NORTH ATLANTIC TREATY ORGANIZATION
ORGANISATION DU TRAITE DE L'ATLANTIQUE NORD

NATO STANDARDIZATION AGENCY (NSA)
AGENCE OTAN DE NORMALISATION (AON)
1110 BRUSSELS

Tel : 0032 (0)2 707 4288

7 April 2003 NSA/0273-AIR/4559

STANAG 4559 (EDITION 1) – NATO STANDARD IMAGE LIBRARY INTERFACE (NSILI)

Reference: AC/224(AG.4)D/95 of 27 June 2000 (Edition 1) (Ratification Draft 1)

1. The enclosed NATO Standardization Agreement, which has been ratified by nations as reflected in the **NATO Standardization Document Database (NSDD)**, is promulgated herewith.

2. The reference listed above is to be destroyed in accordance with local document destruction procedures.

3. APP-4 should be amended to reflect the latest status of the STANAG.

ACTION BY NATIONAL STAFFS

4. National staffs are requested to examine their ratification status of the STANAG and, if they have not already done so, advise the Air Board, NSA, through their national delegation as appropriate of their intention regarding its ratification and implementation.

Jan H ERIKSEN
Rear Admiral, NONA
Director, NSA

Enclosure:

STANAG 4559 (Edition 1)

STANAG 4559
(Edition 1)

NORTH ATLANTIC TREATY ORGANIZATION
(NATO)

NATO STANDARDIZATION AGENCY
(NSA)

STANDARDIZATION AGREEMENT
(STANAG)

SUBJECT: NATO STANDARD IMAGE LIBRARY INTERFACE (NSILI)

Promulgated on 7 April 2003

Jan H ERIKSEN
Rear Admiral, NONA
Director, NSA

RECORD OF AMENDMENTS

No.	Reference/date of Amendment	Date Entered	Signature

EXPLANATORY NOTES

AGREEMENT

1. This NATO Standardization Agreement (STANAG) is promulgated by the Director NATO Standardization Agency under the authority vested in him by the NATO Standardization Organisation Charter.

2. No departure may be made from the agreement without consultation with the tasking authority. Nations may propose changes at any time to the tasking authority where they will be processed in the same manner as the original agreement.

3. Ratifying nations have agreed that national orders, manuals and instructions implementing this STANAG will include a reference to the STANAG number for purposes of identification.

DEFINITIONS

4. Ratification is "In NATO Standardization, the fulfilment by which a member nation formally accepts, with or without reservation, the content of a Standardization Agreement" (AAP-6).

5. Implementation is "In NATO Standardization, the fulfilment by a member nation of its obligations as specified in a Standardization Agreement" (AAP-6).

6. Reservation is "In NATO Standardization, the stated qualification by a member nation that describes the part of a Standardization Agreement that it will not implement or will implement only with limitations" (AAP-6).

RATIFICATION, IMPLEMENTATION AND RESERVATIONS

7. The NSDD gives the details of ratification, implementation, reservations and comments of this agreement. If no details are shown it signifies that the nation has not yet notified the tasking authority of its intentions.

FEEDBACK

8. Any comments concerning this publication should be directed to NATO/NSA – Bvd Leopold III - 1110 Brussels - BE.

STANAG 4559
(Edition 1)

NATO STANDARDISATION AGREEMENT (STANAG)

NATO STANDARD IMAGE LIBRARY INTERFACE (NSILI)

Annexes:
- A. Terms and Definitions
- B. NSIL Interface Operational Concept
- C. NSIL Interface Process Model
- D. NSIL Interface Technical Definition
- E. NSIL Interface Minimum Data Model

The following STANAGS, Military Standards (MIL-STD), ITU-T Recommendations and International Standards contain provisions which, through references in this text, constitute provisions of this STANAG. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this STANAG are encouraged to investigate the possibility of applying the most recent editions of the STANAG, MIL-STDs, Recommendations and Standards listed below. NATO maintains registers of currently valid STANAGS.

Referenced Documents:

ISO/IEC 7498-1	-	Information technology - Open systems interconnection - Basic reference model: The basic model
ISO/IEC 12087-5	-	Information technology - Computer graphics and image processing - Image Processing and Interchange (IPI) - Functional specification: Basic Image Interchange Format (BIIF)
STANAG 4545	-	NATO Secondary Imagery Format (NSIF)
STANAG 7023	-	Air Reconnaissance Imagery Data Architecture
AC/224 (AG/4-SIAR/WG) D56		Air Group IV's Imagery Interoperability Architecture Programme For NATO Reconnaissance Systems.
ADatP-34	-	NATO C3 Technical Architecture, Version 3.0, 15 Dec 2001
AIntP-3	-	Military Intelligence Data Exchange Concept
FIPS Pub 10-4	-	Countries, Dependencies, Areas of Special Sovereignty, and Their Principal Administrative Dependencies

- USIGS Geospatial and Imagery Access Services Specification (GIAS) Version 3.5.1, 6 Aug 2001

- USIGS Common Object Specification (UCOS) Version 1.5.1, 6 Aug 2001

- OMG document formal/99-10-07, CORBA/IIOP 2.3.1, The Common Object Request Broker: Architecture and Specification

Related Documents:

ISO 1000	-	SI units and recommendations for the use of their multiples and of certain other units
MIL-STD-6040	-	Message Text Format
STANAG 1059	-	Distinguishing Letters Geographic Entities for use in NATO
STANAG 3277	-	Air Reconnaissance Request/Task form
STANAG 5500	-	NATO Message Text Formatting System (FORMETS) - ADatP-3
STANAG 7074	-	Digital Geographic Information Exchange Standard (DIGEST)
	-	Combined Forces Support: The evolution in military (intelligence) affairs.
	-	Linked Ops-Intel Center Europe (LOCE) Concept of Operations. July '97
ATP 47	-	Handbook For Air Reconnaissance Tasking And Reporting
	-	CORBAfacilities: The Common Facilities Architecture, Version 4.0, Object Management Group, Framingham, MA, November, 1995
	-	CORBAservices: Common Object Services Specification, Revised Edition, Object Management Group, Framingham, MA, March, 1995
	-	Object Query Service Specification: Joint Submission, Document 95.1.1, Object Management Group, Framingham, MA, December, 1993

AIM

1. The aim of this agreement is to promote interoperability for the exchange of Electronic Secondary Imagery among NATO C^3I Imagery Library Systems. The NATO Standard Imagery Library Interface (NSIL Interface) is the standard interface for querying and accessing heterogeneous imagery libraries maintained by various nations.

2. This standard establishes the requirements for interfacing to heterogeneous imagery libraries. Image Libraries supporting NATO will provide imagery, geospatial information, and product storage mechanisms, which allow users to determine the availability of data and products, and provide the tools to access and retrieve them in a timely manner. A standard interface will enable users to quickly find an image, or information needed to conduct rapid operational missions. Image libraries and the NSIL Interface are envisioned as an augmentation to existing RFI procedures and not as a replacement. There may exist policies (Host Nation) or security and operational restrictions that impose limits on user access. However, technical interfaces will support all authorized users with access to imagery information. The following pages describe the functional architecture and relationship of image library components, and how the library will support digital imagery and geospatial information users supporting NATO operations. The overall goal is for intelligence analysts, imagery analysts, cartographers, mission planners, simulations and operational users, from NATO countries, to have access, from a single workstation, to needed information in a timely manner.

AGREEMENT

3. This NATO Standardisation Agreement (STANAG) is promulgated by the Chairman of the MAS under the authority vested in him by the NATO Military Committee. No departure may be made from the agreement without consultation with the Custodian. Participating nations agree to exchange secondary imagery in accordance with this agreement. Nations may propose changes at any time to the control authority where they will be processed in the same manner as the original agreement. Ratifying nations have agreed that national orders, manuals and instructions implementing this STANAG will include a reference to the STANAG number for purposes of identification.

DEFINITIONS

4. The terms and definitions used in this document are listed in Annex A.

GENERAL SECTION

5. This agreement contains five annexes with associated appendixes. Annex A lists the terms and definitions that apply to this agreement. Annex B explains the NSIL Interface concept of operations. Annex C is the process model. It contains a functional description of the interface in context of interactions between the user and the library. Annex D contains the Technical Definition of the Interface. Annex E contains the Minimum Data Model.

DETAILS OF AGREEMENT

6. The NSIL Interface standardisation agreement is an application program

interface (API) allowing data to be accessed in a common way over a network of heterogeneous imagery databases. This interface is consistent with two and three tier client-server architectures of the NATO C3 Technical Architecture as defined by the NATO Open Systems Working Group. The term API is to be understood in a local sense (e.g. API's between components interfaced on a user desktop) as well as in a distributed sense (e.g. interfaces from legacy or external components using an object request broker (ORB) through interface definition language (IDL) interfaces). Application of this interface assumes a network connection in which database access has already been achieved by another software entity.

IMPLEMENTATION OF THE AGREEMENT

7. This STANAG is implemented by a nation when it has issued instructions that all such equipment procured for its forces will be manufactured in accordance with the characteristics detailed in this agreement.

ANNEX A.

TERMS AND DEFINITIONS

ANNEX A. TERMS AND DEFINITIONS

1. **Acronyms**. The following acronyms are used for the purpose of this agreement.

a.	API	-	Application Program Interface
b.	ATM	-	Asynchronous Transfer Mode
c.	BE	-	Basic Encyclopaedia
d.	BICES	-	Battlefield Information Collection and Exploitation System
e.	BIIF	-	Basic Image Interchange Format. See ISO/IEC 12087-5.
f.	C^3I	-	Command, Control, Communications, and Intelligence
g.	CORBA	-	Common Object Request Broker Architecture
h.	CORCEN	-	Correlation Center
i.	COTS	-	Commercial Off The Shelf
j.	GIAS	-	Geospatial & Imagery Access Services
k.	IEC	-	International Electrotechnical Committee
l.	IPL	-	Image Product Library
m.	ISDN	-	Integrated Services Digital Network
n.	ISO	-	International Organisation for Standardisation
o.	ITU	-	International Telecommunication Union
p.	JPEG	-	Joint Photographic Experts Group
q.	LAS	-	Local Access Subsystem
r.	LDM	-	Logical Data Model
s.	LOCE	-	Linked Ops-Intel Center Europe
t.	MCCIS	-	Maritime Command & Control Information System
u.	MIDB	-	Modern Intelligence Data Base

v. NATO - North Atlantic Treaty Organization

w. NITF - National Imagery Transmission Format

x. NC3TA- NATO C3 Technical Architecture

y. NSIF - NATO Secondary Imagery Format

z. NSIL - NATO Standard Image Library

aa. NSILI - NATO Standard Image Library Interface

bb. OGC - Open GIS Consortium

cc. ORB - Object Request Broker

dd. OSE - Open System Environment

ee. OSI - Open Systems Interconnect model

ff. STANAG - Standardization Agreement

gg. TACOMS - Tactical Communications

hh. WAS - Wide Area Subsystem

2. **Terms And Definitions**. The following terms and definitions are used for the purpose of this agreement.

a. Data communication. The transfer of information between functional units by means of data transmission according to a protocol.

b. Image. A two-dimensional rectangular array of pixels indexed by row and column.

c. Imagery Associated Data. Data which is needed to properly interpret and render pixels; data which is used to annotate imagery such as text, graphics, etc.; data which describes the imagery such as textual reports; and data which support the exploitation of imagery.

d. Interface. (1) A concept involving the definition of the interconnection between two equipment items or systems. The definition includes the type, quantity, and function of the interconnecting circuits and the type, form, and content of signals to be interchanged via those circuits. Mechanical details of plugs, sockets, and pin numbers, etc., may be included within the context of the definition. (2) A shared boundary, e.g., the boundary between two subsystems or two devices. (3) A boundary or point common to two or more similar or dissimilar command and control systems, subsystems, or other entities against

which or at which necessary information flow takes place. (4) A boundary or point common to two or more systems or other entities across which useful information flow takes place. (It is implied that useful information flow requires the definition of the interconnection of the systems which enables them to interoperate.) (5) The process of interrelating two or more dissimilar circuits or systems. (6) The point of interconnection between user terminal equipment and commercial communication-service facilities.

e. Network. (1) An interconnection of three or more communicating entities and (usually) one or more nodes. (2) A combination of passive or active electronic components that serves a given purpose.

f. Open Systems Interconnect Model. This model is defined in ISO/IEC 7498-1.

g. Primary Imagery. Unexploited, original imagery data that has been derived directly from a sensor. Elementary processing may have been applied at the sensor, and the data stream may include auxiliary data.

h. Processed Imagery. Imagery that has been formatted into image pixel format, enhanced to remove detected anomalies and converted to a format appropriate for subsequent disposition.

i. Protocol. (1) [In general], A set of semantic and syntactic rules that determines the behavior of functional units in achieving communication. For example, a data link protocol is the specification of methods whereby data communication over a data link is performed in terms of the particular transmission mode, control procedures, and recovery procedures. (2) In layered communication system architecture, a formal set of procedures that are adopted to facilitate functional interoperation within the layered hierarchy. Note: Protocols may govern portions of a network, types of service, or administrative procedures.

j. Secondary Imagery. Secondary Imagery is digital imagery and/or digital imagery products derived from primary imagery or from the further processing of secondary imagery.

k. Secondary Imagery Dissemination. The process of dispersing or distributing digital secondary imagery.

l. Secondary Imagery Dissemination System. The equipment and procedures used in secondary imagery dissemination.

ANNEX B.

NSIL INTERFACE OPERATIONAL CONCEPT

ANNEX B TO
STANAG 4559
(Edition 1)

ANNEX B

NSIL INTERFACE OPERATIONAL CONCEPT

TABLE OF CONTENTS

.

ANNEX B. NSIL INTERFACE OPERATIONAL CONCEPT

1. **General**.

   a. Nations' image libraries supporting NATO operations have the capability to provide imagery, geospatial information, imagery products, metadata and other imagery derived information in near real time. To exchange this information between the different battlefield nodes that require such information a standard interface to the individual Nations systems libraries is needed. This annex identifies the need for a NATO functional image library interface standard (STANAG) and describes how imagery and imagery derived products are to be exchanged between member nations systems.

   b. Since no current interoperability exists between the various NATO Nations' image libraries certain assumptions as to what will be available when the STANAG becomes operational are identified. For the most part these assumptions are based on the Air Group IV's Imagery Interoperability Architecture Programme for NATO Reconnaissance Systems that is currently being implemented by NATO Nations.

2. **Key Assumptions.**

   a. Image libraries will be capable of handling other forms of digital source information, including tactical unmanned aerial vehicles (UAVs) and weapon systems video sensor formats; commercial satellite imagery; and handheld still and video cameras, but will output imagery/image products in STANAG 4545 format.

   b. Hard copy archives containing imagery and geospatial data and products will continue to exist. If nations choose to archive hard copy imagery they will provide a digital index to the library for users to query on.

   c. Users will have capable tools to display, process, and exploit digital imagery, geospatial information, and products.

   d. The library components will use a wide spectrum of communications capabilities, i.e., Linked Ops-Intel Center Europe (LOCE), Battlefield Information Collection and Exploitation System (BICES). Regardless of the underlying communications medium, the end-to-end communications system must support the Open Systems Interconnection (OSI) protocol architecture.

e. Separate collateral, Top Secret and coalition-releasable security levels will continue to exist. Approved multi-level security (MLS) automated systems with wide functionality are not available at this time, and will not be addressed here.

f. Actions required for user authentication, establishing a secure connection and gaining access to an Image Product Library (IPL) will be outside the scope of the NSIL Interface.

g. The image library interface will be able to operate with legacy systems at each location, where appropriate.

h. All image libraries supporting NATO operations will operate at a NATO releasable classification level. NATO members who construct image libraries will include NATO releasable image libraries and will make those libraries available to other member nations using existing networks for access.

Figure B - 1 CURRENT LIBRARY ARCHITECTURE

3. **Current Imagery Dissemination Architecture.**

a. The traditional reconnaissance cycle process starts and ends with the requester (also referred to as the user) who “requests” imagery acquired intelligence information on a specific target or area of interest. If the item is not held in local archives then an air tasking order (ATO) is issued and passed by communication channels to a NATO reconnaissance wing selected to collect the information. The information required for interpretation is defined prior to take off and includes sensor type, image scale (if appropriate), time-over-target, area of coverage, and all the other requirements to assure the aircraft safety and that the correct target information is collected.

b. The completion of the data collection portion of the reconnaissance cycle is the major difference between softcopy and hardcopy film systems. With hardcopy film systems, the aircraft has to return to a reconnaissance air base to download the film, which is physically transported to the surface station. Currently, within NATO, there is a mix of hardcopy film and softcopy systems. Once the collected data is received at the surface station, it is processed for viewing, exploited, and a report is compiled that is transmitted by communication channels to the requester. With softcopy systems, the collected data can be transmitted directly to the specified surface station and exploited in digital form. A softcopy system will also support the dissemination of the exploited image product to the requester. The dissemination of the image product can be sent directly to that requester or the image product can be stored in a central location, where the requester can access it through a common communications path. An example of this kind of system is shown in Figure B-1, and represents how image products are stored on an image library that users can access through the LOCE network.

c. Today, the backbone of theater intelligence sharing within NATO is the LOCE network. Among 15 NATO member Nations, there are over 283 networked terminals worldwide. LOCE provides connectivity among NATO and other operational users in the form of gateways such as the Allied BICES Initiative, SHAPE’s CRONOS network and SACLANT’s Maritime Command and Control Information System (MCCIS).

d. The LOCE network provides secondary dissemination of imagery products to meet the all-source needs of its users in direct support of NATO operational requirements. This network provides operators access, including the ability to “pull”, NATO classified and NATO releasable imagery files produced by contributing Nations and NATO units. Servers host the Demand Driven Direct Digital Dissemination (5D) imagery server which uses a Sybase Search Query Language (SQL) database to retrieve files loaded by LOCE imagery contributors. All imagery files stored on the 5D servers are saved in the National Imagery Transmission Format (NITF) 2.0 standard.

e. Contributors provide imagery products to the LOCE network using Electronic Light Table (ELT) and 5D Client imagery processing software. Selected contributors load multiple image files with one command to a LOCE 5D server using the 5D Mass Upload utility. ELT is primarily used to process and annotate image products scanned into the LOCE workstation. Some imagery contributors load image files via magnetic media transfer using the LOCE 8mm tape drive application. Imagery products loaded to the server currently originate with any one of seven tactical reconnaissance squadrons/detachments or rotating carriers. Each contributor to the server develops local procedures for its effort and retains responsibility for the products it provides.

4. **Future Imagery Dissemination Architecture.**

a. As more and more nations design, build and implement their respective image libraries, the sources of NATO releasable imagery intelligence will become more distributed (see Figure B-2). Techniques and procedures for the exchange of imagery and data within the reconnaissance cycle must be addressed to take full advantage of the benefits of digital electronics and the new interoperable softcopy system imagery intelligence capabilities they provide. New "tools and techniques" are needed to allow full utilization of digital imagery, from both manned and unmanned platforms, and the near-real-time, high mobility, reconnaissance ground systems that nations will bring into service in the 21st century. Standardization in such an environment is especially important to ensure interoperability in times of crisis. The NATO C3 Technical Architecture (NC3TA) is intended to provide an overall framework for NATO communications that address the military users' requirements for interoperable applications. The NC3TA strategy has been developed to achieve interoperability, maximize the exploitation of commercial-off-the-shelf systems (COTS), and reduce the proliferation of non-standard systems. All future communications and information systems used in NATO are to conform to these standards. [1]

b. One potential tool is a NATO Standard Image Library Interface (NSIL Interface). Instead of uploading NATO Releasable data to a central server, member countries can place selected imagery and imagery products on their local NATO Releasable server. By using the NSIL Interface users will have the capability to access and request image products from all NATO releasable imagery servers regardless of their physical location, or country of origin.

[1] The NC3TA Strategy will not be discussed in detail here as it is beyond the scope of this effort. However, some communication aspects will be addressed elsewhere in this document. A document number is provided in the references for your convenience.

Figure B - 2 FUTURE GENERIC LIBRARY ARCHITECTURE[2]

c. The NSIL Interface will function as a bridge between operational users and image servers supporting NATO. The purpose of this interface is to provide NATO a standardized method for the request and transfer of imagery and imagery products to those battlefield nodes that require such products in a timely manner to accomplish their mission. The NSIL Interface functions provide access support, query/response support, and request/dissemination support. This allows the independent

[2] * See Figure B-4 for a diagram of NATO TACOMS Post-2000

implementation of each imagery server in the theater, all communicating through the NSIL Interface on each library. The interface functions to make each imagery source appear the same, regardless of the nature of the imagery system behind it. This means that each Nation can design an image library to best meet its internal requirements and, by providing the proper interface, allow other imagery users access through the NSIL Interface.

5. **User Interaction with the Image Library**. As noted earlier, an image library provides digital imagery, geospatial information, and products to users to support such things as intelligence production, mission planning, intelligence briefs, and long-term analysis. Essentially an image library provides an interactive augmentation to existing Request for Information (RFI) processes. The following paragraphs describe how users will access and query image libraries and the process to retrieve needed information.

a. Client (Interface). All end users and producers will have access to the image library through a client application. This client is a user interface between an image library and NATO users. This client will interact with an image library through the NSIL Interface. This interface allows the client to use input from the user to search the image library holdings, order products, and transmit digital data on-line. Users at all echelons will employ client software as a means of accessing library components. Client software may be developed and implemented independently by each NATO Nation, yet by implementing the NSIL Interface, interoperability among NATO nation image libraries will be achieved.

b. Log-on Processes. Once users log-on to an image library, they do not need to re-establish their identity to access the NATO releasable data within the library component. The library management will make this determination. Management applications are able to recognize users on the basis of log-on information and will grant access from authorized user retrieval profiles for which they have permission.

c. Shared Information Access. Shared information access has the same controls as access to the appropriate network. Individual organizations control user access to their library. Once granted access to the library the user is free to search through all listings.

d. Information Retrieval Process. Users retrieve information from an image library by submitting a user-generated query. To identify and locate image library information, users perform searches via queries of metadata contained within library indices. Once the information is located, users will review metadata and preview the information before requesting data delivery. Details of this process are discussed in the following paragraphs.

e. Indices and Their Management. Each image library component has an index. Because of the magnitude, diversity, and distributed nature of library holdings, users will access each nation's image library through a

standard system interface that will allow users to interact with the indices in each system in a common manner. Using these indices, users can identify and locate information, and library administrators can manage the distribution, access, and deletion of library holdings efficiently to meet user needs. Users request information from an image library by an ad hoc query request to a specific library component or to all library components. A query is a statement requesting the retrieval of metadata information from an image library.

f. Information Submission Process. Imagery products undergo exploitation to provide value-added information to the end user. The level of information added is dependent on the user requirements for that specific information. Value-added data can go through several iterations of exploitation and can be from any organization level. After this value-added process, the analyst or cartographer places the imagery intelligence products in the appropriate library component. Each NATO nation is responsible for classifying, disseminating and updating its respective products.

g. Metadata. Because of the magnitude, diversity, and distributed nature of library holdings, users rely on library indices to catalog and provide the location of imagery, geospatial information, and products. These library indices present metadata in a format that enables users to query, link, and access library holdings efficiently. All imagery products stored in an image library component must have a minimum set of metadata associated with them as prescribed in Annex E. Image, geospatial information, and product metadata can be entered both manually and automatically. The available metadata from imagery products will populate index fields that sufficiently identify items for a later query, access, and use. Users may make an ad hoc query request to search the metadata index (catalog) on a subset of user specified metadata attributes. Various metadata fields in the index are mandatory and vary depending on data type. This will maximize the usefulness of the image library, since queries require this metadata to ensure the user can find the images they need.

h. Search Function. From a user's point of view the heart of an image library is a search function which provides the user visibility, within access restrictions, to all NATO image library information across the library "network." This function provides the user a tool for review of library inventories. The NSIL Interface provides a standard "bridge" between the user interface that supports this functionality, and the image library itself. The user does not need to know which component contains the data. If the analyst does not specify any particular component the query will search every image library component. Multiple users can simultaneously reference or execute identical queries. After reviewing the search results, selected library holdings may be requested for delivery.

User Access

Data Retrieval

Figure B - 3 User Client/Server Relationship

i. Accessing Library Holdings.

(1) Users create ad hoc queries at an analyst workstation and send them to the image library, through the NSIL Interface (See Figure B-3). The results are returned from the library through the NSIL Interface. Users then review the returned query results listing from the search and determine the information to be requested for delivery to their local workstation. Single items or multiple items from the listing can be displayed. If the requested "hit" is from the hardcopy index, the requested item can either be digitized and sent or reproduced and physically delivered to the user. The users will be notified of when they can expect an order to be filled.

(2) Before requesting delivery of an item from the listing on the workstation, users can review the available metadata and may select a reduced resolution (thumbnail) view of the imagery to determine if the item satisfies their need. A user may opt to view a thumbnail image because the thumbnail takes less time than the request and delivery of an entire file and allows the user to glance quickly at the item. This will ultimately reduce an analyst's search time and reduce demands made on networks.

j. Data Delivery. Users may then request delivery of imagery products from the library. After information has been accessed and delivered, the user can request additional selections from remaining portions of the listing. The listing from a search identifies user-accessible information holdings and will indicate whether this information is immediately accessible or whether a delay will occur. System or network managers restricting access to components for communications or loading reasons may cause such delays. For data delivery, the data will be disseminated either by electronic means or via manual delivery methods specified by the user.

k. Security and Releasibility.

(1) NATO Image library information will be at the lowest permissible classification level (to be determined by individual nation or organization) and is releasable to all NATO members. Product classification will depend on the value-added information, as described in ATP-47. Library owners will determine the

classification level of the storage media for their library components.

(2) There will be separate image library components for TS, collateral information and NATO releasable information, each having its own set of users. For our purpose we are only concerned with the stand-alone "NATO Releasable library" component. Imagery, geospatial information, and products must have release information associated with them and is the responsibility of the contributing nation.

(3) Imagery and product providers will indicate the releasibility of all information provided to any library component. This releasibility will be part of the header information accompanying all formatted information. Without trusted information, the value rules set, which determines the releasibility of information cannot properly operate. Unclassified information, as well as classified information, may have releasibility restrictions.

l. Dissemination Management. NATO standards will be used for data transmitted between and among the various library components and users (STANAG 4545). Communications capacity and loading between the sites is a determining factor for transmission time. The individual organizations are responsible for access to global communications and internal communications. Information will be disseminated either electronically or manually.

m. Library Software. Individual NATO members are responsible for developing and updating all of their image library software at the appropriate point within their architecture. Commonality and interoperability among and between the various library components is achieved through a NATO standard interface. This interface is an extension of each Nation's internal image library software.

n. Equipment. The hardware and software suites necessary to support the library components is entirely determined by individual NATO nations, based on their own needs and requirements. In general, the suites will include central processing units (CPUs), client and server devices, storage devices, input/output devices, connectors, communications interfaces, workstations, and other equipment needed to support the library components within the site infrastructure. This equipment is independent of the NSIL Interface.

6. **Networks and Communications**. As stated in the assumptions, the networks and communications used to support the NSIL Interface will be NC3TA compliant. The NSIL Interface will function over these communications. The following networks are assumed to be NC3TA compliant and functional into the next century, thereby continuing to support intelligence data sharing among NATO nations.

a. Current Network Communications.

(1) LOCE

(a) LOCE provides operational intelligence functions such as situational displays, master reconnaissance lists, operational activity reports, intelligence summaries, intelligence reports, weather data reports, and threat warnings. The LOCE network also provides a number of services; common databases, email, secondary imagery, web servers, secure voice, bulletin boards and other gateways to NATO such as BICES, CRONOS, and MCCIS.

(b) The LOCE communication system is a modified star network with the Correlation Center (CORCEN) at RAF Molesworth UK as the central hub to Remote Communication Servers (RCS) at various locations. The LOCE communications architecture is a multi-node network utilizing commercial Postal Telephone and Telegraph (PTT), MILPATH, microwave and satellite connectivity to LOCE workstations located at NATO, US and Allied commands located throughout Europe and North America. Bandwidth ranges from a minimum 9.6 Kbps to multiple 2.048 Mbps Microwave Line-of-Sight (LOS) links. KIV-7 and KG-84A provide end-to-end encryption.

(c) The network is based on MICOM® multiplexers/routers and regionally located RCS nodes located throughout Europe and North America. They provide connectivity to the regional workstation sites. At each workstation site a MICOM® multiplexes voice and data. This scheme utilizes a fast packet concept that dynamically allocates available bandwidth between the workstation and voice traffic to the RCS node. The dynamic multiplexing allows data full use of the communication line until voice is activated. The voice signal requires 4.8 Kbps of bandwidth. The data fluctuates between 4.8 and 9.6 Kbps while the voice is activated. The voice and data is multiplexed or de-multiplexed and switched at the RCS node through a higher capacity MICOM®. The aggregate is routed to the RAF Molesworth CORCEN over 64 Kbps communication lines to another MICOM® for de-multiplexing.

(2) BICES. BICES exchanges intelligence and related information electronically between individuals, groups and the entire BICES community. Through BICES, one can access, search and retrieve information from archives and National Contribution Data (NCD) Servers. BICES is a shared system comprised of

14 NATO member nations. It provides the capability for participants to easily publish intelligence for mutual sharing. Countries who reside on the network utilize 64 Kbps lease lines. Those who do not utilize dial-up connections. The network features global switching and X.25 MTA/IP Web interfaces. BICES allows the exchange of email between all nations; it is web enabled and is interoperable with LOCE, CRONOS and MCCIS. BICES will soon provide secure voice, open source intelligence, new gateways and new members.

b. Future Network Communications - NC3TA Communications Environment. The NATO military communications environment is complex. It involves many nations, changing concepts of military operations, and many diverse procurement cycles amongst nations and amongst services. The overriding requirement in this environment is interoperability among a diverse array of military command, control and communications systems. NATO has adopted a policy of standardization as the means to improve interoperability among military communications and information systems. This policy, established by the NATO Interoperability Management Plan (NIMP) mandates the creation of NATO Common Interoperability Standards (NCIS) comprising Operational, Procedural, and Technical Interoperability Standards. The NATO C3 Technical Architecture (NC3TA) is intended to provide an overall framework for NATO communications that address the military users' requirements for interoperable applications. The NC3TA strategy has been developed to achieve interoperability, maximize the exploitation of COTS, and reduce the proliferation of non-standard systems. All future communications and information systems used in NATO will conform to these standards.

c. Current NATO Tactical Communications.

(1) Current NATO digital tactical area communications systems can be sorted into three categories: EUROCOM systems, Reconnaissance, Intelligence, and Target Acquisition (RITA) systems and US systems. EUROCOM is a Western European Union (WEU) organization for standardization of tactical area communications systems. The EUROCOM systems are Time Division Multiplex (TDM) systems where each timeslot is normally 16 Kbps and Continuously Variable Slope Delta Modulation (CVSD) is used for voice coding. The specifications are rather complete in the sense that they include most aspects of tactical area communications, i.e. Trunk node Network including Packet Switching, message handling, Command Post Communications Network, Packet Radio etc., but they are not finished in all details. However, many implementations exist, with a variable set of services.

(2) The US systems are also 16 Kbps TDM based, and use a compatible CVSD voice coding. The original RITA system is

TDM-based, but uses 48 Kbps channels and Pulse Code Modulation (PCM) voice coding. The Enhanced RITA (in development) is Asynchronous Transfer Mode (ATM) based and uses 64 Kbps PCM voice coding. To interwork these different tactical systems in NATO, a set of STANAGs for the Multichannel Tactical Digital Gateway (MTDG) has been developed[3]. An Integrated Services Digital Network (ISDN) gateway specification is in development, which will be used for both Strategic-Tactical and Tactical-Public ISDN interworking.

Figure B - 4 NATO TACOMS POST-2000

d. Future NATO Tactical Communications - NATO TACOMS Post-2000.

(1) The former TSGCE SG/11 project group 6 has developed a post-2000 NATO tactical communications architecture. The major components of this tactical communications architecture are the Local Access Subsystem (LAS), the Wide Area Subsystem (WAS) and the Mobile Subsystem (MS) (See Figure B-4). The LAS provides user access to the WAS and local communications services in the local area. It also provides interfaces to civilian commercial networks and military networks other than TACOMS Post-2000.

[3] Please refer to ADatP-34, the NATO C3 Technical Architecture for a complete listing.

(2) The WAS provides long-haul transport and switching of user services through the tactical network as well as local access to mobile user in vicinity of a WAS node. It is envisaged that Asynchronous Transfer Mode (ATM) technology will be utilized in the implementation of the WAS (the NATO ISDN is expected to evolve to a broadband ISDN also using ATM technology). The Mobile Subsystem (MS) provides Radio Access Points to support mobile user access to the WAS.

7. **SUMMARY**. In conclusion, the NSIL Interface is a tool that will serve several functions. It will provide greater interoperability within NATO to efficiently share imagery intelligence from a single workstation. Any NATO user will have access to NATO releasable imagery information regardless of their location on the battlefield (limited only by the communications network). Equally important is the cost of implementing such an interface. The interface will be designed to impact current and future systems as little as possible and to minimize any financial impact. The NSIL Interface is a substantial tool that when implemented will greatly increase NATO theater intelligence sharing capability and enhance theater operations.

ANNEX C TO
STANAG 4559
(Edition 1)

ANNEX C

NSIL INTERFACE PROCESS MODEL

ANNEX C

NSIL INTERFACE PROCESS MODEL

TABLE OF CONTENTS

ANNEX C. NSIL INTERFACE PROCESS MODEL

1. **Introduction**.

a. Purpose. This document presents a model of how the NSIL Interface will function in the context of interactions between library and user. It describes the nature of the interactions that are enabled by the interface and discusses the non-functional issues that affect the interpretation of the interface. This document sets the context for understanding the interface specification in terms of both implementation and operation.

b. Scope. The NSIL Interface is conceived as a read-only interface to an Image Product Library (IPL) and as such, it is not necessary to encapsulate the interactions as formal transactions. Processes involving write operations, such as entering products into the IPL or registering standing orders and queries, are beyond the scope of the NSIL Interface. The process model describes purely the main information flow layer of the NSIL Interface. Error-handling mechanisms are not described in any detail.

2. **NSIL Interface Functional Model**.

a. Processes. The interactions between user and library are grouped into three main generic processes, Login, Query and Order, along with other general utility processes. The processes are illustrated with process diagrams, which show some of the typical uses of the interface. These generic processes are used to describe the capabilities of the NSIL Interface at a conceptual level and do not necessarily represent the exact sequence of actions that would be required by the interface during operation. Similarly, the names of actions shown are not intended to match the operation names in the NSIL Interface Technical Specification [Annex D].

c. Terminology. Before considering the interaction processes, it is necessary to explain some terminology that will be used in the descriptions.

(1) User, Client and IPL. For the purposes of this discussion the User is considered to be the person invoking interactions with the library. The Client is the application (software) that allows the User to interact with the library by providing a mechanism for translating the User's requests into the correct form for transmission to the library and for translating the responses from the library into a suitable form for presentation to the User. The IPL is the library implementation that is responsible for executing the request. Note that the user-interface of the Client is outside

the scope of the NSIL Interface and is implementation dependent.

(2) Interface Manager. The concept of an “Interface Manager” is central to the description of the NSIL Interface functionality. An Interface Manager acts as an intermediary between Client and IPL, receiving requests from the Client and interacting with the IPL to fulfill those requests. It also stores intermediate results from the IPL before passing them on to the Client. There are several types of interface manager, each responsible for a sub-set of the available services, for example performing queries and query related tasks. The client chooses an appropriate interface manager for the task at hand. The NSIL Interface standard specifies the services that are provided by the Interface Managers and the protocols by which clients can request those services. The mechanism by which the Interface Managers provide the requested service is implementation dependent and does not affect the interface between Client and Interface Manager. Figure C-1 shows schematically the relationship between user, client, interface manager and IPL.

(3) Broker. A Library Broker is an application that allows a User to access multiple libraries without the need for multiple Clients. Interaction between a Broker and an IPL uses the NSIL Interface. The IPL will provide the same services and respond in the same manner whether it is being accessed by a Client or a Broker, indeed the IPL need not be aware of the type of application requesting its services. The terms Client and Broker are thus interchangeable as far as the IPL is concerned. As is the case for a Client application, the Broker’s user interface is implementation dependent. A discussion of different architectures for libraries and brokers is held in Section 3.e of this Annex.

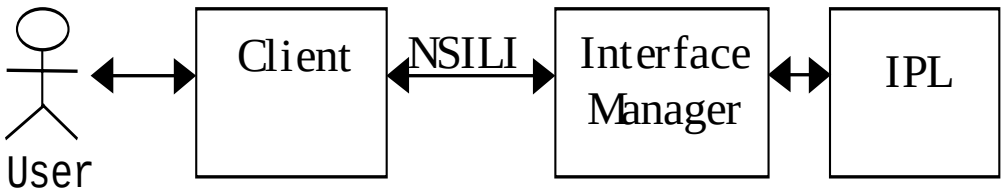

Figure C – 1 Terminology

(4) Metadata. Metadata provides descriptive information relating to products held in the IPL, in the form of a set of attributes with defined values and semantics.

c. The Login Process.

(1) There are two primary activities involved in initiating a session of interaction with an IPL, collectively termed the Login Process. These activities must be accomplished prior to accessing the NSIL Interface.

(2) The first action is to create a network connection to the IPL. This action is performed prior to the interactions covered by the NSIL Interface and will be dependent on the security restrictions and the communications and network technology of the day. From the user's perspective this will most likely be no more complex than entering a library name or IP address.

(3) The second process involves logging on to the library and user validation. This process involves interactions with the outside layer of the library interface, prior to the involvement of any of the Interface Managers described in section 3.b. of Annex D. It is this outside layer that provides access to the Interface Managers upon receipt of a valid request.

d. The Discovery Process. The IPL interface provides services to allow the Client to discover the data model being used by the IPL, along with other descriptive parameters. The Client may use these services at the beginning of a session in order to adopt the configuration necessary for successfully interacting with the IPL, if this information has not already been cached. The data model represents such things as image formats supported and metadata available for performing queries. Examples of descriptive parameters include the IPL name and the version number of the interface it supports. This process of discovering the IPL's data model is a key element of the NSIL Interface definition since it allows for great flexibility in the internal structure of the IPL. A core set of mandatory, standard definition metadata will be supported by all IPLs, which will allow clients to construct queries without first having to discover the data model in use. Optional metadata will be accessed via the discovery mechanism. The core set will allow brokering applications to successfully query multiple IPLs. The IPL implementer must map the core metadata to the data model used by the IPL.

e. The Query Process.

(1) The query process is shown in Figure C-2. This diagram represents a potentially iterative process whereby results from a previous query influence subsequent queries, though each query is in itself a standalone interaction. For the query process the interface functions very much like an Internet search engine, in that a query is sent, and the results come back according to a

request profile. The user will then normally either request some further results, or send a new query.

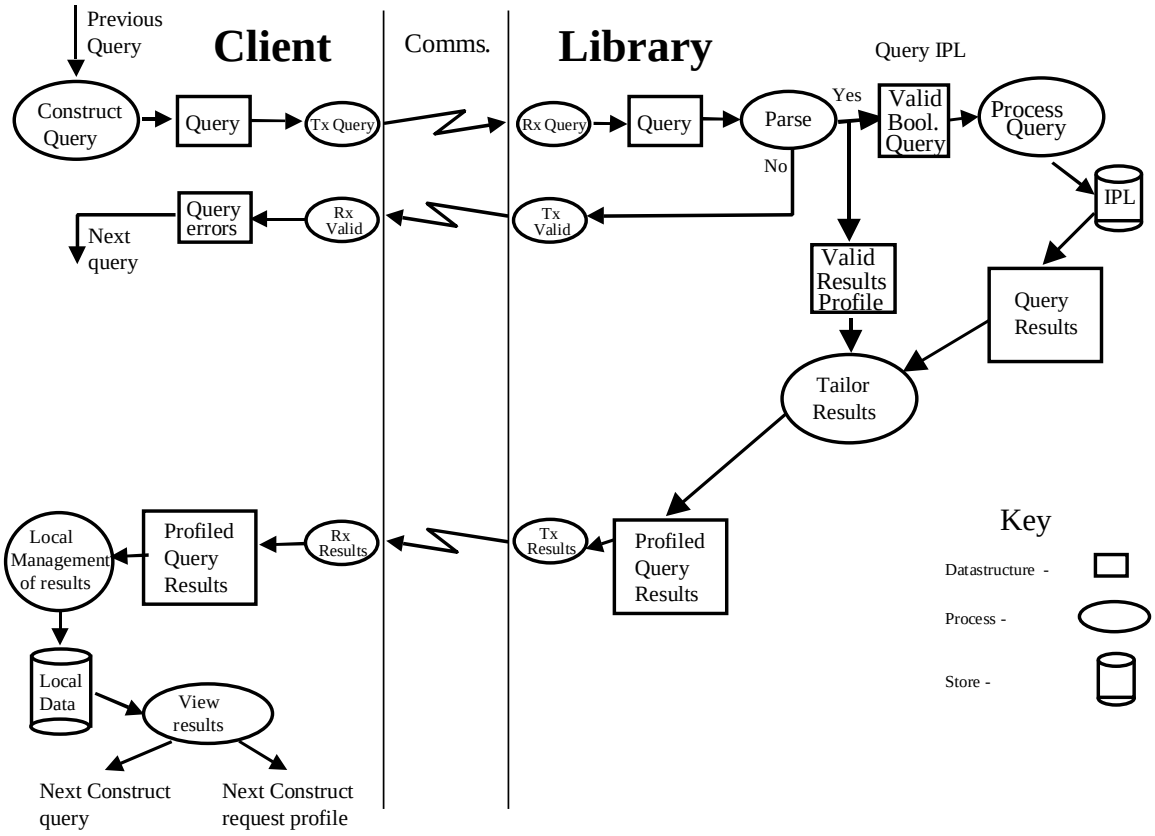

Figure C – 2 The Query Process

(2) The normal information flow shown by the diagram begins with a query being constructed and transmitted to the Interface Manager at the library, where it is parsed. Both user and library will have an interest in the query being syntactically correct. The user checks are assumed to occur prior to transmitting the query and will use knowledge of the library parameters "discovered" during the login process. The syntax of the query is the Boolean Query Syntax (BQS) as defined in Section 4 of the GIAS Specification supporting spatial, Boolean and keyword queries. A Results Profile, or list of desired attributes to be returned including sorting preferences, is sent along with the query. The library checks are performed by the Interface Manager and the Client will be notified if the query is invalid. A validated query will be allocated a reference that will be made known to the Client. Given a syntactically valid query, the IPL performs the search for matching items in the library at its disposal. Upon completion of the search, the IPL produces query results, which are sent to the Interface Manager who stores them ready to send according to the Results Profile. The

Profiled Query Results are tailored to include only the attributes that are in the Results Profile

(3) The profiled query results are then sent back to the Client where they can be organized and managed as required, according to the functionality provided by the Client implementation. The user can then view the results and decide how to continue. The query process can be executed iteratively until the user has identified a set of products that they wish to order, at which time the Order process is invoked.

(4) The NSIL Interface also provides the capability of requesting single or multiple browse-images (thumbnails) to be viewed prior to submitting an order.

(5) Notes.

(a) The user interface for constructing valid queries is a responsibility of the Client application and so is not considered here.

(b) A specialization of a previous Query can potentially be processed by the Interface Manager or the client application without performing a new search in the IPL. Information Management issues such as these are implementation dependent and are left open to the designer of the individual library system.

(c) This style of interface assumes a high availability (in the sense of expected bandwidth) of metadata from either the library or broker accessed by the Client in order to work without irritating the User. In keeping with search engine technology, a broker could keep a compiled database of all libraries' metadata.

f. The Order Process.

(1) The Order Process is shown in Figure C-3. It is similar in many ways to the query process but the Order datastructure is different from the Query datastructure. An Order will specify a set of product identifiers that uniquely define the required products. These product identifiers are returned as part of the earlier Query process. The nature of the product identifiers is discussed in section 3.d. of this Annex.

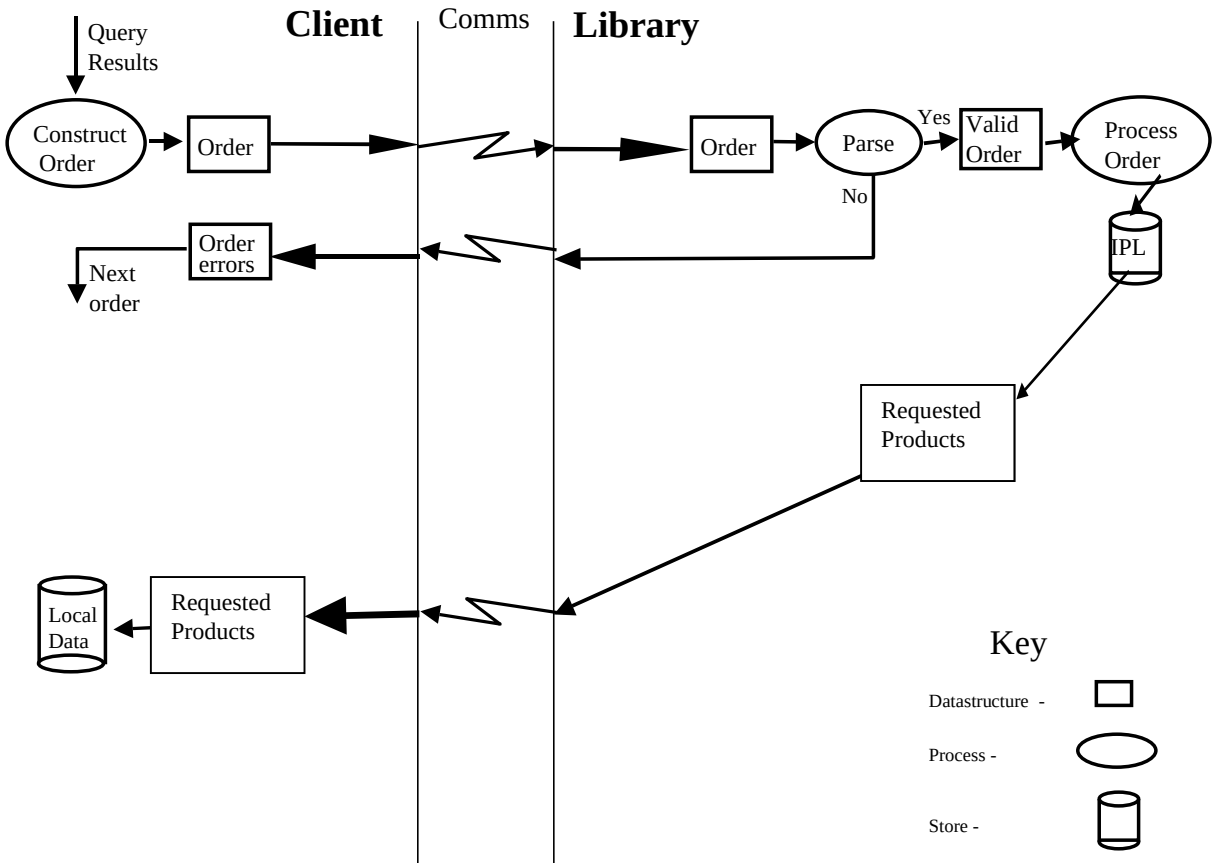

Figure C – 3 The Order Process

(2) To request delivery of a product or set of products, an Order is constructed by the user and submitted to the Interface Manager where it is parsed. As with queries, both user and library will have an interest in the Order being syntactically correct. It is assumed that the client will perform some checking prior to transmitting the Order. To assist further, the Interface Manager provides an operation to "validate" an order, in the sense of syntax checking, prior to submitting it for completion. A validated order will be allocated a reference that will be made known to the Client. The library checks are performed by the Interface Manager and the Client will be notified if the order is invalid. Given a syntactically valid order, the IPL fulfills the order.

(3) In addition to product identifiers, as part of the order the user specifies (for each product) the desired product format including data format and compression, a delivery destination and delivery method (e-mail, FTP or physical), details of any alterations to be made or tailoring to be done prior to delivery, delivery priority and, for physical deliveries, the required media type. Several aspects of creating the order could be automated by the Client application, perhaps using a standard profile. The Client will be notified if the order cannot be filled as requested or if any part of the Order is invalid.

(4) Products may also be retrieved using a metadata field that provides a handle for direct access to the product, such as a URL for http transfer. Product retrieval via the direct access handle does not allow any request for product alteration, compression and so on.

g. General Utility Processes.

(1) Status. The NSIL Interface allows for enquiries regarding the status of an outstanding query or order. The status of a query or order for which the reference is known may be determined, or alternatively a list of all the outstanding queries or orders may be requested and their status queried in turn. The status request is submitted via an Interface Manager and a status datastructure is returned. Status information will include the condition of a request and an estimate for delivery time for orders.

(2) Cancellation. An operation is provided to allow processing of a query or order to be terminated. The cancellation request is submitted via an Interface Manager.

(3) Query and Order Timeouts. Each query or order request being handled by an Interface Manager has a limited lifetime. A client can determine what requests are active, set a default timeout for requests or modify the lifetime of a specific request.

(4) Exit/Logout process. The Exit/logout process is implementation dependent and therefore not a concern for the NSIL Interface specification.

3. **Non-Functional Considerations**.

a. Introduction. There are several important considerations that affect the interpretation of the NSIL Interface specification. In order to achieve interoperability the specification must be interpreted consistently by all implementations. The purpose of this section is to assist the interpretation by describing the context in which the interface specification must be viewed.

b. Communications.

(1) Bandwidth. The bandwidth available to the user from the IPL will vary according to the location and environment of use. Since The NSIL Interface is based in the context of NATO operational use, it must be able to cope with very low bandwidth connections, but be able to take advantage of high-bandwidth links as available.

(2) Continuity and Availability. The transactions between user and IPL should be designed to be as robust as possible against

discontinuities of connection, but should be able to support continuous connection as transparently as possible. This raises issues with respect to the nature of the interface and the services it allows. The NSIL Interface allows product delivery to be achieved by FTP, e-mail and physical delivery, providing the capability for an Internet-like response if the network between user and library can support it, but also allowing graceful degradation to slower response times resulting from limitations in bandwidth and the high demands placed on networks by large-scale imagery. The NSIL Interface also allows for retrieval of products from near on-line and archive storage

c. Product availability.

(1) Product Formats. A variety of product availability should be supported from on-line, through near on-line to archive. The standard product format for the NSIL Interface compliant library is STANAG 4545. A variety of internal processes may be required to convert the stored format of the imagery to STANAG 4545. Some of these internal processes are shown in Figure C – 4.

(2) Format Conversions. Figure C – 4 shows some of the different forms in which imagery and data may be stored within an IPL. Process a) shows the most “natural” way of storing already exploited imagery, as a database having STANAG 4545 format. Some NATO nations also store primary imagery in the same format. However, some other NATO members have a preference for storing primary imagery in STANAG 7023 format (process b), which demands that chip information is supplied with the order which specifies the area of interest of the requester. If information is physically stored on a tape, normally STANAG 7024 will be used as the format (process c). Finally, process d) shows other video formats being used for storage, which requires a different process of extraction prior to sending to the requester.
.

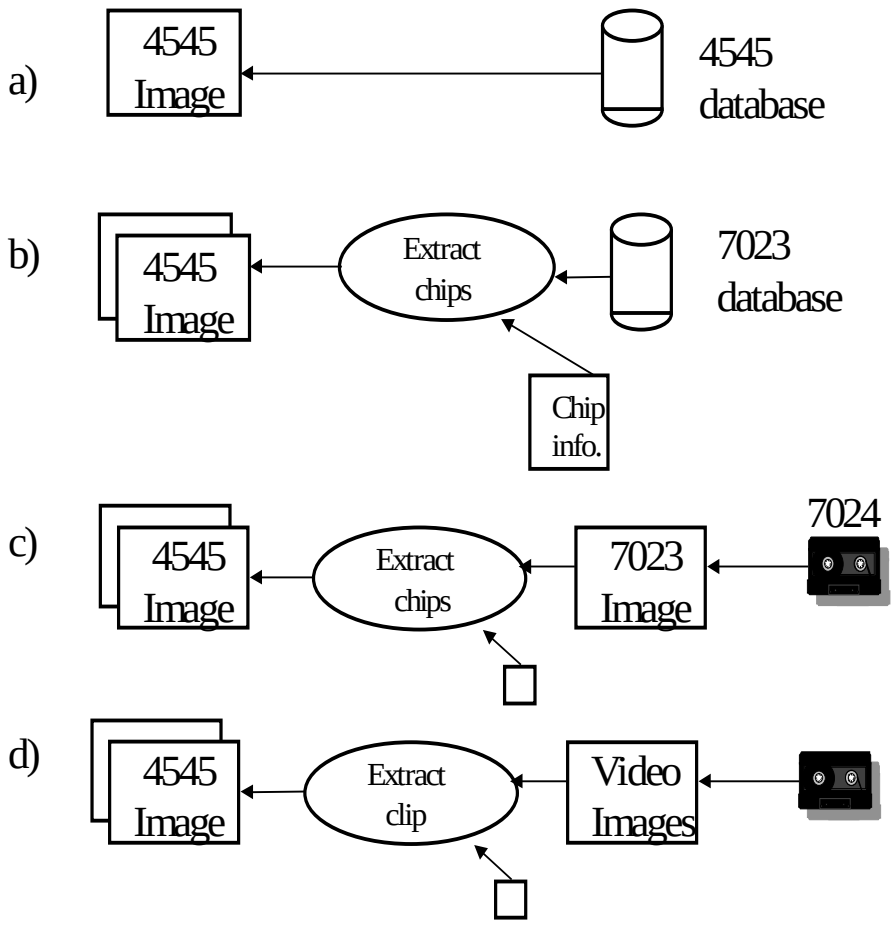

Figure C – 4 Format Conversions

(3) Storage and Retrieval Mechanisms. It should be possible to cope with legacy technologies such as hardcopy prints and reports, as well as archive softcopy. The NSIL interface allows clients to request an estimate of the delay required to make off-line products available for retrieval.

d. Product Identification.

(1) Library ID. The library ID uniquely identifies an individual library and is composed of the Library_name, Library_description, and Library_version as described in Annex D, Table D-1.

(2) Universal Product Identifier (UID). Each product in an IPL must be uniquely identifiable. NSIL Interface core metadata does not include a Unique Product Identifier. However, by combining individual attributes included in the core metadata client applications can differentiate between products or identify duplicate products.

(3) Collection Image Identifier. The definition of the "same" image is not necessarily straightforward since an image may have been exploited several times to look for different targets, or parts of it may be embedded in different RECEXREPs. To resolve this problem, each image has a "Collection Image ID" that remains with the image, however it is processed. This field is a join of Collection DTG, Country owning asset, Mission number and sensor number. For image products consisting of more than one image, the STANAG 4545 notion of a base image is adopted which is referenced.

(4) Target Identifier. Every target in the world of military significance has assigned to it a unique target list number and target name. Target list numbers are pre-assigned and cataloged in the Modern Intelligence Data Base (MIDB). Each target can be uniquely identified by both a target number and a target name. When an image is generated the originator shall be responsible for inserting the Target Name and Number. Both STANAG 4545 and AIntP-3 describe the format for Target ID and Target name.

(a) Target ID. Target ID as a 15 Character field composed of a 10 character BE number and a 5 character Functional Code. The source and format of the field for BE numbers is defined by the MIDB.

(b) Target Name. The Target name field is an 80-character free text format field. The source of the target name is the MIDB.

e. Multiple libraries.

(1) It is assumed that there will be a need for accessing multiple libraries with a single application. This will almost certainly result in the introduction of "library brokers" which will enable a set of libraries to be searched with a single user request. This standard is only concerned with the interface to a single IPL, i.e. the NSIL Interface specifies the minimum set of services that an IPL must make available and the protocol by which those services are requested. An IPL will respond to those requests in the same manner regardless of the nature of the entity (client or broker) accessing the IPL. Coping with the problems of multiple library access is a matter for the broker implementer. However, it is necessary to consider multiple library access in order to ensure that the NSIL Interface provides a sufficient level of functionality to allow a broker to resolve those issues and successfully achieve multiple library access.

(2) Figure C – 5 shows three examples of how multiple libraries may be accessed. In part a), the Client provides a means for the user to logon to individual libraries, effectively as an individual session for each. Each session is supported by the NSIL Interface. In part b), the Client acts as a broker managing login, sending requests to the individual libraries, and collecting the partial results from them. Each Client implementation would be different, and the interaction between user and broker would be non-standard. The broker to library interaction is supported by the NSIL Interface. In part c), the Client accesses a “national broker” which has immediate access to its own national IPL, and can access other national libraries. Again, each national broker implementation would be different, and the national broker to library interaction is supported by the NSIL Interface.

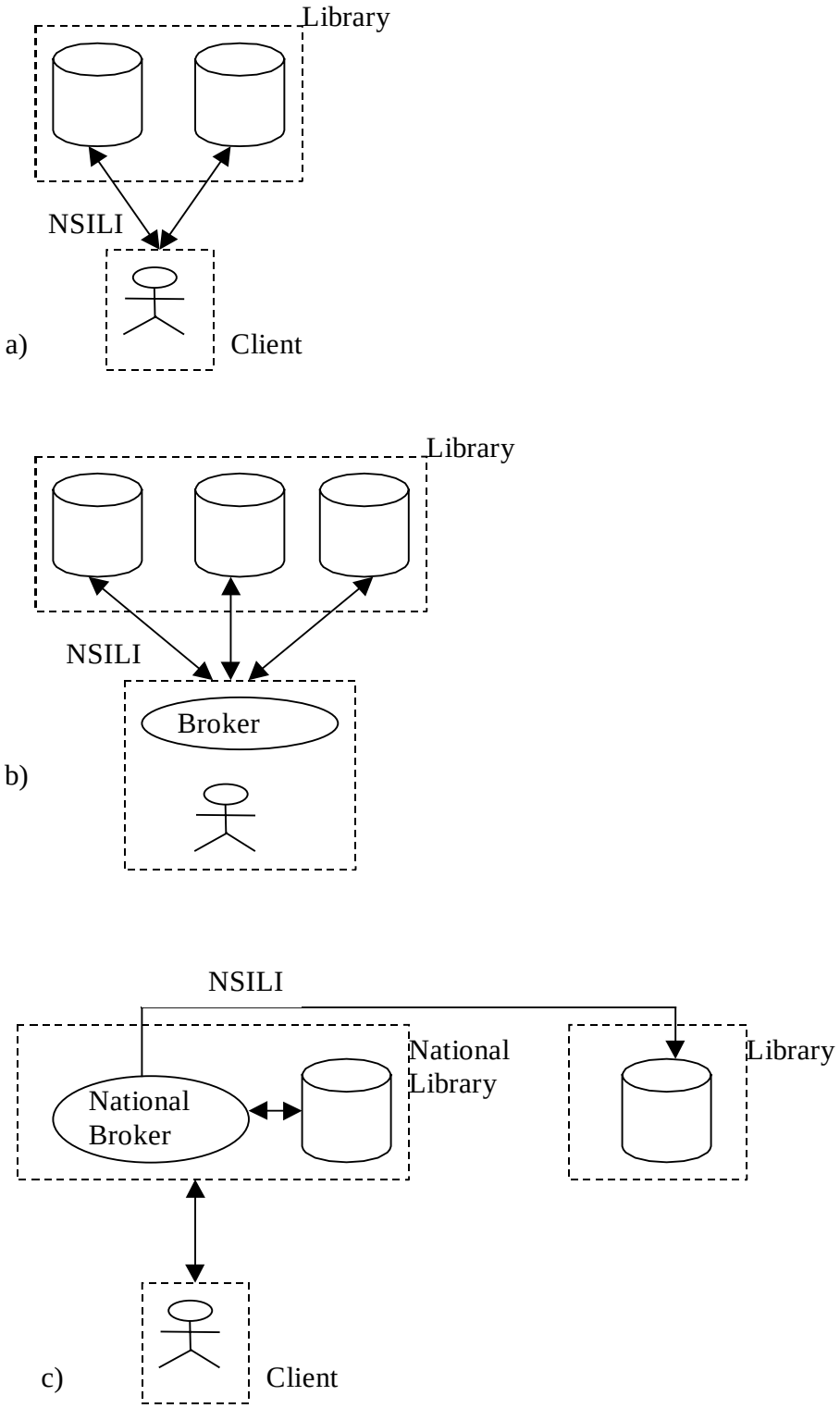

Figure C – 5 Possible Architectures

(3) Accessing multiple libraries would require a mechanism by which the user can identify and select different libraries for access. The NSIL Interface provides a mechanism for discovering the name and interface version of a library.

(4) Duplication of search results is potentially a problem when searching single or multiple libraries, i.e. a search may return multiple “hits” that actually refer to the same image from one or more IPLs. This issue is addressed by the use of Collection Image Identifiers and Universal Product Identifiers.

(5) Management of results returning asynchronously from different IPLs would be a significant issue for a library broker but would not affect the syntax of the interface to individual IPLs. It would produce partially ordered results to a search of multiple libraries, giving fastest first rather than best first ordering.

f. Non-homogeneous Metadata.

(1) An important issue is that of non-homogeneous metadata. Metadata available will vary depending on whether the imagery has been exploited or not, and other factors. It is assumed that some variation in metadata supported by different IPLs is likely to occur.

(2) There may be two types of metadata supported. These are:

(a) Mandatory metadata. These are metadata that are required by the NSIL Interface and must be present for all entries in the library. They are described in Annex E. An example of this type is image date and time.

(b) Optional, metadata. These are metadata that may be implemented by a particular IPL, but that are not required by the NSIL Interface. Using the discovery mechanism provided by the NSIL Interface, Clients can discover the optional metadata attributes supported by a particular IPL.

(3) A brokering application would use only the core set of mandatory metadata in order to ensure that queries are valid across all IPLs. Optional metadata accessed through the discovery mechanism would only be used in a single client to single IPL interaction and would require interpretation by the end-user.

g. Extensibility. The NSIL Interface provides the flexibility to allow for future changes in metadata. The approach is the use of a discovery mechanism for the metadata supported by an IPL. This leaves the set of metadata supported to be defined at run-time, and places more reliance on the client software to adapt to changes in metadata definitions.

ANNEX D TO
STANAG 4559
(Edition 1)

Annex D

NSIL Interface Technical Definition

ANNEX D TO
STANAG 4559
(Edition 1)

ANNEX D

NSIL Interface Technical Definition

TABLE OF CONTENTS

Paragraph	**Title**	**Page**
1.	Order Contents Structure	D-1-1
2.	Image and Ground Coordinates	D-1-4
2.a.	Image Coordinates	D-1-4
2.b.	Ground Coordinates	D-1-5
3.	Packaging Specifications	D-1-6
4.	UID::Product Usage Guidance	D-1-6

ANNEX D. NSIL INTERFACE TECHNICAL DEFINITION

Appendix 1: Recommended Practices and Clarification Information

1. **Introduction**. This Annex is to serve as the Technical Definition of the NSIL Interface. The Technical Definition is based on the Geospatial and Imagery Access Services (GIAS) Specification developed by the United State's National Imagery and Mapping Agency (NIMA). More specifically it is an Implementation Profile of the GIAS specification tailored for NATO operations. It identifies specific GIAS interfaces and any deviations to those interfaces. It contains implementation details, including all parameter values to be passed to and from the interface objects.

a. Scope. The interface defines an application program interface (API) and the services that are provided through the API, as well as communication protocols and data formats.

b. Purpose.

(1) This implementation profile is intended to define clearly the implementation requirements that must be met by vendors and development contractors providing the components that will make up an Image Library System. The GIAS Standard is designed to be as broadly applicable as possible and therefore only contains the most general features and data structures. These general features can be used in many different ways by different domains. To guarantee interoperability within a specific domain, developers using these standards, i.e., developers of client and server implementations must use them in a consistent manner. A "standard profile" or "profile of a standard" serves to document a specific community's standards, conventions, and agreed-to procedures on how that general standard is tailored to that community's requirements. Without these lower level implementation documents to specify the details, it is unlikely that any two developers would make exactly the same interpretation of how to use the standards.

(2) An industry consortium, the Open GIS Consortium (OGC) has committed to defining similar interfaces to support interoperable geoprocessing. It is expected that these interfaces will eventually be adopted by ISO. Interfaces defined by the OGC and eventually adopted by ISO, as they become available, may eventually replace GIAS. In the interim, GIAS will continue to be the basis for the NSIL Interface Technical Specification.

c. Document Organization. This volume is organized as follows:

(1) Section 1 - Introduction contains overview material for this Implementation Profile.

(2) Section 2 - Applicable Documents contains the list of other documents cited or referenced in this Implementation Profile. Together with the requirements specified in Section 4, these documents provide the information needed to understand and implement the implementation profile.

(3) Section 3 - Interoperability Requirements contains the NSIL Interface implementation requirements of STANAG 4559.

(4) Appendix 1: Recommended Practice and Clarification Information

2. **Applicable Documents**.

a. Referenced Documents.

(1) USIGS Geospatial and Imagery Access Specification (GIAS), Version 3.5.1, 6 Aug 2001.

(2) USIGS Common Object Specification (UCOS), Version 1.5.1, 6 Aug 2001.

(3) OMG document formal/99-10-07, CORBA/IIOP 2.3.1, The Common Object Request Broker: Architecture and Specification

b. Other Documents. These documents provide additional information, which may facilitate a reader's understanding of the material contained within this volume:

(1) CORBAfacilities: The Common Facilities Architecture, Version 4.0, Object Management Group, Framingham, MA, November, 1995

(2) CORBAservices: Common Object Services Specification, Revised Edition, Object Management Group, Framingham, MA, March, 1995

(3) Object Query Service Specification: Joint Submission, Document 95.1.1, Object Management Group, Framingham, MA, December, 1993

3. **Interoperability Requirements**.

a. Distributed Computing Services. Image Library Systems shall utilize a common set of Distributed Computing Services for exchanging information as specified in The Common Object Request Broker: Architecture and Specification.

(1) Object Request Broker Interoperability. Object Request Brokers (ORBs) utilized for the NSIL Interface implementations shall support the Internet Inter-ORB Protocol (IIOP) for inter-ORB communication.

(2) Service IORs and Client/Service Binding. Service providers shall provide an FTP server which contains CORBA IORs to support client binding. Clients shall support execution of an anonymous FTP to these servers and retrieve a file(s) containing the IOR strings for use by clients in binding to Services. The IOR file shall be an ASCII file containing only the IOR string. UserID shall be “anonymous” and the password shall be any e-mail address. NOTE: The e-mail address will not be authenticated.

b. Required Interfaces. The required interfaces described in this section are a profile, or a subset with some minor deviations in usage, of the interfaces defined in the GIAS Specification. Deviations from the GIAS specification are described in paragraph 3.c. of this Annex. The NSIL Interface defines seven interface managers that control the functions of the IPL and a top-level interface that provides access to those Interface Managers. Further interfaces define a callback mechanism, a product reference handle and operations common to request objects used by the interface managers as part of their operations. A client interacts with the IPL to select and request access to a specific interface manager, which it then uses to submit requests to the library. Each request returns a request object that the client can use to monitor progress and retrieve results. The client can also obtain information on a specific request or interface manager. The specification of each interface details the operations, data types and error conditions that provide the functionality. A brief overview of these interfaces is given below. The GIAS specification and paragraph 3.d. of this Annex provide the details of each interface. The IDL code for these interfaces is specified in the appendices of the GIAS specification and can be compiled directly into various languages. The information in this section is taken largely from the GIAS specification.

(1) Library Interface. The Library interface is the central point of access to the services provided by the IPL. It provides methods to return library identification information and references to requested Interface Managers.

(2) LibraryManager Interface. The LibraryManager Interface is an abstract interface that defines operations common to other Interface Managers. This comprises methods to determine names and current values of various properties applicable to each manager.

(3) RequestManager Interface. The RequestManager Interface is an abstract interface that defines operations common to all managers that use Request objects as part of their operations. The operations defined allow clients to identify active Request objects and control their lifetimes.

(4) AccessManager Interface. The AccessManager Interface is an abstract interface that defines operations common to managers that allow clients to determine and control the availability of a data set or product, where availability refers to the readiness of that product or data set to be used by other operations provided by the interface manager. It provides operations to determine the usage modes applicable to a specific product or data set, determine or request the availability of a product for those uses, and obtain an estimate of the delay required to make a product available.

(5) OrderMgr Interface. The OrderMgr allows clients to submit orders for products from an IPL and allows orders to be validated prior to submission. Clients use the OrderMgr to specify the desired delivery mechanism and destination, product format, compression, resolution and so on, according to the options available at the IPL.

(6) DataModelMgr Interface. The DataModelMgr Interface provides operations to allow a client to discover and access the data model being used by the library. This Interface provides the mechanism for identifying and using non-standard metadata attributes not defined in the NSIL core metadata set. The DataModelMgr provides two types of operation – access to ancillary data, such as the last modification date and time, and access to the data model itself, such as the set of data views and entities known to the library and the attributes that describe a specific data view or entity.

(7) CatalogMgr Interface. The CatalogMgr Interface provides operations to allow clients to search the IPL for image products of interest based on metadata queries, returning a user defined number of hits. It provides an operation to determine the number of hits that would result for a query, prior to its submission, and allows retrieval of query results in a variety of ways including XML documents and NSIL Interface data structures.

(8) ProductMgr Interface. The ProductMgr Interface provides operations that allow the client to determine characteristics of a specific product and retrieve related files associated with that product, such as thumbnail and overview images.

(9) Request Interface. The Request Interface is an abstract interface that defines operations common to Request objects. All specialized Request objects are derived from this interface. The Request Interface defines operations to allow identification of request objects, monitoring of request status, control of request lifetimes, and a facility for registering callback objects to be notified when a request reaches a terminal state. The specialized request objects all provide operations to invoke completion of request processing, according to the service they provide. The following is a list of the specific request interfaces required:

(a) OrderRequest

(b) SubmitQueryRequest

(c) SetAvailabilityRequest

(d) HitCountRequest

(e) GetParametersRequest

(f) GetRelatedFilesRequest

(10) Callback Interface. The Callback Interface provides a mechanism for client-server communication between Request objects. The Callback Interface provides operations to notify the client of changes in Request object status and to indicate that a callback object will no longer be used.

(11) Product Interface. The Product Interface serves as a reference handle for a specific product in an IPL. No operations are defined for this interface.

(12) Usage. The interfaces defined can be used in a variety of ways to suit the end-goal of the user. In a basic interaction the client would use the Library Interface to gain access to a CatalogManager. An appropriate query would then be submitted and a set of query results would be returned including a Product reference for each result. The ProductMgr could then be used to retrieve thumbnail images for each result, allowing the user to select a specific product for retrieval via the OrderMgr. The query submission, thumbnail retrieval and order submission all result in the generation of a request object which would be used to manage, control and complete the transaction. Valid queries

would be constructed using the NSIL Interface core metadata set, or alternatively after using the DataModelMgr to determine the specifics of any non-standard extension to the data model understood by the IPL.

c. Deviations from GIAS Interfaces.

(1) Library Interface.

(a) Deviation: The *get_other_libraries* method will not be used. The CORBA standard exception NO_IMPLEMENT will be thrown. There is no modification to the IDL.

(b) Reason: It is recommended that knowledge of other libraries and access to other libraries be controlled outside the NSIL Interface. Libraries will require no knowledge of other libraries.

(2) LibraryManager Interface.

(a) Deviation: The *get_libraries, get_property_names* and *get_property_values* methods will not be used. The CORBA standard exception NO_IMPLEMENT will be thrown. There is no modification to the IDL.

(b) Reason: *get_property_names* and *get_property_values* methods are not needed. The *get_libraries* method violates an NSIL Interface assumption that libraries need not know about other libraries. A manager object will support a single library.

(3) DataModelMgr Interface.

(a) Deviation: The ancillary selectors *get_alias_categories*, *get_logical_aliases* and *get_logical_attribute_name* methods will not be used. The CORBA standard exception NO_IMPLEMENT will be thrown. There is no modification to the IDL.

(b) Reason: Discovery can be achieved directly by using the data model set of selectors. There is no requirement for creating aliases to attribute names in the Logical Data Model (LDM) which the ancillary set of selectors support.

(4) SubmitQueryRequest Interface

(a) Deviation: The complete_ StringDAG_results method will not be used. The CORBA standardexception NO_IMPLEMENT will be thrown. There is no modification of the IDL.

(b) Reason: The required method, complete_DAG_results, is sufficiently expressive that no further methods are required.

d. Implementation Details.

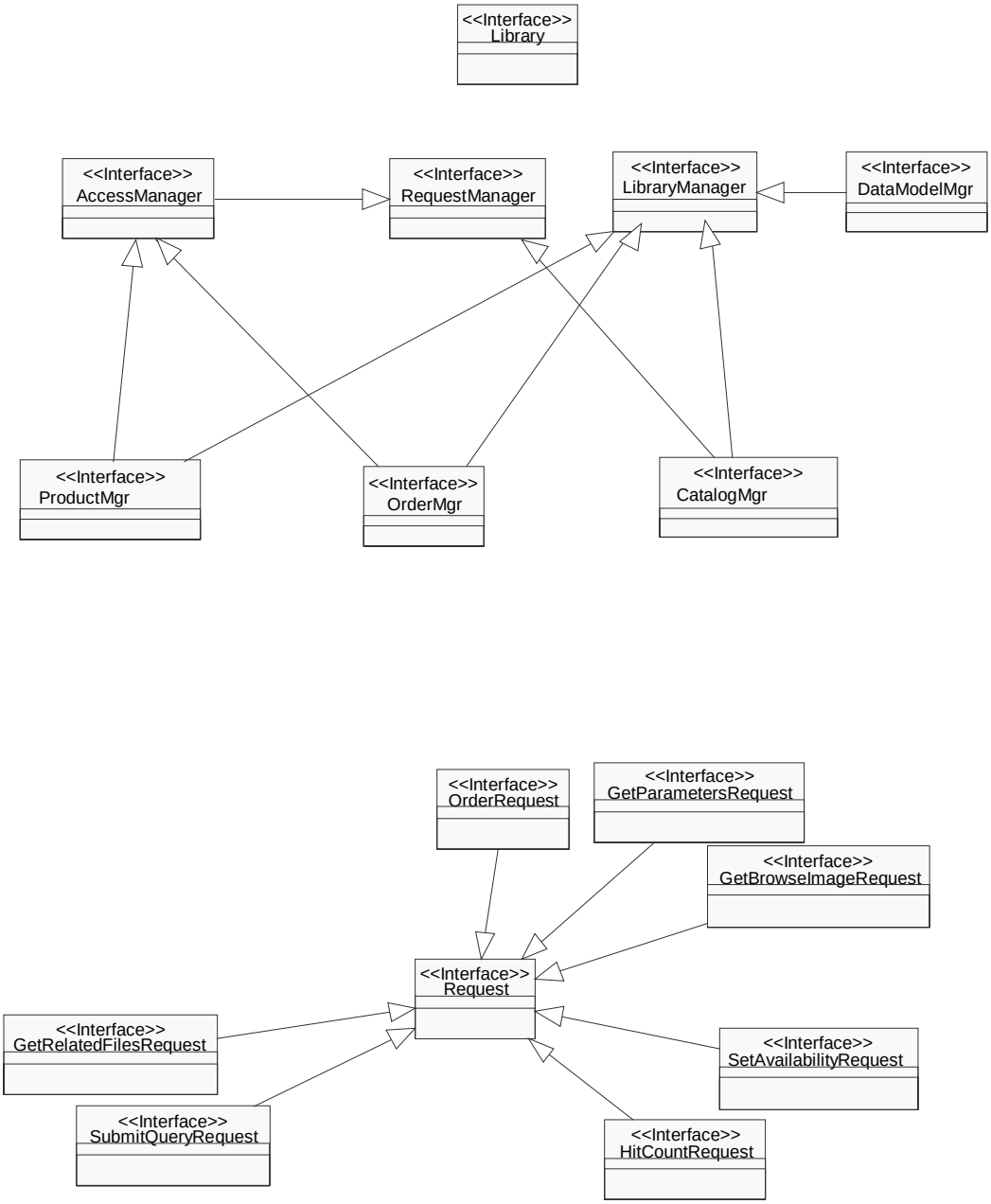

Figure D-1 UML Static Class Diagram of NSIL Interface Profile of GIAS

(1) Relationship to a Logical Data Model. A NSIL Interface implementation is dependent upon a LDM, which should be defined for each implementation. The LDM represents the

natural characteristics of the data and the relationships among the data entities, rather than how the user prefers to query or view the data. Hence, a layer of user view abstraction is required. By using this natural data approach, the impacts of changes in the user's perspective of the data are isolated to the user interface, and do not impact the database design. The Logical Data Model defines the attributes (data elements), structure and relationships of the attributes, and the domain of acceptable values for the attributes. The LDM is implementation dependent, but must contain as a minimum the entities, relationships and attributes defined in Annex E. A "NSIL_CORE" view will also need to be defined which will allow a user view of this core metadata. By levying this requirement clients will be able to construct valid queries across multiple libraries. IPL implementers have the flexibility to add to the LDM as required and can provide unique views of their own. The implementation details described in the next section make reference to this LDM.

(2) The implementation details for each interface are listed in the tables below. Each table heading lists the interface followed by any inherited interfaces e.g. CatalogMgr inherits from RequestManager and so is described as "CatalogMgr : RequestMgr."

Table D-1 Implementation Detail for Library Interface

Method	**Return/ Parameter**	**Type**	**Desc/Value(s)**
get_manager_types	[return]	**ManagerTypeList** sequence<string>	"OrderMgr" ; "CatalogMgr" ; "ProductMgr" ; "DataModelMgr"
get_manager	**LibraryManager**	[Object]	N/A
	Manager_type	**ManagerType** <string>	"OrderMgr" ; "CatalogMgr" ; "ProductMgr" ; "DataModelMgr"
	Access_criteria	**AccessCriteria**	Not Used
get_library_ description	[return]	**LibraryDescription**	**Library_name**<string>Valid Library identifier will be implementation dependent. Recommend using hostname. **Library_description**<string> Organizational affiliation of Library. Implementation unique. Recommend concatenation of country code digraph and organizational unit code separated by a '|'. Example "US|AFRL/IFEC" **Library_version** <string>Text string "NSILI" with STANAG version number separated by '|'. Example: "NSILI|1.0"
get_other_libraries			Throw CORBA exception NO_IMPLEMENT.

Table D-2 Implementation Detail for LibraryManager Interface

Method	Return/ Parameter	Type	Desc/Value(s)
get_property_names			Throw CORBA exception NO_IMPLEMENT.
get_property_values			Throw CORBA exception NO_IMPLEMENT.
get_libraries			Throw CORBA exception NO_IMPLEMENT.

Table D-3 Implementation Detail for Interface RequestManager

Method	Return/ Parameter	TYPE	Desc/Value(s)
get_active_requests	[return]	RequestList	N/A
get_default_timeout	[return]	<unsigned long>	As defined in GIAS
set_default_timeout	new_default	<unsigned long>	As defined in GIAS
get_timeout	[return]	<unsigned long>	As defined in GIAS
	Request	Request	N/A
set_timeout	**Request**	Request	N/A
	new_lifetime	<unsigned long>	As defined in GIAS
delete_request	**Request**	Request	N/A

Table D-4 Implementation Detail for Interface AccessManager : RequestManager

Method	Return/ Parameter	Type	Desc/Value(s)
get_use_modes	[return]	UseModeList	"OrderAccess"; "ProductAccess" Product available for OrderMgr use (i.e. available for ordering) and product available for ProductMgr use, respectively.
is_available	[return]	<boolean>	TRUE if product is available for the specified use, otherwise FALSE.
	Product	UID::Product	N/A
	use_mode	UseMode	"OrderAccess"| "ProductAccess"
query_availability_delay	[return]	<unsigned long>	As defined in GIAS.
	Product	UID::Product	N/A
	Availability_Requirement	**Availability Requirement**	As defined in GIAS.
	use_mode	UseMode	"OrderAccess"| "ProductAccess"
get_number_of_Priorities()	[return]	CORBA:short	As defined in GIAS.
set_availability	[return]	SetAvailability Request	N/A
	Products	UID::ProductList	N/A
	Availability_Requirement	**Availability Requirement**	As defined in GIAS.
	priority	CORBA:short	N/A
	use_mode	UseMode	"OrderAccess"| "ProductAccess"

Table D-5 Implementation Detail for Interface OrderMgr : LibraryManager, AccessManager

Method	Return/ Parameter	Type	Desc/Value(s)
get_package_ specifications	[return]	NameList	Available package options in response to NameList of attributes associated with an Order. See clarification of format types in Appendix 1, section 1.4: “TARUNC” | “FILESUNC” | “TARZIP” | “FILESZIP” | “TARGZIP” | “FILESGZIP” | “TARCOMPRESS”| “FILESCOMPRESS”
validate_ order	[return]	**ValidationResults**	Results table provided in GIAS Table 2-1 **valid**<boolean>TRUE if order is valid FALSE if order is invalid **warning**<boolean>TRUE if there is a warning message with a valid order, otherwise false ("**details**" contains warning message of proposed order) **details**<string>Contains description of error or warning
	order	OrderContents	The OrderContents structure is defined in GIAS. The contents of the "type" and "Desc/Value(s)" columns are provided in Appendix 1, Table D-1-1.
	properties	**PropertyList**	An ordered sequence of Name-Value pairs
order	[return]	**OrderRequest**	N/A
	order	OrderContents	The OrderContents structure is defined in GIAS. The contents of the "type" and "Desc/Value(s)" columns are provided in Appendix 1, Table D-1-1.
	properties	**PropertyList**	GeographicDatum=<string> Datum for GeoRegion of the Alteration Spec;default=“WGS84”

Table D-6 Implementation Detail for Interface DataModelMgr : LibraryManager

Method	Return/ Parameter	Type	Desc/Value(s)
get_data_model_date	[return]	UCO::AbsTime	Return date time group in GMT when the data model was last changed.
get_view_names	[return]	ViewList	Provides list of views supported by the Library. The list of views must at least include the “NSIL _CORE” view, which sees the entire Minimum Data Model in Annex E. Other unique views may also be defined to view other parts of the LDM.
	Properties	**PropertyList**	Not Used.
get_attributes	[return]	**AttributeInformation List**	AttributeInformationList where: attribute_name=<string> identified in the LDM (refer to parenthetical values in the Data Model in Annex E), attribute_type=<AttributeType>enumeration of Data Type identified in the LDM, attribute_domain =<Domain> identified in the LDM, attribute_units=<string> identified in the LDM, mode = <RequirementMode> enumeration of Requirement Mode identified in the LDM, description = <string> identified in the LDM, sortable = <Boolean> identified in the LDM, updateable = <Boolean> identified in the LDM The Data Model in Annex E contains a mandatory subset of the LDM. Attribute information "sortable" and "updateable" were not included in the Annex E Data Model and are left up to the implementor.
	view_name	ViewName (<string>)	Valid Logical Data Model view supported by library.
	Properties	**PropertyList**	Not Used. Empty list.

get_queryable_ attributes	[return]	**AttributeInformationList**	AttributeInformationList where: attribute_name=<string> identified in the LDM (refer to parenthetical values in the Data Model in Annex E), attribute_type=<AttributeType>enumeration of Data Type identified in the LDM, attribute_domain =<Domain> identified in the LDM, attribute_units=<string> identified in the LDM, mode = <RequirementMode> enumeration of Requirement Mode identified in the LDM, description = <string> identified in the LDM, sortable = <Boolean> identified in the LDM, updateable = <Boolean> identified in the LDM The Data Model in Annex E contains a mandatory subset of the LDM. Attribute information "sortable" and "updateable" were not included in the Annex E Data Model and are left up to the implementor.
	view_name	ViewName(<string>)	Valid Logical Data Model view supported by library.
	Properties	**PropertyList**	Not Used.
get_entities	[return]	**UCO::EntityGraph**	EntityGraph containing entities and relationships for the data view requested.
	view_name	ViewName(<string>)	Valid Logical Data Model view supported by library.
	Properties	**PropertyList**	Not Used.
get_entity_attributes	[return]	**AttributeInformationList**	AttributeInformationList where: attribute_name=<string> identified in the LDM (refer to parenthetical values in the Data Model in Annex E), attribute_type=<AttributeType>enumeration of Data Type identified in the LDM, attribute_domain =<Domain> identified in the LDM, attribute_units=<string> identified in the LDM, mode = <RequirementMode> enumeration of Requirement Mode identified in the LDM, description = <string> identified in the LDM, sortable = <Boolean> identified in the LDM, updateable = <Boolean> identified in the LDM The Data Model in Annex E contains a mandatory subset of the LDM. Attribute information "sortable" and "updateable" were not included in the Annex E Data Model and are left up to the implementor.
	aEntity	Entity(<string>)	Valid LDM entity.
	Properties	**PropertyList**	Not Used.
get_alias_categories			Throw CORBA exception NO_IMPLEMENT.

get_logical_aliases			Throw CORBA exception NO_IMPLEMENT.
get_logical_ attribute_name			Throw CORBA exception NO_IMPLEMENT.
get_associations	[return]	AssociationList	As defined in GIAS. Returns a list of Association structures that contain the descriptions of the associations
	properties	**PropertyList**	Not Used.
get_max_vertices	[return]	<unsigned short>	As defined in GIAS. Returns the maximum number of vertices supported in geospatial queries
	properties	**PropertyList**	Not Used.

Table D-7 Implementation Detail for Interface CatalogMgr : LibraryManager, RequestManager

Method	**Return/ Parameter**	**Type**	**Desc/Value(s)**
submit_query	[return]	**SubmitQuery Request**	N/A
	aQuery	Query	Valid BQS query as defined in Section 4 of the *Geospatial and Imagery Access Services (GIAS) Specification.*
	result_attributes	**UCO::NameList**	attribute name(s) to be returned. Attributes are from those available in the *Logical Data Model*
	sort_attributes	**SortAttributeList**	attribute name(s) to be sorted on with a flag denoting whether ascending or descending. The order in the list will indicate the precedence. Attributes are from those available in the *Logical Data Model.*
	properties	**PropertyList**	**GeographicDatum**<string> Datum of the geospatial data in the BQS query; default="WGS84"
submit_complex_Query			Throw CORBA exception NO_IMPLEMENT.
hit_count	[return]	**HitCountRequest**	N/A
	aQuery	Query	Valid query as defined in Section 4 of the *Geospatial and Imagery Access Services (GIAS) Specification*
	properties	**PropertyList** (UCO::NameValueList)	**GeographicDatum**<string> Datum of the geospatial data in the BQS query; default="WGS84"
hit_count_complex			Throw CORBA exception NO_IMPLEMENT

Table D-8 Implementation Detail for Interface ProductMgr : LibraryManager, AccessManager

Method	Return/ Parameter	Type	Desc/Value(s)
get_parameters	[return]	**GetParameters Request**	N/A
	product	UID::Product	Product for which data is being requested
	desired_parameters	**UCO::NameList**	Attribute name(s) from *Logical Data Model.* "ALL" will return all attributes associated with the product. "ORDER" will return attributes associated with product ordering. This includes: order.pSpec.packaging_format_and_compression -See Table D-1-2 for options. order.prod_list.mTypes -See Table D-1-4 for options. order.prod_list.aSpec.pf -See Table D-1-5 for options. order.prod_list.aSpec.ps.imgform -See Table D-1-6 for options. order.prod_list.aSpec.ps.comp -See Table D-1-6 for options. order.prod_list.aSpec.ps.rrds -See Table D-1-6 for options. Cataloging is assuming single image products only.
	properties	**PropertyList**	Not Used
get_related_file_types	[return]	**RelatedFileTypeList**	List of file types available for the product. Related file types are described in Table D-1-8.
	prod	UID::Product	Product for which available file types is being requested
get_related_files	[return]	**GetRelatedFilesRequest**	N/A
	products	UID::ProductList	List of products for which the related files are to be retrieved
	location	UCO::FileLocation	Location where the related files are to be put; FileName will be blank
	type	**RelatedFileType**	The type of related file to be retrieved
	properties	PropertyList	Not Used

Table D-9 Implementation Detail for Request Interface

Method	Return/ Parameter	Type	Desc/Value(s)
get_request_ description	**[return]**	**UCO::RequestDe scription**	**User_info** <string>Previously supplied by client via set_user_info method [Default is empty] **Request_type**<string> “OrderRequest” | “SubmitQueryRequest” | “SetAvailablityRequest” | “HitCountRequest” | “GetParametersRequest” | “GetRelatedFilesRequest” | **request_info**<string> Supplied by server [Default is empty] **request_details<**UCO:: NameValueList> See Tables for specific request interface.
set_user_info	**message**	<string>	Supplied by client [Default is empty]
get_status	[return]	**UCO::Status**	**Completion_state<**UCO::State**>** As defined in UCOS. **Status_message** <string>Human-readable explanatory message
get_remaining_delay	[return]	DelayEstimate	As defined in GIAS.
cancel	N/A	N/A	N/A
register_callback	[return]	**CallbackID**	N/A
	acallback	**CB::Callback**	
free_callback	**id**	**CallbackID**	N/A
get_request_manager	[RETURN]	**RequestManager**	N/A

Table D-10 Implementation Detail for Interface OrderRequest : Request

Method	Return/ Parameter	Type	Desc/Value(s)
complete	[return]	**UCO::State**	As defined in UCOS.
	prods	**DeliveryManifest**	DeliveryManifest is a structure defined in the GIAS providing the package names and the package elements: **package_name=<**string**>**names of the package PackageElementList: *package_element* <UID::Product> file names <UCO::NameList> The DeliveryManifest will be an ordered sequence: For NITF: file_id*, file_id.rl, file_id.r2, etc…(note *file_id is RO)

get_request_ description	[return]	**UCO::RequestDes cription**	**User_info**<string> Previously supplied by the client via the set_user_info method [Default is empty] **Request_type** <string> "OrderRequest" **request_info**<string> Supplied by server [Default is empty] **request_details<UCO::NameValueL ist>** *order<OrderContents> structurecontaining order GeographicDatum<string> OrderSize=<double>Size (in MB/megabytes) of product(s) requested in order. TimeSubmitted=<UCO::AbsTime>*

Table D-11 Implementation Detail for Interface SubmitQueryRequest : Request

Method	**Return/ Parameter**	**Type**	**Desc/Value(s)**
set_number_of_hits	hits	<unsigned long>	As defined in GIAS.
NOTE: Of the following three methods for retrieving query results, **complete_DAG_results** will be **required** to be implemented and the rest will be optional. For those methods not implemented, the CORBA exception NO_IMPLEMENT will be thrown.			
complete_DAG_resu lts	[return]	UCO::State	As defined in UCOS.
	results	UCO::DAGList	DAGs as defined in GIAS of type QueryResult. Node 0 shall be a product reference. Nodenames shall be full attribute names from *Logical Data Model.* Cardinality fields of DAG nodes are not used.
complete_XML_resul ts	[return]	**UCO::State**	As defined in UCOS.
	results	**UCO::XML.doc ument**	Results are as defined in the Document Type Definitions (DTDs) (TBD)
complete_StringDA G_results			Throw CORBA exception NO_IMPLEMENT.

get_request_ description	[default]	**UCO::Request Description**	**user_info**<string> Supplied by client [Default is empty] **request_type**<string> "SubmitQueryRequest" **request_info** <string> Supplied by server [Default is empty] **request_details<**UCO::NameValueList> *view_name*<ViewName> *query*=<Query> query string *result_attributes*<UCO::NameList> *sort_attributes*<SortAttributeList> *RelatedFileDirectory*<UCO::FileLocation> GeographicDatum<string>

Table D-12 Implementation Detail for Interface SetAvailabilityRequest : Request

Method	Return/ Parameter	Type	Desc/Value(s)
complete	[return]	**UCO::State**	As defined in UCOS.
get_request_ description	[return]	**UCO::RequestDe scription**	**user_info**<string> Supplied by client [Default is empty] **request_type**<string> "SetAvailabitiyRequest" request_info<string> Supplied by server [Default is empty] **request_details<UCO::NameValueList>** *product*=<Product> product object reference *availability_requirement*<AvailabilityRequirement> *use_mode*= <UseMode**>**

Table D-13 Implementation Detail for Interface HitCountRequest : Request

Method	Return/ Parameter	Type	Desc/Value(s)
Complete	[return]	**UCO::State**	As defined in UCOS.
	number_of_hits	<unsigned long>	As defined in GIAS.
get_request_ description	[return]	**UCO::RequestDe scription**	**User_info**<string> Supplied by client [Default is empty] **Request_type**<string> "HitCountRequest" **Request_info** <string>Supplied by server [Default is empty] **Request_details**<UCO::NameValueList**>** *Data_view***<DataView>** *query*=<Query> query string *GeographicDatum*<string>

Table D-14 Implementation Detail for Interface GetParametersRequest : Request

Method	Return/ Parameter	Type	Desc/Value(s)
complete	[return]	**UCO::State**	As defined in UCOS.
	parameters	**UCO::DAG**	DAGs as defined in GIAS of type QueryResult. One node shall be a product reference. Node names shall be a subset derived from the order view of the *Logical Data Model.* Cardinality fields of DAG nodes are not used. Cataloging is assuming single image products only.
get-request-description	[return]	**UCO::Request Description**	**user_info** <string> Supplied by client [Default is empty] **request_type**<string> "GetParametersRequest" **request_info**<string> Supplied by server [Default is empty] **request_details** <UCO::NameValueList> *product*=<UID::ProductList>Product object reference *desired_parameters*<UCO::NameList>

Table D-15 Implementation Detail for Interface GetRelatedFilesRequest : Request

Method	Return/ Parameter	Type	Desc/Value(s)
complete	[return]	**UCO::State**	As defined in UCOS.
	locations	**UCO::NameList**	As defined in UCOS.
get_request_description	[return]	**UCO::RequestDescription**	**user_info**<string> Supplied by client [Default is empty] **request_type** <string> "GetRelatedFilesRequest" **request_info** <string>Supplied by server [Default is empty] **request_details** <UCO::NameValueList> *products*<UID::ProductList> *location*<UCO::FileLocation> *type*< **RelatedFileType**> Related file types are described in Table D-1-8.

Table D-16 Implementation Detail for Callback Interface

Method	Return/ Parameter	Type	Desc/Value(s)
notify	**description**	**RequestDescription**	**user_<string>** Supplied by client [Default is empty] **request_type <string>** "Callback" **request_info <string>** Request identifier **request_details** <**UCO::NameValueList>**
release	N/A	N/A	N/A

APPENDIX 1 TO
ANNEX D TO
STANAG 4559
(Edition 1)

APPENDIX 1

RECOMMENDED PRACTICES AND CLARIFICATION INFORMATION

[Note: Intent of this section is to provide additional or clarifying information on specific interfaces, methods, or files (e.g., formats, products, data types, etc…) to assist developers. Interpretations, recommended procedures, or clarifications are provided as value-added to the limiting structure of section 3 and of the tables therein.]

1. **OrderContents Structure**. The OrderContents structure information is redundant to some of the tables in Annex D.,section 3.d.(2). The information provided is intended to provide clarification beyond that in the tables which are limited due to the table formats.

Table D-1-1 order <OrderContents>

Element/Type	Desc/Value(s)
originator <string>	UserID of person or system submitting order. Supplied by client.
tSpec <TailoringSpec>	As defined in GIAS. Not Used where cardinality is (0..1)
pSpec <PackagingSpec>	Supplied by client See Table D-1-2 for details.
needByDate <UCO::AbsTime>	Date product is needed by. As defined in UCOS.
operatorNote <string>	
orderPriority <short>	Priority to be assigned to this order. As defined in GIAS section 2.3.4.3 the priority system is a string value range 0-100 where value = 0 is a crisis or highest priority, and value = 100 is the Null value.
prod_list <ProductDetailsList>	Sequence of ProductDetails structures. See Table D-1-3 for details.
del_list <DeliveryDetailsList>	Sequence of DeliveryDetails structures. See Table D-1-7 for details.

Table D-1-2 order.pSpec <PackagingSpec>

Element/Type	Desc/Value(s)
package_identifier <string>	An identifier for the package so the client can identify the package when it arrives.
packaging_format_ and_compression <string>	Requested format and compression for ordered Product(s). See clarification of format types in section 3, Packaging Specifications, of this Annex: “TARUNC” | “FILESUNC” | “TARZIP” | “FILESZIP” | “TARGZIP” | “FILESGZIP” | “TARCOMPRESS” | “FILESCOMPRESS”

Table D-1-3 order.prod_list <ProductDetailsList>

Element/Type	Desc/Value(s)
mTypes <MediaTypeList>	A list of structures of type MediaType as defined in GIAS. Only relavent for a DestinationType type of PHYSICAL. See Table D-1-4 for details. Allows media type to be specified on a per product basis. Semantics for a single <MediaTypeList> is contained within the sequence.
benums <UCO::NameList>	Not Used. (List of valid BE numbers produced by the client.)
aSpec <AlterationSpec>	Desired format and compression (where applicable) for Product. See Table D-1-5 for details.
aProduct <UID::Product>	Product object reference. Note: UID::Product when returned always represents the entire product, whether it is a single image product or a multi-image product.
info_system_name <string>	Not Used

Table D-1-4 order.prod_list.mTypes <MediaTypeList>

Element/Type	Desc/Value(s)
media_type <string>	Media Type = "4MM"| -4 Millimeter Tape "8MM" | -8 Millimeter Tape "CCT" | -Computer Compatible Tape "CD_ROM" | -Compact Disc – Read Only Memory "FILM" | -Film "FLOPPY" | -Floppy Disk "H/C" | -Hard Copy "D2C" | -Large Volume 150 GB Cartridge "DVD" | -Digital Versitile Disk "ID1_SMALL" | -19mm small cassette, STANAG 4283, MIL-STD-2179B "ID1_MEDIUM" | -19mm medium cassette, STANAG 4283, MIL-STD-2179B "ID1_LARGE" | -19mm large cassette, STANAG 4283, MIL-STD-2179B "Hi8_DIGITAL" | -Digital Tape, STANAG 4283, Mil-STD-2179(AS) "DCRSI" | - Digital 1 inch Tape , 240 Mb/s
Quantity <unsigned short>	Number of copies.

Table D-1-5 order.prod_list.aSpec <AlterationSpec>

Element/Type	Desc/Value(s)
pf <ProductFormat>	Format of product: "NSIF1.0"
ps <ProductSpec>	Contains one ImageSpec data type. See Table D-1-6. Cataloging assumes single image products.
sub_section <GeoRegion>	Not used. (This information is to be specified in the ImageSpec. See Table D-1-6)
geo_region_type <GeoRegionType>	Not used. (This information is to be specified in the ImageSpec. See Table D-1-6)

Table D-1-6 order.prod_list.aSpec.ps <ImageSpec>

Element/Type	Desc/Value(s)
imgform <ImageFormat> typedef string	For NSIF images, recommend using the NSIF image subheader IMODE values (plus "A" for "as is") to indicate format of uncompressed images. Valid values are: "A" | -As is. "B" | -Band Interleaved by Block, for single band images "P" | -Pixel Interleaved by Pixel, for multiple band images "R" | -Band Interleaved by Row, for multiple band images "S" -Band Sequential, for multiple band images
imageid <ImageUniqueIdentifier> typedef string	Alphanumeric identification code associated with an image.
comp <Compression> typedef string	IMAGE-COMPRESSION-Codes: "NC" | -Not Compressed "C1" | -Bilevel "C3" | -JPEG "C4" | -Vector Quantization "C5" | -Lossless JPEG "C6" | -future multi-component compression algorithm "C7" | -future complex SAR "C8" | -future JPEG2000 (tbd) "I1" -Downsampled JPEG See NSIF 1.0, Annex C, paragraph 17&18 for details.
bpp <BitsPerPixel> typedef short	The maximum number of significant bits for the value in each band of each pixel in which the dataset can be delivered. bpp=0 represents no change
algo <Algorithm> typedef string	Not used. (Insert a null string.)
rrds <RsetList> sequence < short >	The degree of reduction for which an image can be manipulated for delivery. Valid values are 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 An empty RsetList means include all available rsets
sub_section <GeoRegion>	Defines geographic subset or "chip" [optional] As defined in GIAS.
geo_region_type <GeoRegionType>	Defines five different types of geo-regions: two line sample types, one lat_lon in decimal degrees and two special cases as defined in GIAS. For the LAT_LON type, the GeoRegion is always in the WGS-84 coordinate system.
encoding <SupportDataEncoding>	ASCII | EBCDIC

Table D-1-7 order.del_list <DeliveryDetailsList>

Element/Type	Desc/Value(s)
dests <Destination>	To be one of the following types as defined in GIAS: For DestinationType FTP: UCO::FileLocation For DestinationType EMAIL: UCO::EmailAddress For DestinationType PHYSICAL: PhysicalDelivery Semantics for a single <Destination> contained within the sequence.
receiver <string>	UserID of person or system receiving order.
shipmentMode <string>	Used to specify the courier service on physical deliveries.

Table D-1-8 Related File Types

Type	Description
THUMBNAIL	A representative JPEG, PNG or GIF image ≤ 128 x 128 pixels.
OVERVIEW	A low resolution, STANAG 4545 formatted image, ≤1024x1024 pixels.
FOOTPRINT	TBD
ADATP3	An ADATP3 formatted text message (These can include Reconnaissance Exploitation or Following Report)

2. **Image and Ground Coordinates**. The meanings of values of "x" and "y" in the UCO::Coordinate2d data type (and in the UCO::Coordinate3d data type) are not currently specified in the UCOS or GIAS documents. Similarly, the meanings of the "upper left" and "lower right" corners of a rectangle are not currently specified in those documents. The following sections provide definitions when they are used for image coordinates and for ground coordinates.

a. Image Coordinates.

(1) When image coordinates are being used, the values of "x" and "y" in the data type UCO::Coordinate2d are interpreted to mean:

(a) The terms used below for the two image pixel indices are "row" and "column" (as used in the NSIF and not dependent on the image sensor technology).

(b) The reference point for image coordinates is the corner of the (full, chippable, or partial) image where the row and column pixel indices are (0, 0). Row and column pixel indices of (0.00, 0.00) apply to the outside corner of this corner pixel (not to the center of this pixel).

(c) The pixel column index is used for the "x" image coordinate value, and the pixel row index is used for the "y" coordinate.

(d) The row and column pixel indices (or "y" and "x") are in pixel spacing (or pixel) units in the referenced (full resolution, chippable, or reduced resolution) image. (These indices are not given in tiles or FAF blocks.)

(e) Values for the row and column pixel indices (or "y" and "x") can be given in either IDL:long or IDL:double data formats.

(f) When GIAS::GeoRegion data is used in a GIAS::AlterationSpec with a GIAS::GeoRegionType value of LINE_SAMPLE_CHIP, all the pixels in the full resolution image are extracted that fall within the pixels specified in the Chippable Image.

(2) When image coordinates are being used, the “upper left” corner of a UCO::Rectangle is the corner where the row and column pixel indices have their numerically smallest values (closest to 0, 0). The “lower right” corner is the diagonally opposite corner, where the row and column pixel indices each take on their maximum values. When integer valued row and column numbers are used for the “lower right” corner, the referenced pixel is included in the rectangle retrieved. When floating point valued row and column numbers are used for the “lower right” corner, those numbers are first rounded to the closest integer, before any conversion to positions in the full resolution image. The pixel referenced by the rounded values is included in the rectangle retrieved.

b. Ground Coordinates.

(1) When ground coordinates are being used, the values of “x” and “y” in the UCO::Coordinate2d data type and in all similar data (including a BQS query and the UCO::Coordinate3d data type) are interpreted to mean:

(a) All latitude and longitude values are in the WGS-84 ground coordinate reference system (whenever the related Geographic_datum property has the value “WGS84”, or defaults to that value).

(b) The longitude is used for the “x” ground coordinate value, and the latitude is used for the “y” value.

(c) Longitude values can range from -180 degrees to +180 degrees. (TBR) Longitude values are positive East of the Greenwich prime meridian, and are negative West of the Greenwich meridian.

(d) Latitude values can range from -90 degrees to +90 degrees. Latitude values are positive North of the equator, and are negative South of the equator.

(e) The longitude and latitude (or “y” and “x”) are in degrees units when recorded as single numbers (not radians or arc-seconds). Note that in the BQS, longitude and latitude can be alternately specified in degrees, minutes, and seconds.

(f) Values for longitude and latitude (or “y” and “x”) can be given in either IDL:long or IDL:double data formats.

(2) When ground coordinates are being used, the “upper left” corner of a UCO::Rectangle and similar data types is interpreted to mean the Northwest corner of the desired rectangle. The “lower right” corner is the diagonally opposite corner, or Southeast corner. (Note that a rectangle cannot be specified to surround the North or South pole.)

3. **Packaging Specifications**. The following describes more specifically the list of packaging specifications identified in Table D-1-2.

a. TARUNC - TAR UNCompressed consists of multiple files into one file with a .tar extension.

b. FILESUNC - Files UNCompressed is returned as a filelist, no bundling required.

c. TARZIP - TAR ZIPped consists of multiple files into one file with a .tar extension, then that file will be zipped into another file with a .zip extension(.tar.zip).

d. FILESZIP - FILES ZIPped consists of multiple files zipped into one archive and shipped with a .zip extension.

e. TARGZIP - consists of multiple files into one tar file with a .tar extension, and then the tar file is gzipped. The resulting extension is .tar.gz.

f. FILESGZIP - consists of multiple files which are individually into multiple files. Each will get a .gz extension.

g. TARCOMPRESS - Multiple files are tarred into one single file with a .tar extension, and the result is compressed with the standard Unix utility. The resulting extension is a .tar.Z.

h. FILESCOMPRESS - Multiple files are compressed individually, each with a .Z extension. Multiple files to multiple files.

4. **UID::Product usage guidance**. This section provides additional information to that contained in the UCOS to assist developers in the interpretation of the Product interfaces intended usage.

a. The attribute name for Product references returned in query results data structures shall be PRODUCT.

b. The UID::Product interface serves as a reference handle for a data set or product. It applies to a data set or product in a specific IPL and is valid only at that IPL. It is not intended to provide a globally unique identifier and attempting to use a Product reference generated by a different IPL will result in an UnknownProduct exception.

c. There are a number of mechanisms through which to associate a Product reference with a data set or product. The basic principle is to attach a product specific identifier to the Product object using an appropriate ORB method for specifying object names or markers. The solution will be implementation dependent and environment specific. This does not compromise interoperability for the reasons stated above.

d. There may be a very large number of Product objects required in a system at any one time. The advantage of the approach outlined above is that the Product objects are very lightweight. In future versions of this standard the Product interface may evolve to provide a more capable and library-neutral Product object. Since this will necessarily be a larger object it will only become possible by utilizing emerging CORBA mechanisms such as object-by-value to achieve the required scalability and performance.

5. **Complete_DAG_results structure.** Because image libraries may contain more complex metadata relationships in addition to the core NSILI data model, a consistent DAG format is required to maintain the entity-attribute relationships between the metadata that describe a product. This format is shown in Figure D-1-1.

ANNEX E TO
STANAG 4559
(Edition 1)

ANNEX E

NSIL INTERFACE MINIMUM DATA MODEL

ANNEX E TO
STANAG 4559
(Edition 1)

ANNEX E. NSIL INTERFACE MINIMUM DATA MODEL

1. **INTRODUCTION**.

a. The purpose of this Annex is to identify and describe metadata attributes that must be contained in the NSIL_CORE data view. Collectively these attributes describe the entity NSIL Interface Image Product. Attributes included in the NSIL_CORE are the those required by users of the NSIL interface to:

(1) Perform queries against library holdings

(2) Uniquely identify library holdings and recognize different products based on the same images

(3) Assure that library holdings are protected and distributed in an appropriate manner depending on product classification, associated security compartments, and additional security control and/or handling instructions.

(4) Retrieve products via the Direct Access method described in Annex C, section 2.f.(4). where this capability is provided by the IPL.

b. The NSIL_CORE data view includes only mandatory metadata as defined in Annex C, Paragraph 3.f.(2). Additional data views, employing optional metadata, may be provided by individual IPLs. These additional views will be revealed through the discovery mechanisms of the DataModelManager interface and will not be addressed in this Annex.

2. **DATA MODEL.**

NSIL Interface IMAGE PRODUCT

File Title (FTITLE) Originating Station Identifier (OSTAID) File Date and Time (FDT)
Complexity Level (CLEVEL) Country Code (CNTRYCODE) Direct Access Identifier (DAID) Direct Access Related File Identifier (DARFI) File Length (FL) Image Category (ICAT) Image Comments (ICOM) Image Coordinate System (ICORDS) Image Date and Time (IDATIM) Image Geographic Location (IGEOLO) Image Identifier 1 (IID1) Image Identifier 2 (IID2) Image Source (ISORCE) Mission ID (MISNID) Product Security Classification (PSCLAS) Product Security Classification System (PSCLSY) Product Codewords (PSCODE) Product Control and Handling (PSCTLH Product Releasing Instructions (PSREL) Product Classification Text (PSCLTX) Target Identifier (TGTID)

3. **DATA DICTIONARY**. This data dictionary provides field names, item descriptions, formats, and field-filling instructions for attributes identified in the NSIL Interface Data Model. The dictionary is organised alphabetically by field name. Each attribute's Data Type is shown as a UCOS or CORBA data type. Maximum Field length, domain, units, and descriptions are as in STANAG 4545. Entity.Attributes are case insensitive. Field values are case insensitive, and unless otherwise noted, fields are variable length.

Complexity Level (CLEVEL)

Data Type:	CORBA:Short
Max Field Length:	
Query Type:	Simple
Requirement Mode:	Mandatory
Domain:	See Description
Units:	

Description: This field shall contain the Complexity Level required to interpret fully all components of the file. Valid entries are integer assigned in accordance with complexity requirements established in Annex E of STANAG 4545. If the CLEVEL of a product is unknown or not available this field shall be set to 00 (zero zero).

Country Code (CNTRYCODE)

Data Type: CORBA:String
Max Field Length: 2
Query Type: Text
Requirement Mode: Mandatory
Domain: See Description
Units:
Description: This field shall contain the two character country code as specified in FIPS Pub 10-4 of the country where the primary target in the image is located.

Direct Access Identifier (DAID)

Data Type: CORBA:String
Max Field Length: 256
Query Type: None (DAID is a non-queryable attribute)
Requirement Mode: Mandatory
Domain: See Description
Units:
Description: This field shall contain the address and access protocol by which a user can directly access a product without the user placing an order for delivery. A valid Universal Resource Locator (URL) is an acceptable value for this field.

Direct Access Related File Indicator (DARFI)

Data Type: UCO::NameNameList
Max Field Length: unspecified
Query Type: None (DARFI is a non-queryable attribute)
Requirement Mode: Mandatory
Domain: See Description
Units:
Description: This field shall contain a list of the available related file types with URLs by which those files may be directly accessed. Related file types and associated file formats are described in Table D-1-8 Related File Types. The structure shall be NameName.name1 is equal to a Related File Type Name and NameName.name2 is the URL.

File Date and Time (FDT)

Data Type: UCOS:AbsTime
Max Field Length:
Query Type: Simple
Requirement Mode: Mandatory:
Domain: See Description
Units:
Description: This field shall contain the date and Universal Time Code (UTC) (Zero Meridian (ZULU)) time of the file's origination. The structure AbsTime is composed of a Date structure and a Time structure. The Date structure consists of an unsigned short in year using the full (four digit) specification of the year, an unsigned short in month, and an unsigned short in day. The Time structure consists of an unsigned short in hour, an unsigned short in minute, and a float in second. The Time structure represents time in Military time (i.e. 24-hour clock). If the FDT is not known or is not available this field shall be filled with all 9s.

File Length (FL)

Data Type: UCOS:FileSize
Max Field Length:
Query Type: Simple
Requirement Mode: Mandatory
Domain: See Description
Units:
Description: This field shall contain the length in bytes of the entire file including all Headers, Subheaders, and data. If the file length is not known or is not available the file length of 0 (zero) shall be used.

File Title (FTITLE)

Data Type: CORBA:String
Max Field Length: 80
Query Type: Text
Requirement Mode: Mandatory
Domain: See Description
Units:
Description: This field shall contain the title of the file.

Image Category (ICAT)

Data Type: CORBA:String
Max Field Length: 8
Query Type: Text
Requirement Mode: Mandatory
Domain: See Description
Units:

Description: This field shall contain a valid indicator of the specific category of image, raster, or grid data. The specific category of an image reveals its intended use or the nature of its collector. Valid categories include VIS for visible imagery, SL for side-looking radar, TI for thermal infrared, FL for forward looking infrared, RD for radar, EO for electro-optical, OP for optical, HR for high resolution radar, HS for hyperspectral, CP for colour frame photography, BP for black/white frame photography, SAR for synthetic aperture radar, SARIQ for SAR radio hologram, IR for infrared, MS for multispectral, FP for fingerprints, MRI for magnetic resonance imagery, XRAY for x-rays, CAT for CAT scans, VD for video, BARO for barometric pressure, CURRENT for water current, DEPTH for water depth, and WIND for air wind charts. Valid categories for geographic products or geo-reference support data are MAP for raster maps, PAT for colour patch, LEG for legends, DTEM for elevation models, MATR for other types of matrix data, and LOCG for location grids.

Image Comments (ICOM)

Data Type: CORBA:String
Max Field Length: 720
Query Type: Text
Requirement Mode: Mandatory
Domain: See Description
Units:
Description: This field, when present, shall contain free-form BCS-A text. It is intended for use as a single comment block and should be used that way. This field shall contain the free text Image Comments contained in the Image Subheader. If the Image Comment is classified, it shall be preceded by the classification, including Codeword(s).

Image Coordinate System (ICORDS)

Data Type: CORBA:String
Max Field Length: 1
Query Type: Text
Requirement Mode: Mandatory
Domain: See Description
Units:

Description: This field shall contain a valid code indicating the type of coordinate system used within the product metadata. The valid values for this field are: U for UTM expressed in Military Grid Reference System (MGRS) form, N for UTM (Northern hemisphere), S for UTM (Southern hemisphere), G for Geographic and D for Decimal Degrees. (Choice between N and S is based on hemisphere of northernmost point. The default Geodetic reference system is WGS84.

Image Date and Time (IDATIM)

Data Type: UCOS:AbsTime
Max Field Length:
Query Type: Simple
Requirement Mode: Mandatory
Domain: See Description
Units:
Description: This field shall contain the date and Universal Time Code (UTC) (Zero Meridian (ZULU)) time of image acquisition. The structure AbsTime is composed of a Date structure and a Time Structure. The Date structure consists of an unsigned short in year using the full (four digit) specification of the year, an unsigned short in month, and an unsigned short in day. The Time structure consists of an unsigned short in hour, an unsigned short in minute, and a float in second. The Time structure represents time in Military time (i.e. 24-hour clock). If the IDATIM is not known or is not available this field shall be filled with all 9s.

Image Geographic Location (IGEOLO)

Data Type: UCOS:Polygon
Max Field Length:
Query Type: Geo
Requirement Mode: Mandatory
Domain: See Description
Units:
Description: This field shall contain an approximate geographic location which is not intended for analytical purposes (e.g., targeting, mensuration, distance calculation); it is intended to support general user appreciation for the image location (e.g., cataloging). The locations of the four corners of the (significant) image data shall be represented using the WGS-84 ground coordinate reference system and shall be given in image coordinate order: (0,0), (0,MaxCol), (MaxRow,MaxCol), (MaxRow,0) using the WGS84 ground coordinate reference system.

Valid corner locations shall be expressed as latitude and longitude in decimal degrees using UCO::Coordinate2d data type.

Note: Provide the value only to the decimal places (precision) warranted by the sources and methods used to determine the location. If the geographic location of the image is not known or is not available the IGEOLO field shall be filled with 9s.

Image Identifier 1 (IID1)

Data Type: CORBA:String
Max Field Length: 10
Query Type: Text
Requirement Mode: Mandatory
Domain: See Description
Units:
Description: This field shall contain a valid alphanumeric identification code associated with the image. The valid codes are determined by the application.

Image Identifier 2 (IID2)

Data Type: CORBA:String
Max Field Length: 80
Query Type: Text
Requirement Mode: Mandatory
Domain: See Description
Units
Description: This field shall contain the title of the image. (The Image Title should include the Target Name listed in the MIDB).

Image Source (ISORCE)

Data Type: CORBA:String
Max Field Length: 42
Query Type: Text
Requirement Mode: Mandatory
Domain: See Description
Units:
Description: This field shall contain a description of the Source of the image. If the Source of the data is classified, then the description shall be preceded by the classification, including Codeword(s) contained in Table C-1-4 of STANAG 4545.

Mission ID (MISNID)

Data Type: CORBA:String
Max Field Length: 8
Query Type: Text
Requirement Mode: Mandatory
Domain: See Description
Units:
Description: This field shall contain the number assigned to the recce mission.

Originating Station Identifier (OSTAID)

Data Type: CORBA:String
Max Field Length: 10
Query Type: Text
Requirement Mode: Mandatory
Domain: See Description
Units:
Description: This field shall contain the identification code of the originating organisation.

Product Security Classification (PSCLAS)

Data Type: CORBA:String
Max Field Length: 1
Query Type: Text
Requirement Mode: Mandatory
Domain: See Description
Units:
Description: This field shall contain a valid value representing the classification level of the entire Product. Valid values are T for Top Secret, S for Secret, C for Confidential, R for Restricted, or U for Unclassified.

Product Security Classification System (PSCLSY)

Data Type: CORBA:String
Max Field Length: 2
Query Type: Text
Requirement Mode: Mandatory
Domain: See Description
Units:
Description: This field shall contain valid values indicating the national or multinational security system used to classify the Product. Country Codes per FIPS PUB 10-4 are used to indicate national security systems. If this field is empty, it shall imply that no Security Classification System applies to the Product.

Product Classification Text (PSCLTX)

Data Type: CORBA:String
Max Field Length: 43
Query Type: Text
Requirement Mode: Mandatory
Domain: See Description
Units:
Description: This field shall be used to provide additional information about the Product Classification to include identification of a declassification or downgrading event if applicable. It may also be used to identify multiple classification sources and/or any other special handling rules. Values are user-defined free text. If this field is empty, it shall imply that additional information about the Product Classification does not apply.

Product Codewords (PSCODE)

Data Type: CORBA:String
Max Field Length: 11
Query Type: Text
Requirement Mode: Mandatory
Domain: See Description
Units:
Description: This field shall contain a valid indicator of the security compartments associated with the Product. Values include one or more of the digraphs found in Tables C-1-4,and C-1-4(A) of STANAG 4545, which is based on NATO C-M(55) 15 (Final) Volume I. Multiple entries shall be separated by a single BCS Space (code 0x20). The selection of a relevant set of Codewords is application specific. If this field is empty, it shall imply that no Codewords apply to the Product.

Product Control and Handling (PSCTLH)

Data Type: CORBA:String
Max Field Length: 2
Query Type Text
Requirement Mode: Mandatory
Domain: See Description
Units:
Description: This field shall contain valid additional security Control and/or Handling instructions (caveats) associated with the Product. Values include digraphs found in Table C-1-4, which is based on NATO C-M(55) 15 (Final) Volume I, and Table C-1-4(A). The digraph may indicate single or multiple caveats. The selection of a relevant caveat(s) is application specific. If this field is empty, it shall imply that no additional Product Control and Handling instructions apply to the Product.

Product Releasing Instructions (PSREL)

Data Type:	CORBA:String
Max Field Length:	20
Query Type:	Text
Requirement Mode:	Mandatory
Domain:	See Description
Units:	
Description:	This field shall contain a valid list of countries outside of NATO to which the Product is valid for release. Typical values include one or more country codes as found in FIPS PUB 10-4 separated by a single BCS Space (code 0x20). If this field is empty, it shall imply that no Product releasing Instructions apply.

Target Identifier (TGTID)

Data Type:	CORBA:String
Max Field Length:	15
Query Type:	Text
Requirement Mode:	Mandatory
Domain:	See Description
Units:	
Description:	This field shall contain the identification of the primary target in the image, formatted as BBBBBBBBBBFFFFF, consisting of ten characters of Basic Encyclopaedia (BE), followed by five characters of functional category code as defined in the MIDB.