	A/N	Variable	Yes
SOURCEDESC	Source description	A/N/S	Variable	Yes
SOURCEREF	World unique source ID	A/N/S	47	Yes
QRYPLUGINNAME	Plug-in used for query.	A/N/S	Variable	Yes
RQSTPLUGINNAME	Plug-in used for request	A/N/S	Variable	Yes
SERVERTYPE	Backside source type (IPL)	A/N/S	Variable	Yes
SITENAME	Backside source site name	A/N/S	Variable	Yes
HOSTID	IP address of backside source	A/N/S	15	Yes
PCRPORT	IP port number of backside source	N	6	Yes
IPLADDRESS	IP address for message xlate	A/N/S	15	Yes
LOGIN	Login user id for backside source	A/N/S	Variable	Yes
PASSWORD	Login user password for backside source	A/N/S	Variable	Yes
HARVESTPORT	IP port number of search engine	N	6	Yes

6.35 Broadsword Admin Gatekeeper Get Groups Response (BAGGP)

This message is sent from the gatekeeper to the Admin API for the Get Groups request. The API will forward this message to the Admin.

Token Name	Meaning or Use	Format	Length	Required
GROUPNAME	The group name	A/N	Variable	Yes

6.36 Broadsword Admin Gatekeeper Get Group Members Response (BAGGM)

This message is sent from the gatekeeper to the Admin API for the Get Members request. The API will forward this message to the Admin.

Token Name	Meaning or Use	Format	Length	Required
NAME_REC	This token identifies the NAME_REC sub-message	A	N/A	Yes
NAME	Group or user name	A/N/S	Variable	Yes
TYPE	Denotes name as a user (U) or (G) group. Default is U.	A	1	No

6.37 Broadsword Admin Gatekeeper Get Discretionary Access Control Response (BAGD)

This message is sent from the gatekeeper to the Admin API for the Get Discretionary Access Control request. The API will forward this message to the Admin.

Token Name	Meaning or Use	Format	Length	Required
DENY	This token identifies the DENY sub-message	A	N/A	No
NAME_REC	This token identifies the NAME_REC sub-message	A	N/A	Yes
NAME	Group or user name or the word "ALL"	A/N/S	Variable	Yes
TYPE	Denotes name as a user (U), (G) group or (A) all. Default is U.	A	1	No
ALLOW	This token identifies the ALLOW sub-message.	A	N/A	No
NAME_REC	This token identifies the NAME_REC sub-message	A	N/A	Yes
NAME	Group, user or "ALL" name	A/N/S	Variable	Yes
TYPE	Denotes name as a user (U), (G) group or (A) all. Default is U.	A	1	No

6.38 Broadsword Admin Gatekeeper Get Users Response (BAGPD)

This message is sent from the gatekeeper to the Admin API for the Get Users request. The API will forward this message to the Admin.

Token Name	Meaning or Use	Format	Length	Required
NAME_REC	This token identifies the NAME_REC sub-message	A	N/A	Yes
NAME	Group or user name	A/N/S	Variable	Yes
TYPE	Denotes name as a user (U) or (G) group. Default is U.	A	1	No

6.39 Broadsword Admin Gatekeeper Get User Discretionary Access Control (BAGUD)

This message is sent from the gatekeeper to the Admin API for the Get User Discretionary Access Control request. The API will forward this message to the Admin.

Token Name	Meaning or Use	Format	Length	Required
GATEKEEPER	This token begins the first BVLMF sub-message. The message defines the applicable gatekeeper and connected backside sources.	A	N/A	Yes
TYPE	Denotes the gatekeeper as (A) Assigned or (R) Remote	A	1	Yes
GKPR_REF	World unique gatekeeper ID	A/N/S	25	Yes
GKPR_DESC	Site gatekeeper description	A/N	Variable	Yes
SOURCE	This token begins the second BVLMF sub-message. This message defines a backside source. There will be at least one source.	A	N/A	Yes
SOURCE_REF	World unique source reference ID	A/N/S	47	Yes
SOURCE_NAME	A source type abbreviation for the backside source	A/N	6	Yes

SOURCE_DESC	Description of the source	A/N	Variable	Yes
-------------	---------------------------	-----	----------	-----

6.40 Broadword Admin Gatekeeper Initialize Registration Response (BAIR)

This message is sent from the gatekeeper to the Admin API for the Initialize Registration request. The API will forward this message to the Admin.

Token Name	Meaning or Use	Format	Length	Required
GKPRIPADDR	Gatekeeper IP address.	N/S	15	Yes
GKPRIPCPOR	TCP/IP port gatekeeper will listen for connections	N	6	Yes
REG_PASSWORD	Unique one time registration password	A/N	Variable	Yes

6.42 Broadword Admin Gatekeeper Get Key-masters Response (BAGKM)

This message is sent from the gatekeeper to the Admin API for the Get Key-masters request. The API will forward this message to the Admin.

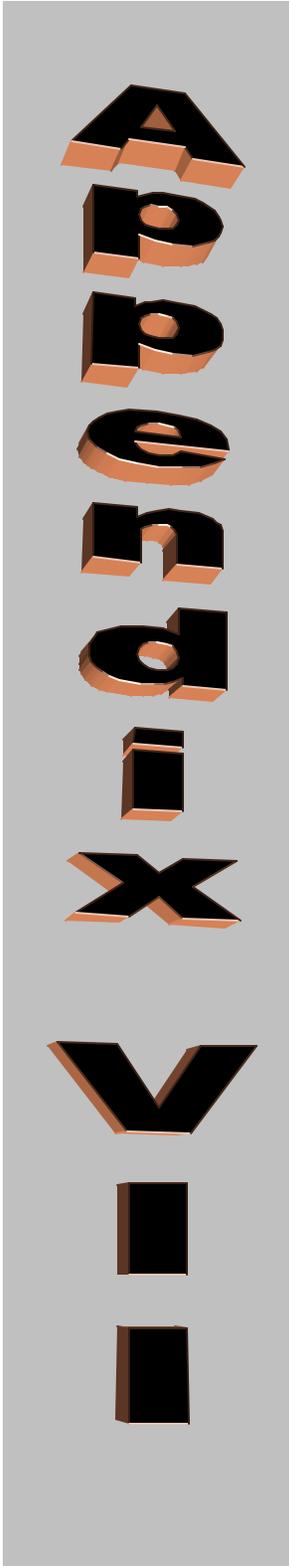
Token Name	Meaning or Use	Format	Length	Required
KEYMASTER	This token identifies the KEYMASTER sub-message.	A	9	Yes
GKPRREF	This token identifies the gatekeeper of a key-master.	A/N/S	25	Yes
KEYMASTER_STATUS	Registration status of this key-master.	A/N	Variable	Yes

6.43 Broadsword Admin Gatekeeper Get Audit Events Response (BAGAE)

This message is sent from the gatekeeper to the Admin API for the Get Audit Events request. The API will forward this message to the Admin.

Token Name	Meaning or Use	Format	Length	Required
EVENT	This token identifies the EVENT sub-message.	A	5	Yes
EVENT_NAME	This token identifies the name of the event.	A/N	Variable	Yes
EVENT_DESC	This token identifies the event description	A/N	Variable	Yes

Admin API Function Definitions



7.1 Admin API Function Definitions

The Broadsword Admin to Gatekeeper API defines the “C” and “C++” functions to generate the various messages defined in this ICD. This API is comprised of the following functions.

- a. BswdAdminGetPluginNames() ...This function will return the names of all installed plug-ins.
- b. BswdAdminGetTemplate ...This function will return the gatekeeper or plug-in specific parameters needed to conTable a source of this type.
- c. BswdAdminGetSrcs() ...This function will return a list of all sources currently conTabled.
- d. BswdAdminGetInfo() ...This function will return all configuration information for a the local gatekeeper or a particular source.
- e. BswdAdminCreateSrc() ...This function will create a NEW source for the plug-in name passed. It does not set any of the plug-in specific data.
- f. BswdAdminDeleteSrc() ...This function will delete a source based on the source reference passed.
- g. BswdAdminSetSrcParam() ...This function will allow setting a parameter to a passed value.
- h. BswdAdminGetGrps() ...This function will return all group names.
- i. BswdAdminGetGrpMembers() ...This function will retrieve members of a group.
- j. BswdAdminAddGrpMember() ...This function will add a member to a group. The member may be another group name.
- k. BswdAdminDelGrpMember() ...This function will delete a member from a group. The member may be another group name.
- l. BswdAdminDeleteGrp() ...This function will delete a group.
- m. BswdAdminAddGrp() ...This function will add a group.
- n. BswdAdminGetDAC() ...This function will retrieve the Discretionary Access Control permissions for a particular gatekeeper or backside source.

- o. BswdAdminAddDAC() ...This function will set the Discretionary Access Control permissions for a particular gatekeeper or backside source.
- p. BswdAdminDelDAC() ...This function will remove the Discretionary Access Control permissions for a particular gatekeeper or backside source.
- q. BswdAdminSetUserDAC() ...This function will set the DAC for a user to a source or gatekeeper.
- r. BswdAdminGetUserDAC() ...This function will get the DAC for a user.
- s. BswdAdminGetUsers() ...This function will get the valid producers, admin users or isso for the local gatekeeper or a particular source reference.
- t. BswdAdminAddUsers() ...This function will add a valid producer for a particular source reference, or an admin or isso user to the local gatekeeper.
- u. BswdAdminDelUser() ...This function will delete a valid producer for a particular source reference, or an admin or isso user from the local gatekeeper.
- v. BswdAdminModElementName() ...This function will modify the parameters of a DATA_NAME.
- w. BswdAdminInitReg() ...This function must be called to initiate a new registration of a gatekeeper. This function will only be done at gatekeepers designated as (CA) certificate authorities.
- x. BswdAdminInitRegStatus() ...This function will either receive registration status or cause a registration to be aborted.
- y. BswdAdminRegOurGkpr() ...This function is used by the new gatekeeper to communicate the gatekeeper's ip address, port and one time password to the CA.
- z. BswdAdminGetAuditEvents() ... This function returns the list of auditable events.
- aa. BswdAdminUnregGkpr() ... This function will unregister a gatekeeper from a keymaster.
- bb. BswdAdminGetKeymasters() ...This function will return the gatekeeper references and registration status of all key-masters that this gatekeeper is registered with.

- cc. BswdAdminGetConnRspMsg() ... This function will return the BGCRM
- dd. BswdAdminGetUserInfo() ... This function will get all user information stored in the LDAP server.
- ee. BswdAdminSetUserInfo() ... This function will set all user information stored in the LDAP server.
- ff. BswdAdminModGrp() ... This function will change a group description.
- gg. BswdAdminClearStats() ... This function will clear the gatekeeper statistics database.
- hh. BswdAdminGetAdminErrMsg() ... This function will translate an admin API error code to a text message.
- ii. BswdAdminSetRemGkrp() ... This function allows an administrator at a keymaster to change a remote gatekeeper's user information.

7.1.1 BswdAdminGetPluginNames()

Calling Sequence:

```
int BswdAdminGetPluginNames(  
    pADMIN_INFO      pAdmin_info, - Pointer to tADMIN_INFO returned  
                                by BswdAdminLock,  
    char             **hPlugin_names - Pointer to admin name list BVLMF  
                                message.  
)
```

RETURNS: API_ADMIN_SUCCESS -successful, phPlugin_names points to response.
API_ADMIN_NULL_PARAM -Required parameter is NULL
API_ADMIN_FAILURE -Some failure, hPlugin_names is null
API_ADMIN_PERMISSION_DENIED -Caller doesn't have ADMIN permission.
API_ADMIN_IPC_READ -Socket read failure
API_ADMIN_IPC_WRITE -Socket write failure
ADMIN permission.
API_ADMIN_IPC_READ -Socket read failure
API_ADMIN_IPC_WRITE -Socket write failure

7.1.2 BswdAdminGetTemplate()

Calling Sequence:

```
int BswdAdminGetTemplate(  
)
```

pADMIN_INFO pAdmin_info, - Pointer to tADMIN_INFO returned
by BswdAdminLock,
char *pPlugin_name, - Plugin name returned from
BswdAdminGetPluginNames or
NULL for gatekeeper template.
char **hTemplate_rsp - Pointer to template BVLMF
message.
)

RETURNS: API_ADMIN_SUCCESS - Function was successful,
hTemplate_rsp points to response.
API_ADMIN_NULL_PARAM -Required parameter is NULL
API_ADMIN_FAILURE - Some failure, hTemplate_rsp is
NULL.
API_ADMIN_PERMISSION_DENIED -Caller doesn't have
ADMIN permission.
API_ADMIN_IPC_READ -Socket read failure
API_ADMIN_IPC_WRITE -Socket write failure
API_UNKNOWN_PLUGIN -Unknown plug-in name

7.1.3 BswdAdminGetSrcs()

Calling Sequence:

int BswdAdminGetSrcs(
pADMIN_INFO pAdmin_info, - Pointer to tADMIN_INFO returned
by BswdAdminLock,
char **hSrcs_rsp - Pointer to sources BVLMF
message.
)

RETURNS: API_ADMIN_SUCCESS - Function was successful, hSrcs_rsp
points to response.
API_ADMIN_NULL_PARAM -Required parameter is NULL
API_ADMIN_FAILURE - Some failure, hSrcs_rsp is NULL.
API_ADMIN_PERMISSION_DENIED -Caller doesn't have

7.1.4 BswdAdminGetInfo()

Calling Sequence:

int BswdAdminGetInfo(
pADMIN_INFO pAdmin_info, - Pointer to tADMIN_INFO returned

char *pSrc_ref, by BswdAdminLock,
- The source reference of the source,
or NULL for local gatekeeper.
char **hSrcs_info_rsp - Pointer to source info BVLMF
message.
)

RETURNS: API_ADMIN_SUCCESS - Function was successful,
hSrcs_info_rsp points to response.
API_ADMIN_NULL_PARAM - Required parameter is NULL
API_ADMIN_FAILURE - Some failure, hSrcs_info_rsp is
NULL.
API_ADMIN_PERMISSION_DENIED - Caller doesn't have
ADMIN permission.
API_ADMIN_IPC_READ - Socket read failure
API_ADMIN_IPC_WRITE - Socket write failure
API_ADMIN_INVALID_REFERENCE - Invalid source reference

7.1.5 BswdAdminCreateSrc()

Calling Sequence:

```
int BswdAdminCreateSrc(  
    pADMIN_INFO pAdmin_info, - Pointer to tADMIN_INFO returned  
    by BswdAdminLock,  
    char *pPlugin_name, - Name returned from  
    BswdAdminGetPluginNames.  
    char *pSrc_desc, - The description of the source.  
    char **hSrc_ref - Upon successful create, the NEW  
    source reference is returned.  
)
```

RETURNS: API_ADMIN_SUCCESS - Function was successful, hSrc_ref
points to new source reference.
API_ADMIN_NULL_PARAM - Required parameter is NULL
API_ADMIN_FAILURE - Some failure, hSrc_ref is NULL.
API_ADMIN_PERMISSION_DENIED - Caller doesn't have
ADMIN permission.
API_ADMIN_IPC_READ - Socket read failure
API_ADMIN_IPC_WRITE - Socket write failure
API_UNKNOWN_PLUGIN - Unknown plug-in name

7.1.6 BswdAdminDeleteSrc()

Calling Sequence:

```
int BswdAdminDeleteSrc(  
    pADMIN_INFO    pAdmin_info, - Pointer to tADMIN_INFO returned  
                                by BswdAdminLock,  
    char            *pSrc_ref    - The source reference of the source  
                                to delete.  
)
```

RETURNS: API_ADMIN_SUCCESS - Function was successful, source is deleted.
API_ADMIN_NULL_PARAM -Required parameter is NULL
API_ADMIN_FAILURE - Some other failure.
API_ADMIN_PERMISSION_DENIED -Caller doesn't have ADMIN permission.
API_ADMIN_IPC_READ -Socket read failure
API_ADMIN_IPC_WRITE -Socket write failure
API_ADMIN_INVALID_REFERENCE -Invalid source reference

7.1.7 BswdAdminSetSrcParam()

Calling Sequence:

```
int BswdAdminSetSrcParam(  
    pADMIN_INFO    pAdmin_info, - Pointer to tADMIN_INFO returned  
                                by BswdAdminLock,  
    char            *pSrc_ref    - The reference of the source or  
                                the gatekeeper.  
    pPARAM_INFO *pParam_name    - A linked list containing the  
                                Param name and associated  
                                Param value to change.  
)
```

RETURNS: API_ADMIN_SUCCESS - Function was successful, source is updated.
API_ADMIN_NULL_PARAM -Required parameter is NULL
API_ADMIN_FAILURE - Some other failure.
API_ADMIN_PERMISSION_DENIED -Caller doesn't have ADMIN permission.
API_ADMIN_IPC_READ -Socket read failure
API_ADMIN_IPC_WRITE -Socket write failure
API_ADMIN_INVALID_REFERENCE -Invalid source reference

7.1.8 BswdAdminGetGrps()

Calling Sequence:

```
int BswdAdminGetGrps(  
    pADMIN_INFO      pAdmin_info, - Pointer to tADMIN_INFO returned  
                                by BswdAdminLock,  
    char              **hGrps_rsp  - The group response BVLMF  
                                message.  
                                )
```

RETURNS: API_ADMIN_SUCCESS - Function was successful,
 hGrps_rsp points to the response.
 API_ADMIN_NULL_PARAM -Required parameter is NULL
 API_ADMIN_FAILURE - Some other failure, hGrps_rsp is
 NULL.
 API_ADMIN_PERMISSION_DENIED -Caller doesn't have
 ADMIN permission.
 API_ADMIN_IPC_READ -Socket read failure
 API_ADMIN_IPC_WRITE -Socket write failure

7.1.9 BswdAdminGetGrpMembers()

Calling Sequence:

```
int BswdAdminGetGrpMembers(  
    pADMIN_INFO      pAdmin_info, - Pointer to tADMIN_INFO returned  
                                by BswdAdminLock,  
    char              *pGrp_name,  - The group name  
    char              *hGrp_det_rsp - The group detail response BVLMF  
                                message.  
                                )
```

RETURNS: API_ADMIN_SUCCESS - Function was successful,
 hGrp_det_rsp points to the
 response.
 API_ADMIN_NULL_PARAM -Required parameter is NULL
 API_ADMIN_FAILURE - Some other failure, hGrp_det_rsp is
 NULL.
 API_ADMIN_PERMISSION_DENIED -Caller doesn't have
 ADMIN permission.
 API_ADMIN_IPC_READ -Socket read failure

API_ADMIN_IPC_WRITE -Socket write failure

7.1.10 BswdAdminAddGrpMember()

Calling Sequence:

```
int BswdAdminAddGrpMember(  
    pADMIN_INFO      pAdmin_info, - Pointer to tADMIN_INFO returned  
                                by BswdAdminLock,  
    char              *pGrp_name,  - The group name  
    char              *pGrp_member - The group name or member to  
                                add.  
)
```

RETURNS: API_ADMIN_SUCCESS - Function was successful,
User or member added.
API_ADMIN_NULL_PARAM -Required parameter is NULL
API_ADMIN_FAILURE - Some other failure.
API_ADMIN_PERMISSION_DENIED -Caller doesn't have
ADMIN permission.
API_ADMIN_IPC_READ -Socket read failure
API_ADMIN_IPC_WRITE -Socket write failure

7.1.11 BswdAdminDelGrpMember()

Calling Sequence:

```
int BswdAdminDelGrpMember(  
    pADMIN_INFO      pAdmin_info, - Pointer to tADMIN_INFO returned  
                                by BswdAdminLock,  
    char              *pGrp_name,  - The group name  
    char              *pGrp_member - The group name or member to  
                                delete.  
)
```

RETURNS: API_ADMIN_SUCCESS - Function was successful,
name or member deleted.
API_ADMIN_NULL_PARAM -Required parameter is NULL
API_ADMIN_FAILURE - Some other failure.
API_ADMIN_PERMISSION_DENIED -Caller doesn't have
ADMIN permission.
API_ADMIN_IPC_READ -Socket read failure
API_ADMIN_IPC_WRITE -Socket write failure

7.1.12 BswdAdminDeleteGrp()

Calling Sequence:

```
int BswdAdminDeleteGrp(  
    pADMIN_INFO    pAdmin_info, - Pointer to tADMIN_INFO returned  
                                by BswdAdminLock,  
    char            *pGrp_name,   - The group name  
)
```

RETURNS: API_ADMIN_SUCCESS - Function was successful,
entire group deleted.
API_ADMIN_NULL_PARAM -Required parameter is NULL
API_ADMIN_FAILURE - Some other failure.
API_ADMIN_PERMISSION_DENIED -Caller doesn't have
ADMIN permission.
API_ADMIN_IPC_READ -Socket read failure
API_ADMIN_IPC_WRITE -Socket write failure

7.1.13 BswdAdminAddGrp()

Calling Sequence:

```
int BswdAdminAddGrp(  
    pADMIN_INFO    pAdmin_info, - Pointer to tADMIN_INFO returned  
                                by BswdAdminLock,  
    char            *pGrp_name,   - The group name  
)
```

RETURNS: API_ADMIN_SUCCESS - Function was successful,
Group added.
API_ADMIN_NULL_PARAM -Required parameter is NULL
API_ADMIN_FAILURE - Some other failure.
API_ADMIN_PERMISSION_DENIED -Caller doesn't have
ADMIN permission.
API_ADMIN_IPC_READ -Socket read failure
API_ADMIN_IPC_WRITE -Socket write failure

7.1.14 BswdAdminGetDAC()

Calling Sequence:

```
int BswdAdminGetDAC(  
    pADMIN_INFO      pAdmin_info, - Pointer to tADMIN_INFO returned  
                                by BswdAdminLock,  
    char              *pRef,       - Gatekeeper or Source Reference  
    char              **hDac_rsp   - DAC BVLMF message.  
)
```

RETURNS: API_ADMIN_SUCCESS - Function was successful,
 hDac_rsp points to the response.
API_ADMIN_NULL_PARAM -Required parameter is NULL
API_ADMIN_FAILURE - Some other failure, Hdac_rsp is
 NULL.
API_ADMIN_PERMISSION_DENIED -Caller doesn't have
 ADMIN permission.
API_ADMIN_IPC_READ -Socket read failure
API_ADMIN_IPC_WRITE -Socket write failure
API_ADMIN_INVALID_REFERENCE -Invalid source or Gatekeeper
 reference.

7.1.15 BswdAdminAddDAC()

Calling Sequence:

```
int BswdAdminAddDAC(  
    pADMIN_INFO      pAdmin_info, - Pointer to tADMIN_INFO returned  
                                by BswdAdminLock,  
    char              *pRef,       - Gatekeeper or Source Reference  
    char              *pAllow      - NULL pointer or the user/group  
                                name, "ALL" or "NONE".  
    char              *pDeny      - NULL pointer or the user/group  
                                name, "ALL", or "NONE".  
)
```

RETURNS: API_ADMIN_SUCCESS - Function was successful.
 API_ADMIN_NULL_PARAM -Required parameter is NULL
 API_ADMIN_FAILURE - Some other failure.
 API_ADMIN_PERMISSION_DENIED -Caller doesn't have
 ADMIN permission.
 API_ADMIN_IPC_READ -Socket read failure
 API_ADMIN_IPC_WRITE -Socket write failure
 API_ADMIN_INVALID_REFERENCE -Invalid source or Gatekeeper
 reference.

7.1.16 BswdAdminDelDAC()

Calling Sequence:

```
int BswdAdminDelDAC(  
    pADMIN_INFO      pAdmin_info, - Pointer to tADMIN_INFO returned  
                                by BswdAdminLock,  
    char              *pRef,       - Gatekeeper or Source Reference  
    char              *pAllow      - NULL pointer or the user/group  
                                or "ALL" name.  
    char              *pDeny      - NULL pointer or the user/group  
                                or "ALL" name.  
)
```

RETURNS: API_ADMIN_SUCCESS - Function was successful.
API_ADMIN_NULL_PARAM - Required parameter is NULL
API_ADMIN_FAILURE - Some other failure.
API_ADMIN_PERMISSION_DENIED - Caller doesn't have
ADMIN permission.
API_ADMIN_IPC_READ - Socket read failure
API_ADMIN_IPC_WRITE - Socket write failure
API_ADMIN_INVALID_REFERENCE - Invalid source or Gatekeeper
reference.

7.1.17 BswdAdminGetUsers()

Calling Sequence:

```
int BswdAdminGetUsers(  
    pADMIN_INFO      pAdmin_info, - Pointer to tADMIN_INFO returned  
                                by BswdAdminLock.  
    char              role,       - Denotes what type of user  
                                information to retrieve:  
                                1 = System Admin, 2 = ISSO  
                                3 = Producer  
    char              *pSrc_ref,  - Source Reference required for  
                                producers  
    char              **hUsers_rsp - Pointer to user name list  
                                BVLmf message.  
)
```

RETURNS: API_ADMIN_SUCCESS - Function was successful,
hProducers_rsp points to response.
API_ADMIN_NULL_PARAM - Required parameter is NULL

API_ADMIN_FAILURE - Some other failure, hProducers_rsp is NULL.
API_ADMIN_PERMISSION_DENIED -Caller doesn't have ADMIN permission.
API_ADMIN_IPC_READ -Socket read failure
API_ADMIN_IPC_WRITE -Socket write failure
API_ADMIN_INVALID_REFERENCE - Invalid source reference

7.1.18 BswdAdminAddUser()

Calling Sequence:

```
int BswdAdminAddUser(  
    pADMIN_INFO    pAdmin_info, - Pointer to tADMIN_INFO returned  
                                by BswdAdminLock.  
    char           role,        - Denotes what type of user  
                                information to set.  
                                1 = System Admin, 2 = ISSO  
                                3 = Producer  
    char           *pSrc_ref,   - Source Reference for user.  
    char           *pUser      - user or group name of new  
                                user.  
                                )
```

RETURNS: API_ADMIN_SUCCESS - Function was successful.
API_ADMIN_NULL_PARAM -Required parameter is NULL
API_ADMIN_FAILURE - Some other failure.
API_ADMIN_PERMISSION_DENIED -Caller doesn't have ADMIN permission.
API_ADMIN_IPC_READ -Socket read failure
API_ADMIN_IPC_WRITE -Socket write failure
API_ADMIN_INVALID_REFERENCE -Invalid source reference

7.1.19 BswdAdminDelUser()

Calling Sequence:

```
int BswdAdminDelUser(  
    pADMIN_INFO    pAdmin_info, - Pointer to tADMIN_INFO returned  
                                by BswdAdminLock.  
    char           role,        - Denotes what type of user  
                                information to delete:  
                                1 = System Admin, 2 = ISSO
```

```

char      *pSrc_ref,      3 = Producer
char      *pUser         - Source Reference for user
                                     - user or group name of
                                     user to delete.
    )

```

RETURNS: API_ADMIN_SUCCESS - Function was successful.
 API_ADMIN_NULL_PARAM -Required parameter is NULL
 API_ADMIN_FAILURE - Some other failure.
 API_ADMIN_PERMISSION_DENIED -Caller doesn't have
 ADMIN permission.
 API_ADMIN_IPC_READ -Socket read failure
 API_ADMIN_IPC_WRITE -Socket write failure
 API_ADMIN_INVALID_REFERENCE -Invalid source reference

7.1.20 BswdAdminModElementName()

Calling Sequence:

```

int      BswdAdminModElementName(
tADMIN_INFO  *pAdmin_info,      Returned by
                                     BswdAdminLock().
char      *pSource_name,      NULL or Bswd source name.
char      *pSection_name,    Section name for this
                                     DATA_NAME.
char      *pData_name,      The DATA_NAME to
                                     modify.
char      *pData_help,      NULL or DATA_HELP
                                     modification.
char      *pDisplay_name,    NULL or DISPLAY_NAME
                                     modification.
char      mode,              Pick list or Data range Mode
                                     (A) add or (D) delete.
char      *pData_range,      NULL or DATA_RANGE to
                                     modify.
char      *pData_range_help,  NULL or help text for
                                     DATA_RANGE
char      *pData_list,      NULL or DATA_LIST to
                                     modify.
char      *pData_list_help   NULL or help text for
                                     DATA_LIST
    )
API_SUCCESS -Function successful.
API_PERM_DENIED -Current caller does not have ISSO

```

API_FAILURE	Permission. -Some other failure.
API_SUCCESS	-Function successful.
API_PERM_DENIED	-Current caller does not have ISSO Permission.
API_FAILURE	-Some other failure.

7.1.21 BswdAdminGetUserDAC()

Calling Sequence:

```
int    BswdAdminGetUserDAC(  
      tADMIN_INFO    *pAdmin_info,      Returned by  
      char            *pUser_name,      BswdAdminLock().  
      )                               The user or group name.
```

Returns:

API_SUCCESS	-Function successful.
API_PERM_DENIED	-Current caller does not have ISSO Permission.
API_ADMIN_INVALID_USER	-Supplied username not valid UNIX account
API_FAILURE	-Some other failure.

7.1.22 BswdAdminInitReg()

Calling Sequence:

```
int    BswdAdminInitReg(  
      tADMIN_INFO    *pAdmin_info,      Returned by  
      char            **hReg_info       BswdAdminLock().  
      )                               Pointer to registration info  
                                      response message (BAIR).
```

Returns:

API_ADMIN_SUCCESS	-Function successful
API_ADMIN_PERMISSION_DENIED	-Current caller does not have Admin Permission.
API_ADMIN_NULL_PARAM	-Required parameter missing
API_ADMIN_FAILURE	-Some other failure.

7.1.23 BswdAdminInitRegStatus()

Calling Sequence:

```
int    BswdAdminInitRegStatus(  
    tADMIN_INFO    *pAdmin_info,    Returned by  
    int             mode             BswdAdminLock().  
                                     REG_ABORT, or  
                                     REG_STATUS  
    )
```

Returns:

API_ADMIN_SUCCESS	-Function successful.
API_ADMIN_PERMISSION_DENIED	-Current caller does not have Admin Permission.
API_ADMIN_REG_PENDING	-Registration is still pending
API_ADMIN_NULL_PARAM	-Required parameter missing
API_ADMIN_FAILURE	-Some other failure.

7.1.24 BswdAdminRegOurGkpr()

Calling Sequence:

```
int    BswdAdminRegOurGkpr(  
    tADMIN_INFO    *pAdmin_info,    Returned by  
    char           *pGkpr_ip        BswdAdminLock().  
                                     Which gatekeeper to register  
                                     with IP address.  
    char           *pGkpr_port      Which gatekeeper to register  
                                     with TCP/IP port  
    char           *pOne_time_pwd   The one-time password  
    )
```

Returns:

API_ADMIN_SUCCESS	-Function successful.
API_ADMIN_PERMISSION_DENIED	-Current caller does not have Admin Permission.
API_ADMIN_NULL_PARAM	-Required parameter missing
API_ADMIN_FAILURE	-Some other failure.
API_ADMIN_REG_BUSY	-Key-master GKPR busy with another Registration
API_ADMIN_NOT_KM	-Attempted GKPR registration on a GKPR which is not setup as a

API_ADMIN_REG_PENDING	-Key-master Gatekeeper
API_ADMIN_REG_ABORTED	-This registration is pending
API_ADMIN_REG_GKPR_CONNECT	-This registration aborted by client
	-Could not connect to remote Key-master Gatekeeper.
API_ADMIN_REG_NO_PUBLICKEY	-Error accessing public key file
API_ADMIN_REG_NO_CERTREQ	-Error accessing certificate request File
API_ADMIN_REG_NO_CERT	-Error accessing certificate file
API_ADMIN_REG_REFUSED	-Registration failed by the Key-master.
API_ADMIN_REG_MISSING_POC	-Local gatekeeper POC information missing.

is

7.1.25 BswdAdminGenPubKey()

Calling Sequence:

```
int    BswdAdminGenPubKey(  
        tADMIN_INFO      *pAdmin_info)    Returned by  
                                             BswdAdminLock().
```

Returns:

API_ADMIN_SUCCESS	-Function successful.
API_ADMIN_PERMISSION_DENIED	-Current caller does not have Admin Permission.
API_ADMIN_NULL_PARAM	-Required parameter missing
API_ADMIN_FAILURE	-Some other failure.

7.1.26 BswdAdminGetKeypmasters()

Calling Sequence:

```
int    BswdAdminGetKeypmasters(  
        tADMIN_INFO      *pAdmin_info,  
        char              **hKeypmasters)  Returned by  
                                             BswdAdminLock().  
                                             Pointer to keymasters list  
                                             message
```

Returns:

API_ADMIN_SUCCESS	-Function successful.
API_ADMIN_PERMISSION_DENIED	-Current caller does not have Admin Permission.
API_ADMIN_NULL_PARAM	-Required parameter missing
API_ADMIN_FAILURE	-Some other failure.

7.1.27 BswdAdminGetAuditEvents()

Calling Sequence:

int	BswdAdminGetAuditEvents(tADMIN_INFO *pAdmin_info,	Returned by BswdAdminLock().
char	**hEvents)	Pointer to events list message

Returns:

API_ADMIN_SUCCESS	-Function successful.
API_ADMIN_PERMISSION_DENIED	-Current caller does not have Admin Permission.
API_ADMIN_NULL_PARAM	-Required parameter missing
API_ADMIN_FAILURE	-Some other failure.

7.1.28 BswdAdminGetConnRspMsg()

Calling Sequence:

int	BswdAdminGetConnRspMsg(tADMIN_INFO *pAdmin_info,	Returned by BswdAdminLock().
response	char **hConnRsp)	Pointer to connection message

Returns:

API_ADMIN_SUCCESS	-Function successful.
API_ADMIN_PERMISSION_DENIED	-Current caller does not have Admin Permission.
API_ADMIN_NULL_PARAM	-pAdmin_info or hConnRsp is
NULL	
API_ADMIN_FAILURE	-Some other failure.

7.1.29 BswdAdminGetUserInfo()

Calling Sequence:

int	BswdAdminGetUserInfo(tADMIN_INFO	*pAdmin_info,	Returned by BswdAdminLock().
char		*pUsername,	for this user
char		**hUsers_info_rsp)	Pointer to users info response

Returns:

API_ADMIN_SUCCESS	-Function successful.
API_ADMIN_PERMISSION_DENIED	-Current caller does not have Admin Permission.
API_ADMIN_NULL_PARAM	-Required parameter missing
API_ADMIN_FAILURE	-Some other failure.

7.1.30 BswdAdminSetUserInfo()

Calling Sequence:

int	BswdAdminSetUserInfo(tADMIN_INFO	*pAdmin_info,	Returned by BswdAdminLock().
char		*pUsername,	for this user
pPARAM_INFO		pParam_info)	Structure containing new user information

Returns:

API_ADMIN_SUCCESS	-Function successful.
API_ADMIN_PERMISSION_DENIED	-Current caller does not have Admin Permission.
API_ADMIN_NULL_PARAM	-Required parameter missing
API_ADMIN_FAILURE	

7.1.31 BswdAdminModGrp()

Calling Sequence:

int	BswdAdminModGrp(
-----	----------------------

tADMIN_INFO	*pAdmin_info,	Returned by
Char	*pGrp_name,	BswdAdminLock().
char	*pGrp_desc)	for this group
		with this description

Returns:

API_ADMIN_SUCCESS	-Function successful.
API_ADMIN_PERMISSION_DENIED	-Current caller does not have Admin Permission.
API_ADMIN_NULL_PARAM	-Required parameter missing
API_ADMIN_FAILURE	

7.1.32 BswdAdminClearStats()

Calling Sequence:

int	BswdAdminClearStats(tADMIN_INFO	*pAdmin_info,	Returned by
			BswdAdminLock().

Returns:

API_ADMIN_SUCCESS	-Function successful.
API_ADMIN_PERMISSION_DENIED	-Current caller does not have Admin Permission.
API_ADMIN_NULL_PARAM	-Required parameter missing
API_ADMIN_FAILURE	

7.1.33 BswdGetAdminMsg()

Calling Sequence:

char	*BswdGetAdminMsg(int	error_code,	error code to translate
------	----------------------	-------------	-------------------------

Returns: A heap pointer to error or message text.

7.1.34 BswdAdminSetRemGkpr()

Calling Sequence:

Broadsword-3.1-ICD-17 JUN 02
FINAL

int BswdAdminSetRemGkpr(
 tADMIN_INFO *pAdmin_info, Returned by
 * BswdAdminLock().
 char *pGkpr_ref) for this gatekeeper

Returns:

API_ADMIN_SUCCESS	-Function successful.
API_ADMIN_PERMISSION_DENIED	-Current caller does not have Admin Permission.
API_ADMIN_NULL_PARAM	-Required parameter missing
API_ADMIN_FAILURE	

Appendix VIII Catalog Manager API

8.1 Broadsword Catalog Manager Review Queue (BCMMSG)

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Catalog Manager function code, GETQ	A/N	4	Yes

8.2 Broadsword Catalog Manager Delete Queue (BCMMSG)

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Catalog Manager function code, DELETEQ	A/N	7	Yes
QUEUE_ID	Queue ID to Delete	A/N/S	Variable	Yes

8.3 Broadsword Catalog Manager Transfer Queue (BCMMSG)

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Catalog Manager function code, TRANSFERQ	A/N	9	Yes
SOURCEREF	World unique source ID	A/N/S	Variable	Yes
QUEUE_ID	Queue ID to Transfer (input) to IPL	A/N/S	Variable	Yes
USERNAME	Login account to use to transfer to IPL	A/N	28	Yes
PASSWORD	Login password to use to transfer to IPL	A/N/S	28	Yes
PRODTITLE	Product title	A/N/S	Variable	Yes

8.4 Broadsword Catalog Manager Lock Queue (BCMMSG)

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Catalog Manager function code, LOCKQ	A/N	5	Yes
QUEUE_ID	Queue ID to Lock	A/N/S	Variable	Yes

8.5 Broadsword Catalog Manager Unlock Queue (BCMMSG)

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Catalog Manager function code, UNLOCKQ	A/N	7	Yes
QUEUE_ID	Queue ID to Unlock	A/N/S	Variable	Yes

8.6 Broadsword Catalog Manager Set Metadata (BCMMSG)

Token Name	Meaning or Use	Format	Length	Required
FUNCTION	Catalog Manager function code, SETMETA	A/N	7	Yes
QUEUE_ID	Queue ID to Set Metadata	A/N/S	Variable	Yes
METADATA	New product metadata to set	A/N/S	Variable	Yes

Appendix VIII Catalog Manager API Function Definitions

9.1.1 The Broadsword Catalog Manager to Gatekeeper API defines the “C” and “C++” functions to generate the various messages defined in this ICD. This API is comprised of the following functions.

- a. BswdCatManagerReviewQ() ... This function allows the catalog manager to view what managed products reside in managed queue.
- b. BswdCatManagerDeleteQ() ... This function allows the catalog manager the ability to delete a managed product.
- c. BswdCatManagerTransferQ() ... This function allows the catalog manager the ability to transfer a managed product to the applicable source.
- d. BswdCatManagerLockQ() ... This function allows the catalog manager the ability to lock a managed product so so another manager may not work with this managed product.
- e. BswdCatManagerUnlockQ() ... This function allows the catalog manager the ability to release a lock on a managed product so another manager may manage the previously locked product.

9.1.1 BswdCatManagerReviewQ()

Calling Sequence:

```
int    BswdCatManagerReviewQ(int    socket,    /* socket id returned by the
                                           * client connect. */
                                           int    mode,    /* caller wants response in
                                           * memory or in a file. */
                                           char    **hQueue_rsp);/* Caller passes filename or
                                           * this function returns a
                                           * pointer to the response in
                                           * memory. */
```

9.1.2 BswdCatManagerDeleteQ()

Calling Sequence:

```
int    BswdCatManagerDeleteQ(char    *pQueue_id, /* Queue id to delete */
                                int    socket,    /* socket id returned by the
                                * client connect. */
                                int    mode,    /* caller wants response in
                                * memory or in a file. */
                                char    **hQueue_rsp);/* Caller passes filename or
                                * this function returns a
                                * pointer to the response in
                                * memory. */
```

9.1.3 BswdCatManagerTransferQ()

Calling Sequence:

```
int    BswdCatManagerTransferQ(char    *pQueue_id, /* Queue id to transfer */
                                char    *pSrc_ref, /* source reference */
                                char    *pUsername, /* user name for transfer */
                                char    *pPassword, /* password for above user*/
                                char    *pProdTitle, /* product title */
                                int    socket,    /* socket id returned by the
                                * client connect. */
                                int    mode,    /* caller wants response in
                                * memory or in a file. */
                                char    **hQueue_rsp);/* Caller passes filename or
                                * this function returns a
                                * pointer to the response in
                                * memory. */
```

9.1.4 BswdCatManagerLockQ()

Calling Sequence:

```
int    BswdCatManagerLockQ( char    *pQueue_id, /* Queue id to transfer */
                                int    socket,    /* socket id returned by the
                                                * client connect. */
                                int    mode,     /* caller wants response in
                                                * memory or in a file. */
                                char    **hQueue_rsp); /* Caller passes filename or
                                                * this function returns a
                                                * pointer to the response in
                                                * memory. */
```

9.1.5 BswdCatManagerUnlockQ()

Calling Sequence:

```
int    BswdCatManagerUnlockQ( char    *pQueue_id, /* Queue id to transfer */
                                int    socket,    /* socket id returned by the
                                                * client connect. */
                                int    mode,     /* caller wants response in
                                                * memory or in a file. */
                                char    **hQueue_rsp); /* Caller passes filename or
                                                * this function returns a
                                                * pointer to the response in
                                                * memory. */
```

9.1.6 BswdCatManagerSetMetadata ()

Calling Sequence:

```
int    BswdCatManagerSetMetadata( char    *pQueue_id, /* Queue id to transfer */
                                int    socket,    /* socket id returned by the
                                                * client connect. */
                                size_t metadata_len, /* length of new metadata */
                                char    *pMetadata); /* New metadata to set */
```

9.1.7 BswdCatManagerSetSiteCatElement ()

Calling Sequence:

```
int    BswdCatManagerSetMetadata( int    socket,    /* socket id returned by the
                                                * client connect. */
                                char    *pSrc_ref,    /* length of new metadata */
                                pMODELEMENT_INFO pElements); /* Element(s) to alter */
```

```
typedef struct MODELEMENT_INFO_STRUCT tMODELEMENT_INFO;
typedef struct MODELEMENT_INFO_STRUCT *pMODELEMENT_INFO;
```

```
struct MODELEMENT_INFO_STRUCT
{
    char *pSource_name;           /* source name */
    char *pSection_name;         /* pointer to element's section name */
    char *pData_name;            /* pointer to sources's element name */
    char *pData_help;            /* help text */
    char *pDisplay_name;         /* display name */
    char required;                /* field required for cataloging? Y/N */
    char mode;                    /* (A)dd or (D)elete datarange or datalist */
    char *pData_range;           /* range values */
    char *pData_range_help;      /* range help */
    char *pData_list;            /* list values */
    char *pData_list_help;       /* list help */
    pMODELEMENT_INFO pNext_element /* next element */
};
```

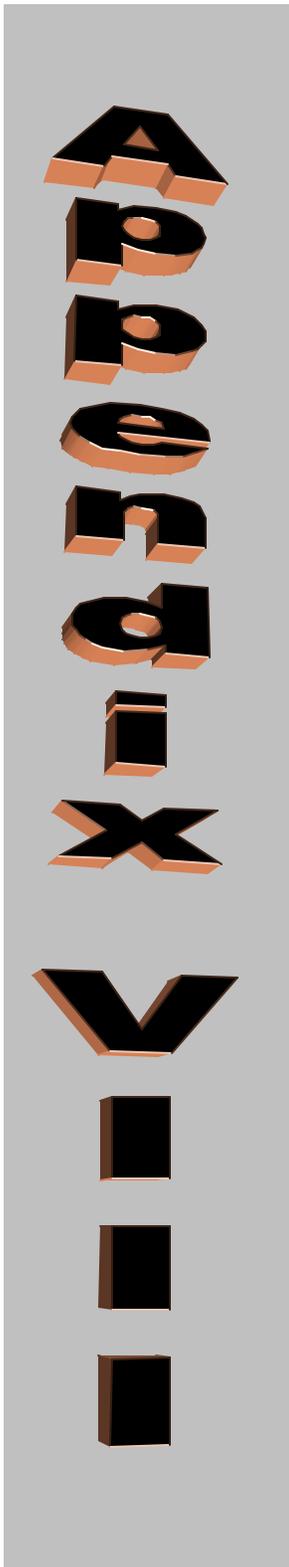
9.2 CATALOG MANAGER ERRORS

```
typedef enum
{
    CATMANAGER_API_FAILURE = -1,           /* general failure status */
    CATMANAGER_API_SUCCESS = 0,           /* good status */
    CATMANAGER_API_CONN_NULL_PARAM,       /* one of the required
                                           * connection parameters is
                                           * NULL */
    CATMANAGER_API_NO_PLUGIN_CONNECT,     /* connect to plug-in failed */
    CATMANAGER_API_UNKNOWN_FUNCTION,      /* unknown function code */
    CATMANAGER_API_NO_CATALOG,            /* no source supports
                                           * cataloging */
    CATMANAGER_API_PERMISSION_DENIED,     /* user has no catalog mgr
                                           * permission */
    CATMANAGER_API_MANSTATUS_UPDATE,      /* Manstatus update failure */
    CATMANAGER_API_Q_DELETE_FAILURE,      /* failure deleting q entry */
    CATMANAGER_API_Q_LOCKED_BY_ANOTHER,   /* product locked by
                                           * another manager */
    CATMANAGER_API_Q_NO_LOCK,             /* operation requires lock that
                                           * is not present */
    CATMANAGER_API_NO_MANAGED_PRODUCT,    /* queue_id is invalid,
                                           * product does not exist */
    CATMANAGER_API_TRAN_PENDING,          /* product in transfer pending
                                           * state */
    CATMANAGER_API_BUILD_MSG,             /* api build msg failed */
    CATMANAGER_API_PARSE_MSG,             /* api parse msg failed */
    CATMANAGER_API_NO_MEMORY,             /* memory failure */
    CATMANAGER_API_FILE_READ,             /* disc read error */
    CATMANAGER_API_FILE_WRITE,            /* disc write error */
};
```

Broadsword-3.1-ICD-17 JUN 02
FINAL

CATMANAGER_API_FILE_OPEN,	/* file open error */
CATMANAGER_API_IPC_FAILURE,	/* communications failure */
CATMANAGER_API_IPC_READ,	/* comm read error */
CATMANAGER_API_IPC_WRITE,	/* comm write error */
CATMANAGER_API_SOCKET,	/* invalid socket id */
CATMANAGER_API_MODE,	/* invalid mode */
CATMANAGER_API_INVALID_PARAM,	/* invalid parameter * supplied*/
CATMANAGER_API_NULL_PARAM,	/* required parameter * NULL*/

Miscellaneous



10.1 Defaults, Parameter Values and Definitions

1. CLIENT_HELP: (Y)es or (N)o .
2. STATUS_PORT: TCP/IP port number where status messages are to be sent.
3. STATUS: Numeric value to indicate Success (0) or Failure (-1).
4. CLIENT_ROLE: Indicates the gatekeeper determination of client privilege; 0 = General user, 1 = Producer, 2 = System Administrator. This value is returned in the BGRM.
5. TYPE: Denotes gatekeeper type A, assigned; R remote.
6. FIND: (Y)es (N)o flag to indicate Keyword search capability.
7. BQS_STMT: A valid Boolean Query Syntax statement. This will contain the users query, find or geo search statement. The current implementation will accept any number of simple attribute names, queries or text attribute names but only one each of geo attribute names. Example: "TGTID = "0000AA0000" AND TGTGEO INTERSECT POINT (012345N0134567W)". All geo query parts must be after the simple, text, or query part of the BQS_STMT prefixed with AND.
8. TARGET_GEO_PNT: A valid geo attribute name to indicate capability to perform a geographic search on a Target point. Example: TGTGEO
9. TARGET_GEO_CIR: A valid geo attribute name to indicate capability to perform a geographic search on a Target by circle. Example: TGTGEO
10. TARGET_GEO_POLY: A valid geo attribute name to indicate capability to perform a geographic search on a Target by polygon. Example: TGTGEO
11. TARGET_GEO_POLY_VERT: Maximum number of vertices supported by TARGET_GEO_POLY. Default minimum is 3.
12. IMAGE_GEO_PNT: A valid geo attribute name to indicate capability to perform a geographic search on a Image point. Example: IGEOLO
13. IMAGE_GEO_CIR: A valid geo attribute name to indicate capability to perform a geographic search on an Image by circle. Example: IGEOLO
14. IMAGE_GEO_POLY: A valid geo attribute name to indicate capability to perform a geographic search on an Image by polygon. Example: IGEOLO
15. IMAGE_GEO_POLY_VERT: Maximum number of vertices supported by IMAGE_GEO_POLY. Default minimum is 3.
16. DATA_TYPE: Data attribute type; (C) character, (N) numeric, (S) signed numeric. A character type can contain alpha, numeric and special characters.
17. DATA_PRECISION: A numeric number of decimal positions for Signed values.
18. DATA_RANGE: A range expression depicting the data type: This token may repeat. The first token will be a LOW limit value, the next the HIGH. Example NIIRS 0.0 NIIRS 10.0.
19. DATA_LIST: A list of valid values for a particular element. This token may repeat. Example: CLASS T, CLASS S, CLASS C, CLASS R, CLASS U.
20. QUERY_TYPE: A numeric flag to indicate type of query; 0 for Sequential, 1 for simultaneous. Default is simultaneous.

21. THUMBNAILS: (Y)es (N)o flag to return thumbnails in the query response. Default is (N)o.
22. QRY_STATUS: A numeric flag to indicate Success (0) or Failure (1) of query.
23. DUP: (Y)es (N)o flag to indicate that this HIT is a duplicate of another HIT.
24. CLASS: Product classification; T,S,C,R,U.
25. CLEVEL: NITF compliance level: 01,02,03,04,05,06 and 99
26. DWNG: Downgrade date: Date or 999999 or 999998
27. PRODOFFLINE: (Y)es (N)o flag to indicate if product is currently Offline.
28. ESD: (Y)es (N)o flag to indicate if Exploitation Support Data is available.
29. RPC: (Y)es (N)o flag to indicate if Rapid positional capability is available.
30. SRP: (Y)es (N)o flag to indicate if product is a Standard Radiometric Product.
31. LAN_CLASS: Classification of the LAN which will transfer image.
32. RRDS: Numeric value 0 – 8 to indicate which Reduces Resolution Data Set to return. 8 indicates return all.
33. AUTO_RED: Image request flag; 1 Send suitable RRDS, 2 Send Image Cropped, 3 Abort request.
34. METADATA: (Y)es (N)o flag to request a flat file in source specific format of all relative product metadata.
35. NOTIFICATION_MODE: Numeric flag to indicate mode of request; 0 = IPC notification, 1 = Email Notification, Blank = No notification.

8.2 Miscellaneous Examples

A. Product delivery email notification text:

From Your Local Gatekeeper:

Product Request From: Local Gatekeeper Description
SOURCE: IPL 1.0 at Somewhere Via Local Gatekeeper Description

REQUESTID:
08135305Zoct99.000000283213001002003004000010000015060
ACCESSID: IPA_123456789_123456
FORMAT: NITF02.00
DEST IP ADDRESS: 123.456.789.1
DESTLOGIN: username
DESTPATH: /home/username
FILENAME: usersfilename
STATUS: Transfer successful.