37-2.0-ICD-08 99-B0
25 August 1999

INTERFACE CONTROL DOCUMENT FOR PROJECT BROADSWORD VERSION 2.0

Prepared for:

497th INTELLIGENCE GROUP
INTELLIGENCE SYSTEMS DIRECTORATE (497IG/IND)
BOLLING AIR FORCE BASE
WASHINGTON, DC 20332-5000

Prepared by:

Air Force Research Laboratory, Rome Research Site
AFRL/IFED
32 Hangar Road
Rome, NY 13441-4114

August 1999

TABLE OF CONTENTS

Foreword **6**
1. Executive Summary **7**
1.1 IDENTIFICATION 7
1.2 SYSTEM OVERVIEW 7
1.3 DOCUMENT ORGANIZATION 7
1.4 DEFINITIONS 8
1.5 LIMITATIONS AND RESTRICTIONS 8
2.1 GOVERNMENT DOCUMENTS 9
2.2 NON-GOVERNMENT DOCUMENTS 9
3. SEGMENT INTERFACES **10**
4. Broadsword Client Gatekeeper Application Program Interface Specification **11**
4.1 CLIENT TO GATEKEEPER APPLICATION PROGRAM INTERFACE 11
4.1.1 INTERFACE IDENTIFICATION 11
4.1.2 INTERFACE DESCRIPTION 11
4.1.3 FUNCTIONAL INTERFACE SPECIFICATION 11
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
5. Broadsword Gatekeeper Plug-in Application Program Interface Specification **13**
5.1 GATEKEEPER TO PLUG-IN APPLICATION PROGRAM INTERFACE 13
5.1.1 INTERFACE IDENTIFICATION 13
5.1.2 INTERFACE DESCRIPTION 13
5.1.3 FUNCTIONAL INTERFACE SPECIFICATION 13
5.1.3.1 GATEKEEPER TO PLUG-IN MESSAGES 14
5.1.3.2 PLUG-IN TO GATEKEEPER MESSAGES. 14
5.1.4 PHYSICAL INTERFACE SPECIFICATION. 14
6. Broadsword ISSO to Gatekeeper Application Program Interface Specification **15**
6.1 ISSO TO GATEKEEPER APPLICATION PROGRAM INTERFACE 15
6.1.1 INTERFACE IDENTIFICATION 15
6.1.2 INTERFACE DESCRIPTION 15
6.1.3 FUNCTIONAL INTERFACE SPECIFICATION 15
6.1.3.2 GATEKEEPER TO ISSO MESSAGES 16
6.1.4 PHYSICAL INTERFACE SPECIFICATION. 16
7. Broadsword SYS ADMIN to Gatekeeper Application Program Interface Specification **16**
7.1.3 FUNCTIONAL INTERFACE SPECIFICATION 17
8. Broadsword Variable Length Message Format (BVLMF). **21**
8.1 BROADSWORD VARIABLE LENGTH MESSAGE FORMAT OVERVIEW 21
8.2 BVLMF BASIC STRUCTURE. 21
8.3 BVLMF MESSAGE CONSTRUCTION AND PARSING 21
Appendix I **22**
1. BROADSWORD GATEKEEPER APPLICATION PROGRAM INTERFACE 22
1.1 BROADSWORD GATEKEEPER TO CLIENT CONNECT MESSAGE (BGCCM) 22
Figure 1.1-1 22
1.2 BROADSWORD GATEKEEPER CONNECT RESPONSE MESSAGE (BGCRM) 23
Figure 1.2.1 29
1.3 BROADSWORD GATEKEEPER TO CLIENT QUERY MESSAGE (BGCQM) 30

FIGURE 1.3-1....31
1.4 BROADSWORD GATEKEEPER QUERY RESPONSE MESSAGE (BGQRM)....32
FIGURE 1.4-1....40
1.5 BROADSWORD GATEKEEPER CLIENT PRODUCT REQUEST MESSAGE (BGCPRM)....41
FIGURE 1.5-1....44
1.6 BROADSWORD GATEKEEPER PRODUCT REQUEST ACKNOWLEDGMENT (BGPRA)....45
FIGURE 1.6-1....45
1.7 BROADSWORD GATEKEEPER PRODUCT REQUEST IPC STATUS RESPONSE. BGIPCR....46
THE BGIPCR MESSAGE IS USED TO COMMUNICATE PRODUCT REQUEST STATUS UPDATES VIA SOCKET INTER-PROCESS COMMUNICATION. THIS MESSAGE IS SENT TO THE IP ADDRESS AND PORT NUMBER IDENTIFIED IN THE BGCPRM. FIGURE 1.7.1 OUTLINES THE BGIPCR MESSAGE. 46
1.8 BROADSWORD GATEKEEPER PRODUCT STATUS REQUEST MESSAGE (BGCRFS)....47
FIGURE 1.8-1....47
1.9 BROADSWORD GATEKEEPER CLIENT CATALOG NEW PRODUCT (BGCCNP)....48
A.... 48
FIGURE 1.9-1....48
1.10 BROADSWORD GATEKEEPER CATALOG NEW PRODUCT ACKNOWLEDGMENT (BGCNPA)....49
1.11 BROADSWORD GATEKEEPER CLIENT REQUEST FOR CATALOG STATUS (BGCRFCS)....50
FIGURE 1.11-1....50
1.12 BROADSWORD GATEKEEPER CLIENT REQUEST FOR TOP TEN REQUESTS (BGCTTR)....51
FIGURE 1.12-1....51
1.13 BROADSWORD GATEKEEPER TOP TEN REQUESTS RESPONSE MESSAGE (BGCTTRM)....52
FIGURE 1.12-1....52
Appendix II.... 53
2.1. API FUNCTION DEFINITIONS....53
2.1.1 BSWDCLIENTCONNECT()....53
2.1.2. BSWDCLIENTQUERY()....55
2.1.3 BSWDCLIENTPRODUCTREQ()....57
2.1.4 BSWDCLIENTRFS() 60
2.1.5 BSWDCLIENTCNP()....61
2.1.6 BSWDCLIENTRFCS()....62
2.1.7 BSWDCLIENTRTT()....63
2.1.8 CLIENTLOGOUT() 64
2.2 API ERRORS....65
2.3 SUPPORT FUNCTIONS 66
Appendix III 68
3. BROADSWORD GATEKEEPER TO PLUG-IN APPLICATION PROGRAM INTERFACE....68
3.1 BROADSWORD GATEKEEPER TO CLIENT QUERY MESSAGE (BGCQM)....68
3.2 BROADSWORD GATEKEEPER TO CLIENT QUERY MESSAGE (BGCPRM)....69
3.3 SUPPORT FUNCTIONS 70
Appendix IV.... 73
4. BROADSWORD ISSO TO GATEKEEPER APPLICATION PROGRAM INTERFACE 73
4.1 BROADSWORD GATEKEEPER AUDIT DETAIL RECORD REQUEST (BAMSG)....73
4.2 BROADSWORD GATEKEEPER AUDIT EXTRACT DATA REQUEST (BAMSG) 73
4.3 BROADSWORD GATEKEEPER AUDIT UPDATE DATA REQUEST (BAMSG) 74
4.4 BROADSWORD GATEKEEPER AUDIT DETAIL RECORD REQUEST FROM ARCHIVE (BAMSG) 74
4.5 BROADSWORD GATEKEEPER AUDIT DETAIL RESPONSE (BGADR)....74
4.6 BROADSWORD GATEKEEPER STATUS RESPONSE (BARSP)....75
4.7 BROADSWORD GATEKEEPER STATUS RESPONSE (BARSP)....75

4.8 BROADSWORD GATEKEEPER STATUS RESPONSE (BARSP) 75
4.8 BROADSWORD GATEKEEPER STATUS RESPONSE (BGARSP) 75
Appendix VI 82
6. BROADSWORD ADMIN TO GATEKEEPER APPLICATION PROGRAM INTERFACE 82
6.29 BROADSWORD GATEKEEPER STATUS RESPONSE (BARSP) 87
6.30 BROADSWORD ADMIN GATEKEEPER PLUG-IN NAME RESPONSE (BAGPN) 87
GATEKEEPER 89
TYPE 89
GKPRREF 89
GKPRDESC 89
ACCESS_PERMS 89
DENY 89
NAME_REC 89
NAME 89
TYPE 89
ALLOW 89
NAME_REC 89
NAME 89
TYPE 89
SOURCE 89
ACCESS_PERMS 90
DENY 90
NAME_REC 90
NAME 90
TYPE 90
ALLOW 90
NAME_REC 90
NAME 90
TYPE 90
SOURCENAME 90
GROUPNAME 91
NAME_REC 91
NAME 91
TYPE 91
DENY 92
NAME_REC 92
NAME 92
TYPE 92
ALLOW 92
NAME_REC 92
NAME 92
TYPE 92
NAME_REC 92
NAME 92
TYPE 92
A 93
GKPRREF 93
A 94
EVENT_NAME 94

Appendix VII 95
8.1 DEFAULTS, PARAMETER VALUES AND DEFINITIONS 113

FOREWORD

The following DRRs have been incorporated into this document:

DRR#	Short Description
37-1.0-ICD-001	Add Document Reference
37-1.0-ICD-002	Increase Field Length
37-1.0-ICD-003	Increase Field Length

37-2.0-ICD-08 99-B0
25 August 1999

1. EXECUTIVE SUMMARY

1.1 Identification

This Interface Control Document (ICD) establishes the requirements for the interface between the Broadsword Gatekeeper version 1.0 and Client Applications. This document was developed by AFRL/IFED.

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, IESS and IDEX 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.

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

RFC N01-0044 N0101-B	Geospatial and Imagery Access Services Specification
5 February 1998	

STANDARDS

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

DRAWINGS

None.

OTHER PUBLICATIONS

None.

2.2 Non-Government documents

None.

3. SEGMENT INTERFACES

There are four interfaces controlled by this document. The first interface is the Client to Gatekeeper and Gatekeeper to Client interface. The second interface is the Gatekeeper to Plug-in and Plug-in to Gatekeeper interface. The third interface is the ISSO to Gatekeeper audit interface. The fourth and final interface is the SYS ADMIN to Gatekeeper interface. This is the only document related to these interfaces.

4. BROADSWORD CLIENT GATEKEEPER APPLICATION PROGRAM INTERFACE 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. 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.

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 (BGPRA) 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.

4.1.4 Physical Interface Specification.

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

5. BROADSWORD GATEKEEPER PLUG-IN APPLICATION PROGRAM INTERFACE 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.

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.

5.1.4 Physical Interface Specification.

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

6. BROADSWORD ISSO TO GATEKEEPER APPLICATION PROGRAM INTERFACE 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.

7. BROADSWORD SYS ADMIN TO GATEKEEPER APPLICATION PROGRAM INTERFACE 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 configured.
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 configured 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 discretionary access control permissions for a particular gatekeeper or backside source.
r. Remove for a particular gatekeeper or backside source.
s. Set the discretionary access control permissions to a particular gatekeeper or backside source.
t. Retrieve the user discretionary access control permissions.
u. Retrieve the valid users, producers, admin users or isso users for the local gatekeeper or a particular backside source.
v. Add users as a valid producer to a particular backside source or add users as valid sys admins or issos to the local Gatekeeper.
w. Delete users as valid producers to a particular backside source or from valid sys admins or issos to the local Gatekeeper.
x. Modify a configured DATA_NAME's parameters
y. Initialize a gatekeeper registration.
z. Retrieve gatekeeper registration status.
aa. Register a gatekeeper with the Keymaster.
bb. Generate a public/private key pair.
cc. Retrieve the list of Keymasters.
dd. Retrieve the list of Auditable Events.

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 configured 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

CREATSRC will create a new backside source.

h. **Broadsword Gatekeeper Admin Delete Backside Source.** This message with function code DELETESRC will delete a configured 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. **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.

t. **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.

u. **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.

v. **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.

w. **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.

x. **Broadsword Gatekeeper Admin Get User Discretionary Access Control.** This message with function code GETUDAC obtains the access control for all Gatekeepers and backside Sources for a user or group.
y. **Broadsword Gatekeeper Admin Initialize Gatekeeper Registration.** This message with function code INITREG initializes the Gatekeeper registration process with a Key-master.
z. **Broadsword Gatekeeper Admin Initialize Registration Status.** This function allows the administrator to either wait for a registration status response from the Gatekeeper or send a message with function code ABORTREG to abort the registration in progress.
aa. **Broadsword Gatekeeper Admin Register Our Gatekeeper.** This message with function code REGOURGKPR processes the registration with the Key-master.
bb. **Broadsword Gatekeeper Admin Generate Public/Private Keys.** This message with function code GENPUBKEY causes the public/private key pair to be generated on a Gatekeeper.
cc. **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.
dd. **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.

7.1.4 Physical Interface Specification.

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

8. BROADSWORD VARIABLE LENGTH MESSAGE FORMAT (BVLMF).

8.1 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.

8.2 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.

8.3 BVLMF Message Construction and Parsing

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

APPENDIX I

1. Broadsword Gatekeeper Application Program Interface

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 Figure 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

Figure 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. Figure 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 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
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

Token Name	Meaning or Use	Format	Max Length	Required
SOURCE_DESC	Description of the source	A/N	Variable	Yes
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_VERT	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_VERT	Numeric value to indicate maximum number of vertices supported for IMAGE_GEO_POLY.	N	2	No
PRODUCER	User has producer permissions (Y/N). Default is N	A	1	No

Token Name	Meaning or Use	Format	Max Length	Required
CHIP_MODE	Method of image chipping. **G**eo, **F**af or **N**one. 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-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 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 queryable field per section	A	N/A	Yes
DATA_NAME	Data field name	A/N	Variable	Yes

Token Name	Meaning or Use	Format	Max Length	Required
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
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	A	N/A	Yes

Token Name	Meaning or Use	Format	Max Length	Required
	required when a query is performed on this source.			
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

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
DATA_HELP	Data field element description or help text	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

Token Name	Meaning or Use	Format	Max Length	Required
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
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

Figure 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. Figure 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	Binary query syntax statement for query and or geo search.	A/N/S	Variable	No
FIND_STMT	Binary 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 token may repeat)	A/N/S	Variable	No
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	A/N/S	Variable	No

Token Name	Meaning or Use	Format	Max Length	Required
	TOKEN NAMES for a particular source. The name or names are returned in the BGCRM.			

FIGURE 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. Figure 1.4-1 outlines the message format for the BGQRM. This message may contain many BVLMF sub-messages.

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 duplicate(Y/N)	A	1	Yes
SOURCE_REF	Source reference generating this hit.	A/N/S	47	Yes
PRD	This token identifies the PRODUCT sub-message. This message will contain at least the ACCESSID.	A	N/A	Yes
ACCESSID	Ipl/5D ACCESSID, IESS/NDS IMAGEID, MIDB FAC_KEY	A/N/S	64	Yes
AUTHORITY	Classification authority	A/N/S	20	No

Token Name	Meaning or Use	Format	Max Length	Required
CLASS	Product classification	A	1	Yes
CLEVEL	NITF compliance level	N	2	No
CODEWORDS	Codewords	A/N/S	40	No
CONTROL	Control and handling	A/N/S	40	No
DWNG	Downgrade date	N	8	No
DWNGEVENT	Downgrade event. This field is mandatory if DWNG = 999998	A/N/S	40	Cond.
FMCONTROL	Foreign material control	A/N/S	32	No
MAPID	Map identifier	A/N/S	40	No
NUMFILES	Number of files in product	N	2	No
ONAME	Originators name	A/N	27	No
OSTAID	Originators station identifier	A/N	10	No
PRODCODE	Product code	A/N	2	No
PRODCRTIME	Product create time (ZULU)	A/N/S	14	No
PRODFMT	Product file format	A/N/S	Variable	No
PRODFSIZE	Total product file size	N	12	No
PRODIDNO	Product identification number	A/N/S	20	No
PRODOFFLINE	Product is off-line flag (Y/N) default is N	A	1	No
PRODSNME	Product short name	A/N/S	10	No
PRODTITLE	Product title	A/N/S	80	No
PRODUCERCD	Producer code	A	2	No
PRODUCERSE	Producer source	A/N/S	6	No
RELEASE	Countries which product is authorized for release.	A/N	80	No
SUBDET	Subjective detail	A/N	1	No
DATECREATED	Date product entered archive (ZULU)	A/N/S	14	No
DATEMODIFIED	Date product was modified (ZULU)	A/N/S	14	No
UTAPEID	Archive tape identifier	A/N/S	40	No
NUM_THUMBNAILS	Total number of thumbnails available for this product	N	Variable	No
ATEXT	Associated text	A/N/S	25245	No
DATASET	This code identifies the type of dataset	C	40	No

Token Name	Meaning or Use	Format	Max Length	Required
SECURITYTEXT	Provides additional information about classification	A	1	No
THM	This token identifies the THUMBNAIL sub-message.	A	N/A	No
FILE_NUMBER_TH	Image file origin of thumbnail	N	2	Yes
IMAGENBR_TH	Image number origin of thumbnail	N	3	Yes
FORMAT	Image format of thumbnail	A/N/S	Variable	Yes
THUMBNAIL	Thumbnail image data	A/N/S	Variable	Yes
SCT	This token identifies the SECTION details sub-message	A	N/A	No
FILE_NUMBER_S	File number origin of Section details	N	2	Yes
PPNUM	Page part number	A/N	5	No
TPP	Total pages/parts	N	3	No
SECTITLE	Section title	A/N	40	No
ARP	This token identifies the ASSOCIATED REPORTS sub-message	A	N/A	No
FILE_NUMBER_A	File number origin of Associated Report	N	2	Yes
ASSRPT	Associated report	A/N/S	20	No
ID	Unique source or report identifier	A/N	8	No
CUTOFFDATE	Date of the information of the referenced source	N	8	No
TYPE	Type or intelligence report or Document referenced	A/N	9	No
KEY	This token identifies the KEYWORD sub-message	A	N/A	No
FILE_NUMBER_K	File number origin of Keyword	N	2	Yes
KEYWORD	Keyword	A/N/S	255	No
RQO	This token identifies the REQUESTING ORGANIZATION sub-message	A	N/A	No
FILE_NUMBER_R	File number origin of Requesting Organization	N	2	Yes
REQORG	Requesting Organization	A/N/S	64	No

Token Name	Meaning or Use	Format	Max Length	Required
PER	This token identifies the PERSON sub-message	A	N/A	No
ASSOCTRY	Associated country	A	2	No
DOB	Date of birth (CCYYMMDD)	N	8	No
FIRSTNME	Person first name	A/N	28	No
LASTNME	Person last name	A/N	28	No
MIDNME	Person middle name	A/N	28	No
FILE_NUMBER_P	File number origin of Person	N	2	No
IMAGENBR_P	Image number origin of Person	N	2	No
GRADE	Military grade of person	A/N	2	No
PROFESSION	Job, type of employment	A/N	54	No
UTMCOORDS	UTM coordinate of person	A/N	16	No
GEOCOORDS	Geo coordinate of person	A/N	15	No
EQP	This token identifies the EQUIPMENT sub-message	A	N/A	No
CTRYDSN	Country code where product designed	A/N	2	No
CTRYPROD	Country code where product produced	A/N	2	No
EQPCODE	Equipment code	A/N/S	7	No
EQPMAN	Equipment manufacturer	A/N/S	64	No
EQPNOMEN	Equipment nomenclature	A/N/S	45	No
OBTYPE	Order of battle type	A/N/S	1	No
ORDBAT	Order of battle	A/N/S	3	No
OBJVIEW	Objective view	A/N/S	6	No
FILE_NUMBER_E	File number origin of equipment	N	2	Yes
IMAGENBR_E	Image number origin of equipment	N	3	No
ALC	Airfield load class	A	1	No
APT	Airfield pavement type	A	1	No
CAP	The actual or ability to do something	N	Variable	No
ELNOT	Electronic intelligence ELINT notation.	A/N	5	No
ELSITE	Site number of a specific electronic site or equipment	A/N	5	No
LENGTH	Measurement of the extent of an	N	Variable	No

Token Name	Meaning or Use	Format	Max Length	Required
	object.			
WAC	World area code	A/N	4	No
PRIMOPERAREA	Primary operating area	A	2	No
QOH	Quantity on hand	N	Variable	No
ALG	Country this entity owes its allegiance.	A	2	No
AZIMTH	This entity's orientation	N	Variable	No
EVT	This token identifies the EVENT sub-message	A	N/A	No
EVENTNAME	Event name	A/N/S	38	No
EVENTTYPE	Event type	A/N/S	8	No
FILE_NUMBER_EV	File number origin of event	N	2	Yes
IMAGENBR_EV	Image number origin of event	N	3	No
BEGINDATE	Date event started	N	14	No
ENDDATE	Date event ended	N	14	No
EVTID	Event identifier	A/N	11	No
LOC	Location of event	A	2	No
UTMCOORDS	UTM coordinate of event	A/N	16	No
GEOCOORDS	Geographic coordinate of event	A/N	15	No
TGT	This token identifies the TARGET sub-message	A	N/A	No
CAT	Target category	A/N/S	5	No
CTRYCD	Target country code	A/N	2	No
DATUM	Datum of map	A/N	3	No
PERCOVER	Percent of coverage	N	3	No
TGTGEO	Target geographic coordinate	A/N/S	15	No
TGTID	Target identifier (BE_NUMBER)	A/N/S	10	No
SUFFIX	Target identifier suffix	N	5	No
TGTNAME	Target or facility name	A/N/S	38	No
TGTUTM	Target UTM coordinates	A/N/S	15	No
FILE_NUMBER_T	File number origin of target	N	2	Yes
IMAGENBR_T	Image number origin of target	N	3	No
ADA	Air defense area	A	5	No
ALG	Country or political entity facility owes allegiance	A	2	No
AZIMUTH	Orientation of facility	N	Variable	No

Token Name	Meaning or Use	Format	Max Length	Required
DEPTH	Depth of facility	N	Variable	No
ELEV	Elevation of facility	N	Variable	No
REGION	General area of the world where facility is located.	A	2	No
LENGTH	Length of facility	N	Variable	No
OPERSTATUS	Operational status of facility	A	3	No
PERALG	Country allegiance of personnel	A	2	No
QUANPER	Number of personnel at facility	N	Variable	No
SITENUMBER	Facility site number	N	Variable	No
TGTDATAINV	Target data inventory program	A	3	No
TGTRAD	Radius of the target	N	Variable	No
WAC	World area code	A	4	No
WIDTH	Width of facility	N	Variable	No
IMG	This token identifies the IMAGE sub-message	A	N/A	No
ABPP	Actual bits per pixel	N	2	No
ANGLETONORTH	Angle to north	N	3	No
CAMSPECS	Camera specifications	A/N	32	No
CLOUDCVR	Percent of cloud cover	N	3	No
COMGEN	Compression generation	N	2	No
COMRAT	Compression rate	A/N/S	4	No
ESD	Exploitation support data available	A	1	No
GENERATION	Image generation	N	1	No
IC	Image compression type	A/N	2	No
ICAT	Image category	A/N/S	8	No
ICORDS	Image Coordinate type	A	10	No
IGEOLO	Image four corner geographic coordinates	A/N/S	60	No
IMAGEID	200 EA Image Identifier	A/N/S	60	No
IREP	Image representation	A/N	8	No
MEANGSD	Mean GSD	N	7	No
MSNNUM	Mission number	A/N	7	No
NBANDS	Number of image bands	N	4	No
NCOLS	Number of pixel columns	N	8	No
NROWS	Number of pixel rows	N	8	No
NIIRS	National image interpretability	S	3	No

Token Name	Meaning or Use	Format	Max Length	Required
	rating scale			
OBLANGLE	Obliquity angle	S	5	No
OTHERCOND	Other conditions	A/N	2	No
PROJID	Project identifier	A	2	No
RPC	Rapid positional capability	A	1	No
SENSMODE	Sensor mode	A/N/S	12	No
SENSNAME	Sensor name	A/N/S	18	No
SOURCE	Source of image	A/N/S	255	No
SRP	Standard radiometric product	A	1	No
SUBQUAL	Subjective quality	A	1	No
SUNAZ	Sun azimuth	S	5	No
SUNEL	Sun elevation	S	5	No
STEREOID	Stereo identifier	A/N/S	120	No
TIMECOLL	Time of image collection (ZULU)	A/N/S	14	No
IDATUM	Datum of image	A/N/S	10	No
LIC	License	A/N/S	10	No
PREPROC	Preprocessing required	A/N/S	20	No
SATID	Satellite identification	A/N/S	10	No
SATTRACK	Satellite track location	A/N/S	8	No
START_FAF_H	Start faf horizontal	N	5	No
START_FAF_V	Start faf vertical	N	2	No
END_FAF_H	End faf horizontal	N	5	No
END_FAF_V	End faf vertical	N	2	No
FILE_NUMBER_I	File number origin of image	N	2	Yes
IMAGENBR_I	Image number	N	3	Yes
PROJSYS	Projection system	A/N	2	No

Token Name	Meaning or Use	Format	Max Length	Required
NWL	This token identifies the Network Link Sub-message.	A	3	No
COMMID	Uniquely identifies each communications network	A/N	13	No
LEVEL	Level of this link in construction	A/N	5	No
ALG	Country or political entity to which network owes allegiance	A	2	No
NAME	Recognized network or link name	A/N	54	No
TYPE	Function of the network link	A	3	No
NWN	This token identifies the Network Node Sub-message	A	3	No
ROLE	Role node plays in network	A/N	3	No
UTMCOORDS	UTM coordinate of node	A/N	16	No
GEOCOORDS	Geographic coordinate of node	A/N	15	No
OBS	This token identifies the Observation Sub-message	A	3	No
CLUSTERID	Unique id for a cluster of intercepts or reports	A/N	15	No
EXTID	External Id assigned to the observation or track	A/N	30	No
UTMCOORDS	UTM coordinate of observation	A/N	16	No
GEOCOORDS	Geographic coordinate of observation	A/N	15	No
CC	Country code of observation	A	2	No
SIG	This token identifies the Signal sub-message	A	3	No
ID	Unique signal identifier	A/N	6	No
TRK	This token identifies the TRACK sub-message	A	3	No
ALG	Country or political entity to which track owes its allegiance	A	2	No
SPEED	Estimated speed or track	N	Variable	No
NAME	Name of entity being tracked	A/N	54	No
UTMCOORDS	UTM coordinate of track	A/N	16	No
GEOCOORDS	Geographic coordinate of track	A/N	15	No
CC	Country code of track	A	2	No

Token Name	Meaning or Use	Format	Max Length	Required
UNT	This token identifies the UNIT sub-message	A	3	No
ECHELON	Organization level of unit	A/N	4	No
FLEETID	Naval fleet unit is assigned	A/N	1	No
MILAREA	Military district, region or zone	A/N	5	No
OPERSTATUS	Operational status of unit	A/N	3	No
PERSTYPE	Type of personnel in unit	A/N	3	No
PRIMISSION	Primary mission of unit	A/N	4	No
UNITID	Unique identifier of unit	A/N	10	No
QOH	Quantity of personnel on hand	N	Variable	No
NAME	Name of the unit	A/N	54	No
GEOCOORDS	Geographic coordinate of unit	A/N	15	No
UTMCOORDS	UTM coordinate of unit	A/N	16	No
CC	Country code of unit	A	2	No
ALG	Country or political entity to which unit owes allegiance	A	2	No
OBTYPE	Order of battle type	A	1	No
OTHERS	This token is used to denote other tables, views or data groups. These names must be unique as well as the token names. These new views will be maintained by their creator.			
ANY_TOKEN_NAME				

FIGURE 1.4-1

1.5 Broadsword 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. Figure 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 toke is part of the parent **BGCPRM** message.	A/N/S	64	Yes
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	N/S	Variable	No

Token Name	Meaning or Use	Format	Max Length	Required
	images.			
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	Binary query syntax statement for report generation.	A/N/S	Variable	No
NOTIFICATION_MODE *	Identifies the request as either: Synchronous, asynchronous, email notification of IPC notification. This release only supports IPC notification.	A	1	Yes
EMAIL_ADDR *	User full email address if notification method is email.	A/N/S	Variable	No
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

Token Name	Meaning or Use	Format	Max Length	Required
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
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

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

FIGURE 1.5-1

1.6 Broadsword Gatekeeper Product Request Acknowledgment (BGPRA)

The BGPRA 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 BGPRA will be composed of multiple BVLMF sub-messages. The message format is outlined in figure 1.6-1.

Parent Message Name **BGPRA**

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	CISL 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

FIGURE 1.6-1

1.7 Broadsword Gatekeeper Product Request IPC Status response. BGIPCR

The **BGIPCR** message is used to communicate product request status updates via socket inter-process communication. This message is sent to the IP address and Port number identified in the **BGCPRM**. Figure 1.7.1 outlines the **BGIPCR** message.

Token Name	Meaning or Use	Format	Max Length	Required
HOSTNAME	Requestors host name	A/N/S	30	Yes
USERID	Requestors user id.	A/N/S	28	Yes
REQUESTDTM	Request date/time stamp	A/N/S	14	Yes
DESTHOSTNAME	Request destination host name	A/N/S	30	No
DEST_NUMBER	Destination number	N	2	No
ACCESSID	Requested product's Accessid	A/N/S	64	Yes
DEST_PATH	Destination path name	A/N/S	255	No
FILENAME	Requestor's filename	A/N/S	80	No
REPORT_TIME	Status report time	A/N/S	14	Yes
TRANSFER_SIZE	File transfer byte size	N	12	Yes
SOURCE_IP	Product's source IP address	A/N/S	15	Yes
NUM_FILES	Number of files transferred	N	3	Yes
CLIENT_PID	Process id of requestor	N	10	Yes
REQUEST_NUMBER	One up number of client's request	N	5	Yes
STATUS_TEXT	Current product request status	A/N	Variable	Yes

Figure 10-7.1

1.8 Broadsword Gatekeeper Product Status Request Message (BGCRFS)

The BGCRFS message is used to request product delivery status from the Gatekeeper. Figure 1.8-1 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

FIGURE 1.8-1

1.9 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. Figure 1.9-1 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	Yes
CATALOG_PASSWORD	The catalog user's password on the backside source	A/N	28	Yes
SOURCE_REF	World unique source ID where product will be pushed	A/N/S	47	Yes
PRODTITLE	Product title	A/N	80	Yes
PRODUCT_FILES	This token identifies the **PRODUCT_FILES** sub message. This sub-message may repeat.	A	N/A	Yes
PATH	New product's full path	A/N/S	255	Yes
FILENAME	New product's file name	A/N/S	255	Yes
IP_ADDRESS	New product's ip address	A/N/S	15	Yes

FIGURE 1.9-1

1.10 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, (BGCRFCS). The message format is outlined in figure 1.10-1.

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	47	Yes
RPT_DATE_TIME	Current status date – time	A/N	14	Yes
CATALOG_STATUS	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

FIGURE 1.10-1

1.11 Broadsword Gatekeeper Client Request for Catalog Status (BGCRFCS)

The BGCRFCS message is used to request product catalog status from the Gatekeeper. Figure 1.11-1 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

FIGURE 1.11-1

1.12 Broadsword 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. Figure 1.12-1 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

FIGURE 1.12-1

1.13 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. Figure 1.13-1 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

FIGURE 1.12-1

APPENDIX II

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. 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
                                                         connection parameters
        int                 *pSocket_id,                 - Socket ID returned
        int                 mode,                        - FILE_MODE,
```

		MEMORY_MODE
unsigned char	**hConnect_rsp)	- 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            *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 */
        pSOURCES    pNext_src;          /* next source in list, NULL if no more */
}
```

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
{
        char    *pSource_ref;           /* The source reference which has the
                                           product/report */
        char    *pAccessid;             /* The identifier of the product returned in
                                           the query response */
        char    *pRrds;                 /* Reduced resolution data set value 0-8 */
        char    *pFile_number;          /* Which file of a multi-file product: comma
                                           separated list 1,2,3… or 0 for all files */
        char    *pImagenbr;             /* comma separated list of images to
                                           extact 1,2,3  or 0 for all image(s)*/
        char    *pMetadata;             /* Want source specific metadata file sent
                                           with product Y or N */
        char    *pFormat;               /* destination format or report */
        char    *pCompression;          /* compression mode C3=  JPEG, C2 =
                                           ARIDPCM */
        char    *pRate;                 /* compression rate based on
                                           pCompression */
        char    *pFilename;             /* destination file name */
        char    *pBqs_stmt;             /* BQS for  (MIDB…) reports */
        char    *pNotify_mode;          /* notification mode: Currently supported
                                           IPC notification */
        char    *pEmail_addr;           /* email address for notification: Not
                                           currently supported */
        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 8,12,24 */
        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 */
  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_path;         /* path where new product file exists */
        char        *pProduct_filename;     /* new product file name */
        char        *pProduct_ip;           /* IP address where new product exists */
        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 hRrcs_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
                                                     REQUESTS is >= 0 */
                  char        *pMode,             /* either Q for top ten queries or R
                                                     for top ten product requests */
                  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 BGCTRRM for applicable top ten information.

2.1.8 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 params was
                                                   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 was
                                                   NULL*/

        /* errors unique to the client product request */

        API_PROD_REQ_NULL_PARAM,                /* One of the Product Req params
                                                   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 */
}eAPI_ERRORS;
```

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 hValids table
                char         **hValids,    --- Null terminated char array of valid token types.
                int          mode,         --- IGNORE or FLAG unidentified token types
                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 (STDOUT).

APPENDIX III

3. 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 Query 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.

APPENDIX IV

4. Broadsword ISSO to Gatekeeper Application Program Interface

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
EVENT_DTM	Event date time stamp (ZULU)	A/N	14	Yes
EVENT_DETAILS	Event details sub-message	A	N/A	Yes

Token Name	Meaning or Use	Format	Max Length	Required
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

Apppendix V

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        BVLMF 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 archieved
                            )
```

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_REFUSED,          /* Registration failed by KM */
        API_ADMIN_REG_TO_ITSELF,        /* Registration to own gkpr is wrong */
} tADMIN_API_ERRORS;
```

APPENDIX VI

6. Broadsword Admin to Gatekeeper Application Program Interface

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 Broadsword 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 Broadsword 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 Broadsword 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 Broadsword 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, ADDGRP	A/N	6	Yes
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 Broadsword 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 Broadsword 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 Broadsword Admin Initialize Registration

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

6.24 Broadsword Admin Initialize Registration Status

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

6.25 Broadsword 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_PASSWOR	The one-time registration password	A/N/S	Variable	Yes

D				

6.26 Broadsword Admin Generate Public Key

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

6.27 Broadsword Admin Get Keymasters

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

6.28 Broadsword Admin Get Audit Events

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

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

6.29 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

6.30 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

6.31 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
GKPRIPCPORT	Gatekeeper IP port number	N	6	Yes
GKPRCDIMUSER	Account for CDIM	A/N/S	Variable	Yes
GKPRCDIMPWD	CDIM password	A/N/S	Variable	Yes
GKPRLOGINTIME OUT	Gatekeeper user login timeout	N	Variable	Yes
GKPRALTACCOU NT	Gatekeepers Alternate transfer Account	A/N/S	Variable	Yes

6.32 Broadsword 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 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

6.33 Broadsword 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	A/N/S	Variable	Yes

	source			
HARVESTPORT	IP port number of search engine	N	6	Yes

6.34 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

6.35 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

6.36 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

6.37 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

6.38 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

6.39 Broadsword 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
GKPRIPCPORT	TCP/IP port gatekeeper will listen for connections	N	6	Yes
REG_PASSWORD	Unique one time registration password	A/N	Variable	Yes

6.40 Broadsword 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

6.41 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

APPENDIX VII

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 configure a source of this type.

c. BswdAdminGetSrcs() …This function will return a list of all sources currently configured.

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. BswdAdminGenPubKey() …This function must be called to generate a new public/private key pair.

aa. BswdAdminGetKeymasters() …This function will return the gatekeeper references and registration status of all key-masters that this gatekeeper is registered with.

7.1.3 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
```

7.1.4 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
```

7.1.5 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
```

7.1.6 BswdAdminGetInfo()

Calling Sequence:

```
int BswdAdminGetInfo(
    pADMIN_INFO      pAdmin_info,   - Pointer to tADMIN_INFO returned
                                      by BswdAdminLock,
    char             *pSrc_ref,     - 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 writefailure
            API_ADMIN_INVALID_REFERENCE       -Invalid source reference
```

7.1.7 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
```

7.1.8 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
```

7.1.9 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
```

7.1.10 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
```

7.1.11 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
```

7.1.12 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
```

7.1.13 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
```

7.1.14 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
```

7.1.15 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
```

7.1.16 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.
```

7.1.17 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.

7.1.18 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.

7.1.18 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
```

7.1.19 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
```

7.1.20 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
                                      3 = Producer
    char             *pSrc_ref,     - Source Reference for user
    char             *pUser         - 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_list,            NULL or DATA_LIST to
                                                modify.
                          )
    API_SUCCESS                 -Function successful.
    API_PERM_DENIED         -Current caller does not have ISSO
                                 Permission.
    API_FAILURE                 -Some other failure.

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

7.1.22 BswdAdminGetUserDAC()

Calling Sequence:

```
    int     BswdAdminGetUserDAC(
            tADMIN_INFO     *pAdmin_info,  Returned by
                                                BswdAdminLock().
            char            *pUser_name,            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
                                        BswdAdminLock().
        char            **hReg_info     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
                                        BswdAdminLock().
        int             mode            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
                                        BswdAdminLock().
        char            *pCountry_code, CC of registering GKPR.
        char            *pState,        State of registering GKPR.
        char            *pOrgname,      Org. name of registering
                                        GKPR.
        char            *pPOC,          POC of registering GKPR
        char            *pPOC_email,    POC email address of
                                        registering GKPR
        char            *pGkpr_ip       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 -Key-master Gatekeeper
API_ADMIN_REG_PENDING	-This registration is pending
API_ADMIN_REG_ABORTED	-This registration aborted by client
API_ADMIN_REG_GKPR_CONNECT	-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.

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 BswdAdminGetKeymasters()

Calling Sequence:

```
int     BswdAdminGetKeymasters(
        tADMIN_INFO     *pAdmin_info, Returned by
                                      BswdAdminLock().
        char            **hKeymasters) 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.

7.1.26 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.

Appendix VIII

8.1 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 = Synchronous, 2 = Asynchronous, 3 Email Notification.