



Committee on Earth Observation Satellites
Working Group on Information Systems and Services

Interoperable Catalogue System (ICS) Guide Design and Protocol Specification

CEOS
Working Group on Information Systems and
Services
Protocol Task Team

Doc. Ref.: CEOS/WGISS/PTT/GDPS
Date: 28 July, 1998
Issue: Version 1.1

AUTHORITY

Issue:	Version 1.1
Date:	28 July, 1998
Location:	Maryland

This document has been submitted for approval for publication to the Protocol Task Team of the Committee on Earth Observation Satellites (CEOS).

Authors:

Larry P. Rosen (lrosen@harp.gsfc.nasa.gov)
Joe McMahon
Eryq (Erik Dorfman)
Raytheon STX Corporation

Contributions:

Janet Hylton
Jan Dreisbach

This document is published and maintained by:
Gerhard Triebnig (CEOS PTT Chair)
European Space Agency/ESRIN
Via Galileo Galilei
00044 Frascati
Italy
E-mail: gerhard.triebning@esrin.esa.it

TABLE OF CONTENTS

1. INTRODUCTION	1
1.1 GDPS PURPOSE AND SCOPE.....	1
1.1.1 G1 - ICS Client Search and Retrieval of ICS Guide Documents.....	1
1.1.2 G2 - General Web Search and Retrieval of ICS Guide Documents.....	1
1.1.3 G3 - Minimize Burden	1
1.1.4 G4 - A Distributed System.....	1
1.2 ORGANIZATION OF THE GDPS.....	2
1.3 ICS/PTT DEVELOPMENT PROCESS	2
1.4 GLOSSARY	2
1.4.1 Acronyms	2
1.4.2 Definitions	3
1.5 REFERENCES	4
2. ARCHITECTURAL FRAMEWORK OF THE ICS GUIDE SYSTEM	5
2.1 ICS GUIDE SYSTEM COMPONENTS - HIGH LEVEL	5
2.2 GUIDE SYSTEM COMPONENTS - LOW LEVEL	6
2.3 GUIDE DATA	6
3. ICS GUIDE PROTOCOL AND DESIGN DETAILS.....	8
3.1 ICS GUIDE PROTOCOL (IGP).....	8
3.1.1 Guide Attributes	8
3.1.2 ICS Guide Protocol Messages.....	11
3.2 FILE FORMAT SPECIFICATIONS	14
3.2.1 Attribute Mapping File	14
3.2.2 Collections Mapping File	14
3.2.3 Guide Metadata Database	15
3.2.4 Attribute Defaults File	16
3.2.5 icsdoc.html Site Listing Page.....	16
3.2.6 ICS Free-Text Index	17
3.2.7 ICS Fielded Index.....	17
3.3 ICS GUIDE SYSTEM COMPONENTS	17
3.3.1 ICS Guide Server.....	17
3.3.1.1 HTTP Daemon.....	17
3.3.1.2 icsearch Software.....	18
3.3.1.3 Files	19
3.3.2 Guide Translator.....	19
3.3.2.1 icsdoc Software.....	19
3.3.2.2 ics_collection_update	21
3.3.2.3 ics_metadata_update	21
3.3.2.4 Files	22
3.3.3 ICS Guide Indexer.....	22
3.3.3.1 icsindex Batch Mode.....	22
3.3.3.2 icsindex Add Mode	22
3.3.3.3 HTTP Daemon.....	24
3.3.3.4 Files	24
3.3.3.5 Summary of Actions.....	24
3.4 RELATED COMPONENTS.....	25
3.4.1 Collection Management Tool.....	25
3.4.2 ICS Client	25
3.4.3 IGP Client.....	25
3.4.4 Guide Documents	25

4. SCENARIOS	26
4.1 ICS GUIDE SEARCH WITH GENERAL WEB SEARCH ENGINE	26
4.2 ICS GUIDE RETRIEVAL FROM GENERAL WEB SEARCH ENGINE RESULT	27
4.3 ICS CLIENT FREE-TEXT GUIDE SEARCH	28
4.4 ICS FIELDDED GUIDE SEARCH	29
4.5 ICS GUIDE RETRIEVAL FROM ICS CLIENT	31
4.6 LINKING TO COLLECTIONS FROM GUIDES	32
4.6.1 <i>Linking to Collections from Guides using ICS Client</i>	32
4.6.2 <i>Linking to Collections from Guides using General Browser</i>	33
4.7 LINKING TO GUIDES FROM DATA COLLECTIONS.....	33
4.8 ADDING GUIDES TO THE SYSTEM.....	33
APPENDIX A. GENERATING VIRTUAL DOCUMENTS.....	34
APPENDIX B. ADDING NEW GUIDE DOCUMENTS TO SYSTEM.....	39
APPENDIX C. INDEXING GUIDE DOCUMENTS	42

LIST OF FIGURES

FIGURE 1. ARCHITECTURAL FRAMEWORK OF ICS GUIDE SYSTEM	5
FIGURE 2. ADDING A NEW GUIDE DOCUMENT TO THE SYSTEM.....	23
FIGURE 3. SEARCHING FOR ICS GUIDE DOCUMENTS	26
FIGURE 4. RETRIEVING ICS GUIDE DOCUMENTS FROM ALTA VISTA SEARCH RESULTS	27
FIGURE 5. ICS CLIENT FREE-TEXT GUIDE SEARCH	29
FIGURE 6. ICS CLIENT DOING FIELDDED GUIDE SEARCH	30
FIGURE 7. RETRIEVING ICS GUIDE DOCUMENTS FROM ICS SEARCH RESULTS	32
FIGURE 8. VIRTUAL DOCUMENT	35
FIGURE 9. ALL STEPS FOR ADDING A NEW GUIDE TO THE SYSTEM.....	39
FIGURE 10. GENERAL WEB INDEXING AND AGENCY SITES	44

LIST OF TABLES

TABLE 1. ACRONYMS.....	2
TABLE 2. DEFINITIONS.....	3
TABLE 3A. SUMMARY OF THE ICS GUIDE SYSTEM COMPONENTS	6
TABLE 3B. ICS GUIDE SYSTEM RELATED COMPONENTS.....	6
TABLE 4. GUIDE ATTRIBUTES.....	9
TABLE 5A. DEFINITIONS OF MESSAGE ELEMENTS.....	11
TABLE 5B. MESSAGES IN THE ICS GUIDE PROTOCOL.....	12
TABLE 6. ICS GUIDE SERVER ACTIONS FOR MESSAGES.....	18
TABLE 7. GUIDE TRANSLATOR ACTIONS FOR MESSAGES.....	19
TABLE 8. ICSDOC ACTIONS FOR MESSAGES	21
TABLE 9. ICS GUIDE INDEXER ACTIONS FOR MESSAGES	24

DOCUMENT STATUS SHEET

Version	Date	Comments
0.1	18 November 1998	Original version entitled "Proposal for a CIP Guide Implementation" was presented at Core Engineers Meeting in Landover, Maryland, USA.
1.0	22 January 1998	Renamed "ICS Guide Design and Protocol Specification". Document reworked to be official design and specification of ICS Guide System. Major restructuring. Document to be presented at PTT meeting in Salzburg, Austria, 20 February, 1998.
1.1	28 July 1998	Minor corrections and added references in preparation for RID review.

DOCUMENT CHANGE RECORD

Change No.	Date	Changes in Version 1.0 from 0.1
1	22 January 1998	Title changed from "Proposal for a CIP Guide Implementation". Interconnection within ICS discussed. Protocols summarized. Many details reworked. Much more discussion of protocol messages and components. Document put into CEOS format.
2	28 July 1998	Minor corrections for RID review.

1. INTRODUCTION

1.1 GDPS Purpose and Scope

This paper describes the high-level design for an ICS (Interoperable Catalogue System) Guide System [R1: 3.4]. Much of the metadata for EO data collections is not easily stored in a structured form. This information is stored in documents called Guides. Since guide documents provide information that is required for the understanding of some EO data collections, they must easily be accessed via ICS mechanisms that provide search and retrieval of catalogs. Guide documents also provide human readable descriptions of EO data collections and are often used by new EO data users as a discovery mechanism to identify collections of interest. It is the goal of the ICS to make this discovery mechanism as simple and widely available as possible to extend the uses of EO data to communities which have not traditionally used EO data. This goal has resulted in the definition of a guide system in ICS which uses ICS Guide Protocol (IGP) based on HTTP and enables general purpose Internet Search/Discovery Engines such as Alta Vista to locate EO guide documents either by free text or attribute value searches.

This ICS Guide System is not based on Z39.50 and is not a mandatory capability of an ICS node. However there is a strong linkage between the CIP client/retrieval manager and the HTTP based client and indexing method for guide documents. The ingest of guides and collections into the ICS is coordinated by the Collection Management Tool (CMT) to assist in maintaining the consistency of the collection descriptor and the HTTP index that enable search and access of guide documents. The system is designed to integrate CIP and HTML document spaces, allowing direct access to the guides via distributed HTML gateways, while minimizing the direct impact on guide document authors.

This paper defines the ICS components that comprise the ICS Guide System, the sub-components of the system, and the protocol of messages used in this system. We discuss the interaction of the ICS Guide System with the CIP space, and provide example scenarios. The concept of a virtual document provides the basis of the system. How it is generated and used both inside and outside of ICS is discussed. Through this design, general-purpose Web search engines can index and access ICS virtual documents. The components provide ICS internal indexing which allows fielded and free-text searches within ICS. The process of adding a new document to the system is detailed.

The following are the goals of the ICS Guide System and are considered requirements:

1.1.1 G1 - ICS Client Search and Retrieval of ICS Guide Documents

The ICS Guide System shall enable users of ICS Client software to search for and retrieve ICS guide documents by specifying free-text and/or Guide Attribute/Keyword pairs. It is desirable to find and return only ICS guide documents with this search.

1.1.2 G2 - General Web Search and Retrieval of ICS Guide Documents

To enable more people to enter the ICS environment ("discovery") and make the ICS guide documents more broadly accessible, it is suggested that the system allow users of common WWW free-text search engines (Alta Vista, Lycos, Yahoo, etc.) to search for and retrieve ICS guide documents. The users of these engines can find ICS guide documents by specifying free-text and/or Guide Attribute/Keyword pairs. It is expected that, as with all such searches using general indexing engines, documents other than ICS guide documents may be returned unless the user takes particular steps to filter their results. The guide documents shall be accessed by URL (or URN).

1.1.3 G3 - Minimize Burden

The ICS Guide System must not place an undue document maintenance burden on the document authors, the maintainers of the agency sites, or any centralized ICS authority. In particular, this prohibits the system from requiring the manual alteration of existing guide documents.

1.1.4 G4 - A Distributed System

The ICS Guide System will be distributed among the agency sites. There will be no centralized part of the system.

1.2 Organization of the GDPS

The GDPS is organized in the following manner. First we have presented the goals of the system. These can be considered requirements. Next, the context of the system is displayed in a diagram that shows how the guide document system interrelates with other ICS components. This is followed by a summary of the components of the ICS Guide System. The ICS Guide Protocol, the specification of Guide Attributes and system messages, is presented. Detailed file formats are given for the files used in the system. Remaining system components are described in detail. Related components are discussed. Scenarios are presented with figures, showing how the system accomplishes its tasks. The fine details of some of the system's processes are presented in the appendices. Appendix A details the process of generating virtual documents, the key part of the system. Appendix B details the process of adding guide documents to the system. Appendix C details the process of indexing guide documents in the system.

1.3 ICS/PTT Development Process

This document was developed for the CEOS Protocol Task Team (PTT). The PTT is part of the WGISS Access Subgroup within CEOS. The lead agency guiding this document's development was NASA. A complete list of organizations participating in the PTT is provided in the PTT Terms of Reference (<http://ceos.ccrs.nrcan.gc.ca/taskteam/cip.html>).

1.4 Glossary

1.4.1 Acronyms

The following acronyms are used in this document.

Table 1. Acronyms

CDB	Collection Database
CEO	Centre for Earth Observation (European Commission)
CEOS	Committee on Earth Observation Satellites
CIP	Catalogue Interoperability Protocol
CGI	Common Gateway Interface
DB	Database
ECS	EOSDIS Core System
EO	Earth Observation
EOSDIS	Earth Observing System Data and Information System (NASA)
G1 (G2 . . .)	Goal 1, Goal 2 . . . (presented in this document)
GDPS	Guide Design and Protocol Specification (This document)
GUI	Graphical User Interface
HTML	Hyper Text Mark-up Language
HTTP	Hypertext Transfer Protocol
HTTPD	Hypertext Transfer Protocol Daemon
ICS	Interoperable Catalogue System
ID	Identifier
IGP	ICS Guide Protocol
ISA	ICS Site Administrator
NASA	National Aeronautics and Space Administration (US)
PTT	Protocol Task Team (Part of CEOS WGISS-AS)
RM	Retrieval Manager
RMA	Retrieval Manager Administrator
URL	Uniform Resource Locator
V0	Version 0 of EOSDIS
URN	Uniform Resource Name
WGISS	Working Group on Information Systems and Services (Part of CEOS)
WWW / Web	World Wide Web (Web for short)

1.4.2 Definitions

This section provides a table of definitions of terms and phrases used in this document.

Table 2. Definitions

Agency Site / Site	An ICS agency site.
Browser / Web Browser	Any software used as an HTTP Client (examples: Netscape Navigator, Microsoft Internet Explorer).
CIP Client	A software element composed of a <i>Presentation Layer</i> , a <i>Local User Management Layer</i> , and an <i>Application Layer</i> . Only the <i>CIP Client Application Layer</i> is part of ICS. See CIP Specification.
Collection	A grouping of item descriptors that have commonality. A collection consists of a number of attributes that describe the collective contents of the collection; the values of these attributes can then be searched on to select items of interest to the user. Collections also have members; these are the unique identifiers of the items that are grouped by the collection rather than their collective descriptions. As collection members can be identifiers of other collections, a hierarchy of collections and product/guide members can be established, therefore permitting a flexible and powerful organization of data.
Fielded Search	Searching for guides that have the specified field and value pairs associated with them.
Free-Text Search	Searching for guides that contain specified text anywhere within them.
Guide Attribute	The name of the feature of an ICS guide document that can be used to identify guides of interest to the user. For example: "PublicationDate". See Table 4.
Guide Documents / ICS Guide Documents / Guides	Documents that provide human readable descriptions of EO data collections. They contain unstructured metadata for data collections and can provide information that is required for the understanding of some EO data collections. In this system, we also include white papers, publications, system documents, EO documents, product description documents, user manuals, periodicals, proceedings, and other relevant documents. This includes documents at EOSDIS V0, ECS, and international sites.
Guide Keywords	The valid values (or "valids") for the Guide Attributes.
Guide URL	URL of a virtual guide document. It is the GetVirtualDocument message defined in the ICS Guide Protocol Messages section 3.1.2.
HTTP Client	Software that presents an interface to a user, accepts input from the user, makes HTTP connections to HTTP Daemons (httpd), sends messages to the httpd, receives information from the connection (usually HTML pages), and displays the results to the user.
HTTP Daemon / httpd	Software, running on a computer system, that listens for HTTP connections on designated port numbers. Once a connection is made, the process spawns a new process to handle software requests made - usually executing some other software called a server, while the parent process goes back to listening at the port for other connections.
ICS Client	Component of ICS that provides user access to CIP services via a CIP Client, and access to ICS Guide System services via an IGP Client.
ICS Guide Search / Guide Search	A search for ICS guide documents using an ICS Client.
ICS Site Administrator	The human operator that performs all tasks needed to establish and maintain a <i>Retrieval Manager</i> and Guide System. In practice this is usually more than one person as the tasks are various types: scientist for collection definition, data base expert for maintaining CDB, system operator for diagnosing and correcting operational activities, etc. For convenience purposes all of these tasks are performed by the ISA. Previously the ISA was referred to as the RMA.
IGP Client	ICS component software part of ICS Client that acts as an HTTP client that sends and receives IGP messages. Displays HTML pages.

Meta Tags	Field/Value text within an HTML document that is not shown in a browser display of the page. Several formats are possible. Most widely used format is: <code><META name=FieldName, value=Value></code>
Physical Documents	On-line HTML documents as composed by authors and stored where they can be retrieved by software at an agency site.
Retrieval Manager	<p>A Retrieval Manager services (and may be installed at) each catalogue site, it is used to integrate together the local catalogue systems and provide communication between users and other catalogue site Retrieval Managers. It is anticipated that each catalogue site would have at least one Retrieval Manager and that Retrieval Manager would 'know about' or 'own' a number of collections. The data within these collections would be the responsibility of that Retrieval Manager, with external collections referenced only and managed by their respective Retrieval Managers.</p> <p>The Retrieval Managers at each catalogue site would also communicate with each other using the CIP. The Retrieval Manager would then also communicate with local catalogue servers, such as archives and inventories, within its own site to services requests received from users. Another key function of the Retrieval Manager is to route search queries to other relevant Retrieval Managers and consolidate the search results before returning them to the user.</p>
Virtual Documents	<p>HTML documents generated on-the-fly by CGI software, from the physical documents. They contain a reproduction of the entire original guide documents with two additions:</p> <ul style="list-style-type: none">• An invisible Meta tag section containing Guide Attribute/Keyword pairs identifying each guide document.• A visible link to data collections related to the guide document.

1.5 References

[R1] *Interoperable Catalogue System (ICS) System Design Document (SDD)*, CEOS/WGISS/PTT/SDD, Version 1.4, June 1998.

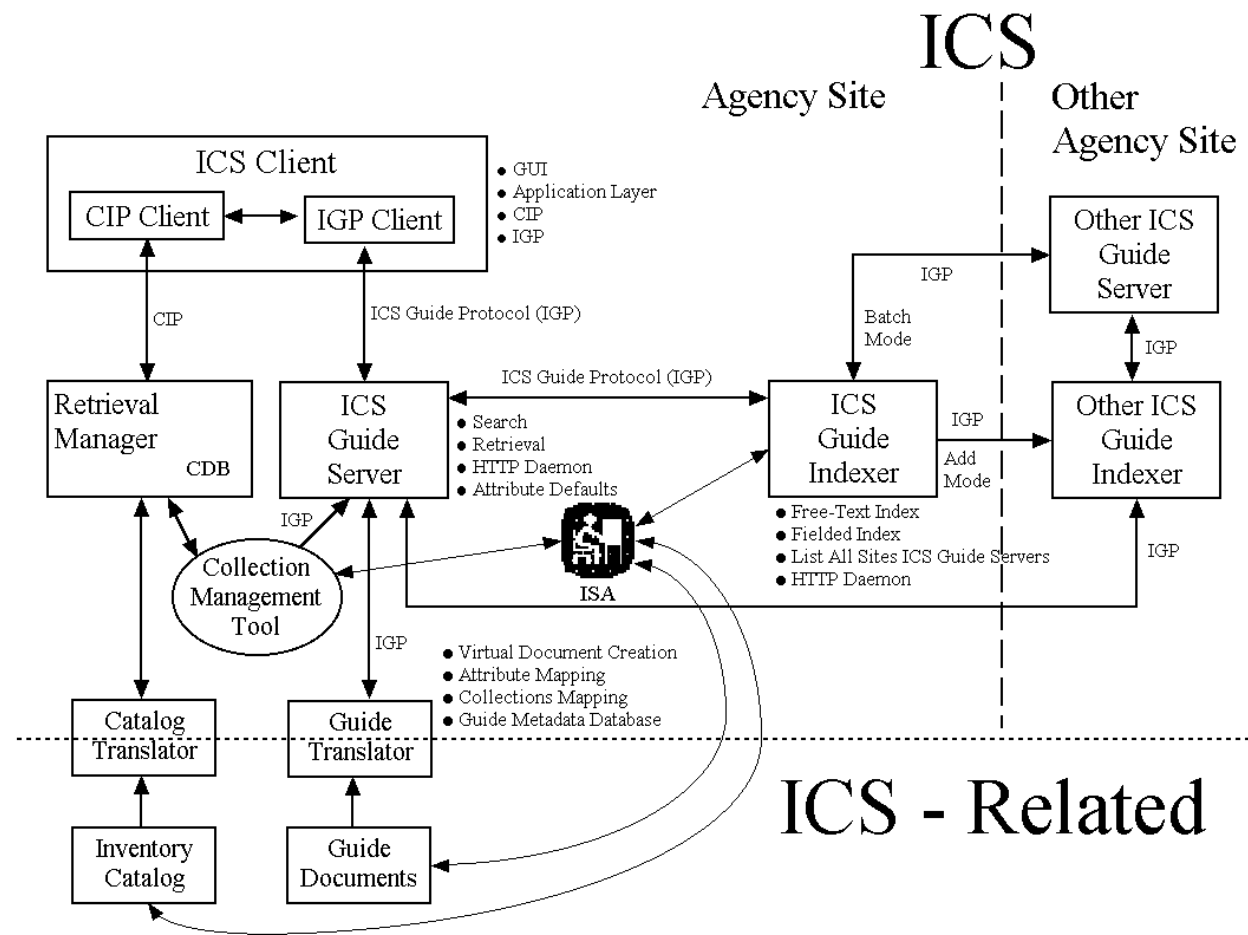
[R2] *Catalogue Interoperability Protocol (CIP) Specification - Release B*, CEOS/WGISS/PTT/CIP, Issue 2.4, June 1998.

2. ARCHITECTURAL FRAMEWORK of the ICS GUIDE SYSTEM

2.1 ICS Guide System Components - High Level

At a high level, the ICS Guide System consists of three ICS components: the ICS Guide Server, the ICS Guide Indexer, and the Guide Translator. It also requires a collection of documents (which may or may not be a database). It also requires an ICS Component, the Collection Management Tool, to maintain part of the system. The following figure shows how these components are related to CIP components and ICS related components at an agency site. (see also reference [R1]). It also shows how the ICS Guide System components relate to components at another agency site.

Figure 1. Architectural Framework of ICS Guide System



In brief, the ICS Guide Indexer creates and updates ICS indexes of guides. The Guide Translator modifies guides in ways that add value and enable the system to meet its goals. The ICS Guide Server handles HTTP connections, guide search and retrieval requests, and provides the interface to the system. The Collection Management Tool must be used to send messages, to update the Collections Mapping File used by the Guide Translator. These components are described in detail in this document.

As we go through the detailed design of the system, you may wish to refer back to this figure to see how the pieces interrelate at the ICS component level.

2.2 Guide System Components - Low Level

The ICS Guide System requires that each site have **all** of the components displayed in the following tables. The components and their interaction will be discussed in this paper. A brief mention of the functions of the components is listed in the tables. These will be explained in more detail in following sections.

Table 3a. Summary of the ICS Guide System Components

Component	Functions
ICS Guide Server:	
HTTP Daemon	Handle HTTP access to ICS Guide Server.
<i>icssearch</i>	Present searchable fields to ICS Client. Search free-text and fielded indexes
Attribute Defaults File	Default values for all mandatory Guide Attributes.
Guide Translator:	
<i>icsdoc</i>	Virtual document and directory server.
Attribute Mapping File	Map agency attributes to Guide Attributes. Map agency valids to Guide Keywords (maybe).
Collections Mapping File	Collection identifiers for each document.
Guide Metadata Database	Values for all mandatory Guide Attributes for each guide document. Any values for optional Guide Attributes.
<i>ics_collection_update</i>	Create and update Collections Mapping File.
<i>ics_metadata_update</i>	Create and update Guide Metadata Database.
ICS Guide Indexer:	
<i>icsindex</i>	Make free-text and fielded indexes of all guides at all sites. Index new guides and notify <i>icsindex</i> at other sites. Build and update Guide Metadata Database.
Free-Text Index	Index all guides by words in guides.
Fielded Index	Index all guides by Guide Attributes and Keywords.
icsdoc.html	List of ICS Guide Server URLs at all agency sites.
HTTP Daemon	Handle HTTP access to <i>icsindex</i> and icsdoc.html.

The following ICS components are required for the ICS Guide System, but are not strictly components of the guide system.

Table 3b. ICS Guide System Related Components

Component	Function (in ICS Guide System)
Collection Management Tool	Send message to update the Collections Mapping File.
ICS Client	Provide user with integrated environment for using CIP services and the ICS Guide System services.
IGP Client	Part of the ICS Client that provides the interface to ICS Guide System services.
Guide Documents	Set of guide documents at a site that can be accessed by the ICS Guide System.

2.3 Guide Data

Although the guide protocol supports the insertion and retrieval of any type of document, the most common example from the Earth Observation (EO) perspective is the Guide Document. Guides are textual documents that are developed to assist the user in understanding the EO environment. Guide documents provide additional detailed information about a specific mission, sensor, or organization. The most frequently covered guide topics include:

Data Centers - provides details about the center that holds the data.

Project/Campaign - describes projects and associated intensive field campaigns coordinated to collect data for a focused study.

Source/Platform - describes the source that held the instrument and sensors during data collection. Source is intended to communicate the data collection environment, which includes satellite, aircraft, buoys, ships, and ground station platforms. It also includes humans in the case of hand-held sensors or human observations, paper or electronic surveys in the case of questionnaires, and computers in model analysis.

Sensor/Instrument - describes the instrument and its component sensor(s) that actually collected the data. This includes the human eye as a sensor, as well as, paper or electronic questionnaires used to collect data. It also includes computer model analyses that generate data.

Dataset - describes the data set, collection procedures, algorithms, and processing data.

For this system, guide documents must be provided in HTML format. If the guides are in HTML (or plain text), the system can easily work with the files. If not, then part of the Guide Translator must convert the guide to HTML on the fly. For instance, MS-Word documents could be included in the system only if the translator had an additional "plug-in" to convert the format to plain text or HTML.

In addition to guide documents there are Metadata which serves to characterize the related document. This metadata is used by the system to identify potential guides of interest. Each guide document in ICS will be characterized in the form of document metadata. The management of documents and their related metadata are the focus of the discussions in the following sections.

3. ICS GUIDE PROTOCOL and DESIGN DETAILS

3.1 ICS Guide Protocol (IGP)

The following sections define the communication protocol for the ICS Guide System, the ICS Guide Protocol (IGP). It is presented in two parts. First are the Guide Attributes and second are the system messages. Communications in the IGP domain are accomplished using a set of messages that are based on the Hypertext Transfer Protocol (HTTP). The HTTP is an application-level protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless, object-oriented protocol which can be used for many tasks, such as name servers and distributed object management systems, through extension of its request methods. A feature of HTTP is the typing and negotiation of data representation, allowing systems to be built independently of the data being transferred. HTTP allows an open-ended set of methods that indicate the purpose of a request. It builds on the discipline of reference provided by the Uniform Resource Identifier (URI), as a location (URL) or name (URN), for indicating the resource to which a method is to be applied. Messages are passed in a format similar to that used by Internet mail as defined by the Multipurpose Internet Mail Extensions (MIME). Within the scope of the IGP Domain, IGP messages based on the HTTP protocol are passed between the IGP Client, the ICS Guide Server, and the Guide Translator in order to provide guide document searching and retrieval. IGP is also used by the ICS Guide Indexer, which creates indexes of ICS guides.

3.1.1 Guide Attributes

Throughout this document we will refer to the Guide Attributes. ICS Client users will be able to search for guide documents by specifying values for the Guide Attributes. The matching guides will have the specified values. This is called a fielded search. The user will not have to specify values for all of the Guide Attributes, but only those fitting the criteria the user wants. Since these attributes are used in the messages passed between the ICS Client and the ICS Guide Server, we include them here prior to presenting the messages of the Protocol. Fielded searches will be discussed in more detail later. Table 4 on the following pages contains the Guide Attributes to which we refer.

Table 4. Guide Attributes

Mandatory	Name	Short Meaning	Value Syntax	Str.	Meaning
X	Abstract	Abstract	STRING	bib-1 108	A brief narrative summary of the content of an item.
X	CreationDate	Creation Date	TIME	CIP 1	Date at which the data was created.
X	ItemDescriptorId	Item Descriptor Identifier	STRING	bib-1 101	Unique identifier used by the data system or data producer to distinguish a collection, product or guide from all others owned by a single Retrieval Manager. Identifier of the member of a collection, product or guide.
X	OrganisationName	Organisation Name	STRING	bib-1 101	The name of the organisation to which the contact apply.
X	AuthorName	Author Name	STRING	bib-1 101	Name of the author of the document. The last name must be given first, followed by the first name and, if applicable, the middle name.
	JobPosition	Job Position	STRING		The Name of the position that the author holds within the organization
	StreetAddress	Street Address	STRING		The street address of the author
	City	City	STRING		The city of the authors address
	State	State/Province	STRING		The state or province of the address
	PostalCode	Postal Code	STRING		The postal code of the address
	Country	Country	STRING		The country of the address
	EmailAddress	E-Mail Address	STRING		The address of the electronic mailbox of the author.
	FaxNumber	Fax Number	STRING		The telephone number of the facsimile machine or the author
	TelephoneNumber	Telephone Number	STRING		The telephone number by which individuals can speak to the author of the document
X	RevisionDate	Revision Date	TIME	CIP 1	Date and possibly time at which the data was created or the latest date and time of its modification or update.
	Purpose	Purpose	STRING		Summary of the intentions for which the document was developed.
X	Version ID	Version Identifier	STRING		Version identifier of the document
	ExternalPublicationCitation	Citation for External Publication	STRING		The recommended reference to be used when referring to the document in publications.
X	DocumentType	Document Type	ENUM	bib-1 101	Identification of the type of the guide document. For example. 'user manual', 'sensor guide', 'mission guide', etc.
X	DocumentLanguage	Document Language	STRING		Available Language or languages of the document.
X	DocumentName	Document Name	STRING		Name of the document
	PublicationPlace	Publication Place	STRING		Place of publication of the document

Mandatory	Name	Short Meaning	Value Syntax	Str.	Meaning
X	PublicationDate	Date of Publication	TIME	CIP 1	Date of formal/informal publication of the reference paper.
	DocumentFormat	Document Format	STRING		Name of the format in which the document can be delivered
	DocumentCompression	Document Compression	STRING		Identification of the algorithm used to compress the guide document.
	ItemByteSize	Item Byte Size	INTEGER		Size of the item in bytes
	SpatialKeyword	Spatial Keywords	ENUM		The geographical place names associated with the document; e.g. Atlantic Ocean
	TemporalKeyword	Temporal Keywords	ENUM		This attribute specifies a word or phrase which serves to summarize the temporal characteristics referenced in the document, including phenomenological events.; e.g. Mt Pinotubo Eruption
X	GeneralKeyword	General Keyword	STRING	bib-1 6	This element provides the capability of entering general keywords that are not found in the parameter, discipline, or spatial keywords. It could be used, for example, to specify fine resolution location words or more specific discipline-dependent words or phenomena. For example, 'Lightening'.
	UpdateFrequency	Maintenance and Update Frequency	STRING		The frequency with which changes and additions are made to the document after the initial document is completed.
	ScienceReviewDate	Science Review Date	TIME		Date and time of the latest review of the document for accuracy of scientific or technical content
	ScienceReviewStatus	Science Review Status	ENUM		Type of review that occurred on the science review date
	FutureReviewDate	Future Review Date	TIME	CIP 1	This date, suggested by the author, indicates a time at which the collection should be reviewed for technical content.
	RelatedCollectionID	Related Collection Identifier	STRING		Unique Identifier used by the system to distinguish a collection from all others and relate the collection to the document.

3.1.2 ICS Guide Protocol Messages

In this section we define the ICS Guide Protocol (IGP) for the various links and messages in the ICS Guide System. The protocol uses HTTP as a base, and then has specific formats for the various functions of the guide document system. The following table shows definitions of elements that are used in the message definitions.

In the message definition formats, the following conventions are used. In all of them, the special characters are in the definitions, not in the messages.

- Literal strings are quoted with double quotes, "".
- Optional elements are enclosed in square brackets, [].
- Elements may be followed with an asterisk, *, to designate 0 or more repetitions of the preceding element.
- Parentheses, (), groups some elements, usually so they can be repeated, indicated by the * symbol.
- Plus sign, +, indicates string concatenation.
- Curvy braces, {}, indicate an explanation of an element, ie. not necessarily defined in terms of other elements.

Format element definitions (see reference R2):

Table 5a. Definitions of Message Elements

Element	Definition
site_url	"http://" + host + [":" + guide_port] + "/"
host	{The fully qualified domain name of a network separated by ".". Fully qualified domain names take the form as described in Section 3.5 of RFC 1034 and Section 2.1 of RFC 1123: a sequence of domain labels separated by ".", each domain label starting and ending with an alphanumerical character and possibly also containing "-" characters. The rightmost domain label will never start with a digit, though, which syntactically distinguishes all domain names from the IP addresses.}
guide_port	{The port number to connect to for httpd in ICS Guide Server. Most schemes designate protocols that have a default port number. Another port number may optionally be supplied, in decimal, separated from the host by a colon. If the port is omitted, the colon is as well.}
document_path	[path] + filename
text	{Any amount of free text contained in a pair of double quotes ""}.
attribute	{Any Guide Attribute from Table 4.}
value	{Any valid value that a Guide Attribute from Table 4 can have.}
cip_url_prefix	zscheme + "://" + host + [":" + cip_port] + "/"
zscheme	"z39.50s"
cip_port	{port used for CIP Client}
cip_client	{pathname of CIP client script from cip_port directory}
collection_id	"CID" + "_" + dddddddd
dddddddd	{an ASN.1 GeneralString of exactly 8 characters which is unique within the scope of all items owned of this type (collection descriptors) by a single Retrieval Manager.}
guide_url	site_url + "icsdoc" + "/" + document_path {Same as GetVirtualDocument message.}

Table 5b. Messages in the ICS Guide Protocol

Message Name	Message Format	How It Is Used	Specific Examples
FieldSearchForm	site_url + "icssearch" + "/" + "searchform"	ICS Client sends this message to ICS Guide Server, which runs <i>icssearch</i> script. This message causes script to read Attributes Default File, generate a fielded search form, and returns form to client to display.	http://ijk.gov/icssearch/searchform
GetCollection	cip_url_prefix + cip_client + "?" + ("collection=" + collection_id + ["&"])*	Link from document to collection for an ICS Client. CIP Client returns collection search results to CIP Client.	z39.50s://pdq.org/zclient?collection="CID_erbe_012"&collection="CID_erbe_013"
GetCollectionGuest	cip_url_prefix + cip_client + "?" + "user=guest&passwd=guest" + ("&collection=" + collection_id)*	Link from document to collection for a general Web browser. Links to CIP Client. Starts a session, shows collection search results.	z39.50s://pdq.org/zclient?user="guest"&passwd="guest"&collection="CID_erbe_012"
GetDirectory	site_url + "icsdoc"	Indexing engines send this message to ICS Guide Server, which has Guide Translator run the <i>icsdoc</i> script without path information. Returns a list of URLs of all virtual documents reachable through the site.	http://abc.org/icsdoc
GetVirtualDocument	site_url + "icsdoc" + "/" + document_path + ["&ICS_CLIENT"]	Indexing engines, browsers, and ICS Clients send this message to ICS Guide Server, which has Guide Translator run the <i>icsdoc</i> script with the path information to a document. Returns a virtual document. ICS Clients append the "&ICS_CLIENT" to notify the Guide Translator which type of collection link to insert.	http://abc.org/icsdoc/docs/toms.html
IndexAdd	site_url + "icsindex" + "?" + "mode=add&guide=" + guide_url	ICS Guide Indexer sends this message to the ICS Guide Indexer at other sites, which instructs them to update their indexes with the new guide document.	<a guide_dir="" href="http://abc.org/icsindex?mode=add&guide=" http:="" icsdoc="" newdoc.html"="" pdq.gov="">http://abc.org/icsindex?mode=add&guide="http://pdq.gov/icsdoc/guide_dir/newdoc.html"
IndexNew	site_url + "icsindex" + "?" + "mode=new&guide=" + guide_url	An ICS Site Administrator sends this message manually (or browser client software sends it) to ICS Guide Indexer at his site, notifying it of a new guide to be indexed at the site.	<a guide_dir="" href="http://pdq.gov/icsindex?mode=new&guide=" http:="" icsdoc="" newdoc.html"="" pdq.gov="">http://pdq.gov/icsindex?mode=new&guide="http://pdq.gov/icsdoc/guide_dir/newdoc.html"

IndexValues	site_url + "icsindex" + "?" + "mode=values&guide=" + guide_url + ("&" + attribute + "=" + value)*	ICS Site Administrator browser sends this message to ICS Guide Indexer to update the Guide Metadata Database with the information provided. This is done during guide ingest.	http://uvw.org/icsindex?mode=values&guide="http://uvw.org/icsdoc/newdoc.html"&AuthorName="John Doe"&PublicationDate="11 June 1997"& . . .
SearchFields	site_url + "icssearch" + "?" + (attribute + "=" + value + ["&"])*	ICS Client sends this message to ICS Guide Server, which runs <i>icssearch</i> script. This queries the fielded index with the supplied values for specified attributes. If more than one attribute/value pair is specified, an "&" appears before each additional pair. Returns a list of URLs of virtual documents that have at least one pair in its entry in the Guide Metadata Database. (Note this implies logical OR between pairs). A search of both fielded and free-text indexes can be done by using "free_text" as one of the attributes and the text as the value.	http://xyz.gov/icssearch?AuthorName="John Doe"&PublicationDate="11 June 1997"&free_text="radar images"
SearchFreeText	site_url + "icssearch" + "?" + "free_text=" + text	ICS Client sends this message to ICS Guide Server, which runs <i>icssearch</i> script. This queries the free-text index with the supplied text. Returns a list of URLs of virtual documents that contain the text.	http://abc.org/icssearch?free_text="OZONE GSFC"
UpdateCollections	site_url + "ics_collection_update" + "?" + "guide=" + guide_url + "&" + "collection=" + collection_id	Collection Management Tool sends this message to ICS Guide Server when the ISA adds a guide_url to a collection in the Collections Data Base. The server has the Guide Translator run <i>ics_collection_update</i> script, which updates Collections Mapping File.	http://xyz.gov/ics_collection_update?guide="http://xyz.gov/icsdoc/arb_48_in_lidar.html"&collection="CID_arblidar"
UpdateMetadata	site_url + "ics_metadata_update" + "?" + "guide=" + guide_url + "&" (attribute + "=" + value + ["&"])*	ICS Guide Index sends this message to ICS Guide Server when it has Guide Attribute/Value pairs for a new document added to the system. The server has the Guide Translator run <i>ics_metadata_update</i> script, which updates the Guide Metadata Database.	http://pdq.gov/ics_metadata_update?guide="http://pdq.gov/icsdoc?srb.html"&AuthorName="John Doe"&PublicationDate="11 May 1995"&DocumentName="Surface Radiation Budget (SRB) Langley DAAC Data Set Document"

3.2 File Format Specifications

3.2.1 Attribute Mapping File

Each agency site will have and maintain an Attribute Mapping File. This file is used by the Guide Translator during document ingest, when guides are added to the ICS Guide System (described fully in Appendix B). This file will contain a map of agency attributes to Guide Attributes (listed in Table 4). The file can also contain a map of agency keywords to Guide Keywords, if these are specified. The map will be used quite simply as a lookup table. The format of the file is as follows:

- For each attribute mapped, there will three lines.
 - The first line contains exactly one agency attribute.
 - The next line contains the Guide Attribute that corresponds to the agency attribute.
 - The next line will be blank, used as a separator.

For example, the file at one site may look like this:

```
Author
AuthorName

Published
PublicationDate

Title
DocumentName

Updated
RevisionDate
```

This file maps only four attributes. During guide document ingest, the Guide Translator's *icsdoc* script will search through the guide document for values for attributes mapped in the Attribute Mapping file. If found, *icsdoc* will create Meta tags with the mapped Guide Attribute and the found value, and insert them into the virtual document it produces.

3.2.2 Collections Mapping File

Each agency site will have and maintain a Collections Mapping File. This file contains the collection IDs for those collections to which each ICS guide document refers, if any. Each guide can generally be associated with one or more data collections. The file is used by the Guide Translator's *icsdoc* script, when it produces the virtual document. The format of the Collections Mapping File is quite simple:

- For each guide document there is one line containing its virtual document URL.
- This is followed by the collection IDs (ItemDescriptorId) for that document, one per line.
- Finally there is a blank line after the collection IDs as a separator.

So multiple collections for a single guide will be grouped under a single document entry. For example, the file at one site may look like this (where . . . implies more entries):

```
http://abc.org/icsdoc/documents/erbe.html
CID_erbe_001
CID_erbe_002
CID_erbe_003
CID_erbe_004

http://abc.org/icsdoc/documents/arb.html
CID_arblidar

http://abc.org/icsdoc/documents/nimbus7.html
CID_nimbus_7

. . .
```

The file will be updated by software, *ics_collection_update*, in the Guide Translator component. When the ICS Site Administrator (ISA) updates a collection descriptor in the Collections Database at a site, he may enter any associated guide document URLs in the RelatedItemID field. The ISA uses the Collection Management Tool to update the collection descriptor (see Figure 1). The guide URL must be in the form of the GetVirtualDocument IGP message with the string "&ICS_CLIENT" appended. This is so that when a guide is retrieved from a data collection, the guide is served through the ICS Guide Server, and the server will know the guide is being served to an ICS Client. The Collection Management Tool then contacts the ICS Guide Server with an UpdateCollections message (see Table 5b). The server then has the Guide Translator run *ics_collection_update* with the collection ID and guide URL, which in turn updates the Collections Mapping File with the new information.

3.2.3 Guide Metadata Database

Each agency site will have and maintain a Guide Metadata Database. For each ICS guide reachable through the site, it will have values for all of the mandatory Guide Attributes defined in Table 4. The format of the database will depend upon the database software used for implementation. Here we simply specify the fields.

- Field 1 (or the primary key) will be the guide URL.
 - The field name will be "Guide".
 - The value stored in the field will be the URL of the virtual document.
- Field 2 will be the first mandatory Guide Attribute.
 - The field name will be the attribute name.
 - The value stored in the field will be the attribute's value for the guide document.
- Field 3 will be the next mandatory Guide Attribute.
 - The field name will be the attribute name.
 - The value stored in the field will be the attribute's value for the guide document.
- . . . and so on for the remaining mandatory attributes.

The database is used by the *icsdoc* script in the Guide Translator when generating virtual documents. The script will insert invisible Meta tags containing the attribute/value pairs into the virtual document.

The Guide Metadata Database will be generated and updated during document ingest (described fully in Appendix B). When a new guide document is added to the system, ICS Guide Index is run in order to index the new document. Through attribute mapping, value defaulting, and the ICS Site Administrator entering values, the software gathers values for all mandatory Guide Attributes, and possibly values for optional ("use") attributes. The guide URL and attribute/value pairs are sent to the ICS Guide Server with the UpdateMetadata message (see Table

5b). The server has the Guide Translator run the *ics_metadata_update* script, which in turn updates the Guide Metadata Database with the supplied values.

3.2.4 Attribute Defaults File

Each agency site will have a local version of a file that will contain site specific default values for all the mandatory Guide Attributes (see Table 4). The file can also contain default values for optional ("use") attributes. The file will be used by ICS Guide Server to present the ICS Client user the field names that can be used for a fielded search. It will also be used by ICS Guide Indexer to speed adding guides to the system (see Appendix B). The format for this file is as follows:

- For each Guide Attribute, there will be three lines.
 - The first line contains exactly one Guide Attribute.
 - The next line contains the default value for the attribute.
 - The next line will be blank, used as a separator.

For example, the file at one site may look like this (where . . . implies more entries):

```
AuthorName
European Space Agency

PublicationDate
1 January 1990

DocumentName
ESA Guide Document

RevisionDate
1 January 1990

. . .
```

This file must be hand made at each site, probably by the ICS Site Administrator (ISA). However, once it is made it will greatly reduce the burden on the ISA later during document ingest.

3.2.5 icsdoc.html Site Listing Page

A top-level Web page, *icsdoc.html*, must be created which contains hyperlinks to the ICS Guide Servers at all known agency sites. These URLs are actually GetDirectory messages to the ICS Guide Servers, which pass the message to the Guide Translator's *icsdoc* script. This top-level document will be mirrored at each agency site with conventional mirror software. In this way, the document only needs to be updated at one site, yet it can be accessed at any agency site. The document format is:

- Simply a list of GetDirectory messages, one for every ICS Guide System site, one per line.

For example, this document could look like this (where . . . implies more entries):

```
http://abc.org/icsdoc
http://def.org/ics/icsdoc
http://ghi.org/cip/scripts/icsdoc

. . .
```


`http://pdg.gov/icsdocserver/icsdoc`

This file only needs to change when sites are added, removed, or have their hostnames changed. It doesn't need to change when guides are changed or added to an existing collection of guide documents. The file is used by general Web index engines and ICS Guide Indexers to index all of the ICS guide documents. It is also used by ICS Guide Index to determine sites to send IndexAdd messages when adding new guides to the system.

3.2.6 ICS Free-Text Index

Each site will have a free-text index. This index is created by the ICS Guide Indexer. The index will contain words found in ICS guide documents at all sites and the corresponding guide URLs. The format of the index will depend upon the indexing engine software used for the implementation. Here we simply specify the fields.

- The index contains one entry for every unique word found in all of the guide documents indexed.
- For each entry, there will be multiple values of guide URLs of virtual guides that contained that word.
- The guide URL is of the format specified in the ICS Guide Protocol Messages section 3.1.2 for the GetVirtualDocument message.

A search for free formatted text must return a list of guide URLs of virtual guides that contain the specified text.

3.2.7 ICS Fielded Index

Each site will have a fielded index. This index is created by the ICS Guide Indexer. The index will contain Guide Attributes and the corresponding guide URLs for every ICS guide document at all sites. The format of the index will depend upon the indexing engine software used for the implementation. Here is an approach in which we can easily specify the fields, and to give you an idea of how it works:

- The index will contain a record for each guide document indexed.
- Each record will contain a field named "Guide".
- The value of the "Guide" field will be the guide URL of the document indexed.
- The guide URL is of the format specified in the ICS Guide Protocol Messages section 3.1.2 for the GetVirtualDocument message.
- Each record will contain a field for every Guide Attribute (see Table 4).
- The name of each field is the name of the Guide Attribute.
- The value stored in each field will be the attribute's value for the guide document, if any.

A fielded search must return a list of guide URLs of virtual guide documents that contain any of the specified values for the specified Guide Attributes.

3.3 ICS Guide System Components

3.3.1 ICS Guide Server

The ICS Guide Server handles all HTTP connections to the system. It handles the guide search and retrieval requests. It provides the interface to the system, handling some messages by passing them to the Guide Translator. The server is composed of the following components.

3.3.1.1 HTTP Daemon

The ICS Guide Server has an HTTP daemon. The daemon listens at a computer port for HTTP connections. It accepts HTTP connections, spawning a process to handle each one so that it can continue listening for more connections. The handling of the connection depends on the IGP message (see Table 5b). Here are the actions based on the IGP message:

Table 6. ICS Guide Server Actions for Messages

IGP Message	ICS Guide Server Action
GetDirectory	Invoke Guide Translator, pass message.
GetVirtualDocument	Invoke Guide Translator, pass message.
SearchFreeText	Invoke <i>icssearch</i> with parameters.
FieldSearchForm	Invoke <i>icssearch</i> with parameter.
SearchFields	Invoke <i>icssearch</i> with parameters.
UpdateCollections	Invoke Guide Translator, pass message.
UpdateMetadata	Invoke Guide Translator, pass message.

3.3.1.2 *icssearch* Software

icssearch is a software component of the ICS Guide Server that performs two functions: generating a fielded search form and performing guide searches against the ICS indexes.

If *icssearch* is invoked with the "searchform" parameter, it will return a fielded search form. Here are the steps it follows to perform this function:

- Read the Attribute Defaults File.
- Form the list of Guide Attributes.
- Create an HTML search form with text fields preceded by each Guide Attribute.
- Add one more text field for free-text entry.
- Send form back to ICS Client via HTTP.

If *icssearch* is invoked with the "free_text=<text>" parameters, it will return a list of matching guide documents. Here are the steps it follows to perform this function:

- Construct a query containing the specified text. This may be simply one string of the specified text, but more likely will require parsing the text for Boolean elements such as "and", "or", and "not".
- Send the query to the ICS free-text index.
- This will return a list of guide URLs that contain the specified text.
- Format the list into an HTML page.
- Append the string "&ICS_CLIENT" to each of the URLs. This will notify the ICS Guide Server that the connection is from an ICS Client during guide retrieval.
- Send page back to the ICS Client via HTTP.

If *icssearch* is invoked with attribute=value pairs, it will return a list of matching guide documents. Here are the steps it follows to perform this function.

- Construct a query containing the specified attribute=value pairs.
- If one of the attributes is "free_text", then form a free-text index query from the specified text.
- Send the query to the ICS fielded index.
- If there is a free-text query, send this to the free-text index.
- This will return a list of guide URLs that contain the specified values for the specified attributes.
- If there is a free-text query, merge the list returned from the free-text query with the list from the fielded query.
- Format the list into an HTML page.
- Append the string "&ICS_CLIENT" to each of the URLs. This will notify the ICS Guide Server that the connection is from an ICS Client during guide retrieval.
- Send page back to the ICS Client via HTTP.

An advanced service of the fielded search would be to allow some additional Boolean logic such as "greater than" and "less than". For example, it could provide the capability of requesting guides published after a specific date by specifying the PublicationDate value as ">1 January 1997". The *icssearch* search engine must know how to parse and query this form, and the index must provide the capability, in order for this service to be available.

3.3.1.3 Files

The ICS Guide Server also contains one file: the Attribute Defaults File previously described in sections 3.2.4.

3.3.2 Guide Translator

The Guide Translator primary function is to modify guide documents, generating virtual documents that are passed back via HTTP. This modification is the primary solution that enables this system to meet its requirements. The Guide Translator is composed of three functionally independent pieces of software. The translator invokes each one of them depending upon the message it receives from the ICS Guide Server. Here are the actions based on the IGP message (see also Table 5b):

Table 7. Guide Translator Actions for Messages

IGP Message	Guide Translator Action
GetDirectory	Invoke <i>icsdoc</i> without path information.
GetVirtualDocument	Invoke <i>icsdoc</i> with path information.
UpdateCollections	Invoke <i>ics_collection_update</i> with parameters.
UpdateMetadata	Invoke <i>ics_metadata_update</i> with parameters.

3.3.2.1 icsdoc Software

icsdoc is the primary software component of the Guide Translator. It must have efficient on-line access to all the physical documents accessible through the site. The software runs when guide documents are being indexed **and** when they are being retrieved. It will perform two different functions depending on how it is invoked.

3.3.2.1.1 icsdoc Without Path Information Returns a Directory

icsdoc can be invoked without parameters. This happens when an ICS Guide Server gets contacted by a Web indexing engine with a GetDirectory IGP message. For example, consider contacting this (fake) URL:

`http://abc.org/icsdoc`

This tells the ICS Guide Server to invoke the Guide Translator, which invokes *icsdoc* without any additional path information. When *icsdoc* is invoked in this manner, it returns a "directory" document. This is an HTML document consisting of guide URLs; hyperlinks to all guide documents reachable through the agency site, with one hyperlink per line. Note that in this ICS Guide System, *icsdoc* will NOT be invoked in this way by any person. This huge directory document will not be viewed by people. The software is invoked this way by automatic Web crawling indexing engines, including the ICS Guide Indexer. This is how the indexing engines will find all of the guide documents.

The directory document could be a static maintained file but this would pose a manual maintenance issue, possibly leading to out of date information and system failure. Therefore, the directory document will be generated dynamically by having *icsdoc* trace the document directory structure to all guide documents reachable at the agency site. Recall that this operation occurs during indexing, not while people are retrieving a guide document, and so does not pose any delay for the users.

Note that the hyperlink to each guide document in the directory document is NOT the URL of the physical document, but instead is a guide URL which is really a GetVirtualDocument IGP message that links back to the site's ICS Guide Server. Recall that this message is of the form: `site_url + icsdoc/" + document_path`. So, the directory document returned by *icsdoc* could look like:

```
http://abc.org/icsdoc/docs/erbe.html  
http://abc.org/icsdoc/docs/fire.html  
http://abc.org/icsdoc/docs/isccp.html  
http://abc.org/icsdoc/docs/nvap_tovs.html  
http://abc.org/icsdoc/docs/sage.html
```

In this example, the document path "docs/<filename>" must be enough information for the *icsdoc* software to access the desired document.

3.3.2.1.2 *icsdoc* with Path Information Returns a Virtual Document

If *icsdoc* is invoked with any path information following "*icsdoc*" in the URL, then the path information will be interpreted as identifying a particular physical document. In this case, the *icsdoc* will return a "virtual document", generated from the physical document referred to in the path information. For example, consider contacting this (fake) URL:

```
http://abc.org/icsdoc/docs/erbe.html
```

This tells the ICS Guide Server to invoke the Guide Translator, which invokes *icsdoc* with the additional path information "docs/erbe.html". When *icsdoc* is invoked in this manner, it will read and modify the erbe.html guide document and return it. In short, this generated virtual document will contain all the HTML of the physical document, plus invisible Meta tags containing Guide Attribute/Keyword pairs and a visible link to related data collection(s). The basic steps are:

- Look the guide URL up in the Guide Metadata Database.
- Extract the Guide Attributes and Keywords for the guide document.
- Construct invisible HTML Meta tags containing the Attribute/Keyword pairs, and insert them into the virtual document.
- Look the guide URL up in the Collections Mapping File.
- Extract any collection IDs for the guide.
- Create a visible hyperlink to the collection in the virtual document.
- Return the modified, virtual document.

The format for the Meta tags is:

```
<meta name="Keywords"  
  content="attribute1=keyword1, keyword1,  
          attribute2=keyword2, keyword2,  
          attribute3=keyword3, keyword3,  
          . . .">  
<meta name="attribute1" value="keyword1">  
<meta name="attribute2" value="keyword2">  
<meta name="attribute3" value="keyword3">
```

Note that there are more steps involved when *icsdoc* is invoked during guide ingest, adding a new guide document to the system. A full detailed description of *icsdoc* document processing and the reason for the Meta tag format is explained in Appendix A. The collection links are in the form of IGP messages (See Table 5b). If the ICS Guide

Server was contacted by an ICS Client, the guide URL will have the string "&ICS_CLIENT" at the end of the GetVirtualDocument message. In this case, the collection link is in the form of the GetCollection message. For example:

```
z39.50s://pdq.org/zclient?collection="CID_erbe_012"
```

If the ICS Guide Server was contacted by anything else (indexing engines or browsers), then the collection link is in the form of the GetCollectionGuest message. For example (ignore line wrapping):

```
z39.50s://pdq.org/zclient? user="guest"&  
passwd="guest"&collection="CID_erbe_012"
```

In Figure 1 you will note that there is part of the Guide Translator that is considered ICS related and not strictly part of ICS. This part of the Guide Translator is site specific software that enables *icsdoc* to access the guide documents, and provides those guides in HTML format. If the guides are in HTML or plain text, and are located in a directory at the site, then this part of the translator is nothing - *icsdoc* has everything it needs. If the guides are in a database or not located locally, then a small site-specific software layer must provide the guides to *icsdoc*. If the guides are not in HTML or plain text format, then this part of the translator must convert the guide to HTML on the fly. For instance, MS-Word documents could be included in the system only if the translator had an additional "plug-in" to convert the format to plain text or HTML.

The *icsdoc* actions based on IGP message are summarized in Table 8.

Table 8. *icsdoc* Actions for Messages

IGP Message	<i>icsdoc</i> Action
GetDirectory	Return HTML page with all site guide URLs.
GetVirtualDocument (with "&ICS_CLIENT")	Return virtual document - guide with attribute/value Meta tags and GetCollection link to data collections.
GetVirtualDocument (without "&ICS_CLIENT")	Return virtual document - guide with attribute/value Meta tags and GetCollectionGuest link to data collections.

3.3.2.2 *ics_collection_update*

ics_collection_update is a software component of the Guide Translator. It is invoked when the UpdateCollections IGP message is sent to the ICS Guide Server, which passes it to the Guide Translator. The translator invokes *ics_collection_update* with the parameters: guide=<guide_url> and collection=<collection_id>. The software performs the following steps:

- Open the Collections Mapping File.
- If the guide URL is not already in the file, create a new entry containing the guide URL on one line, and the collection ID on the next. See file format in section 3.2.2.
- If the guide URL is in the file, add the new collection ID to the end of the list of collection IDs for that guide URL. See file format in section 3.2.2.

3.3.2.3 *ics_metadata_update*

ics_metadata_update is a software component of the Guide Translator. It is invoked when the UpdateMetadata IGP message is sent to the ICS Guide server, which passes it to the Guide Translator. The translator invokes *ics_metadata_update* with the parameter list:

```
guide=<guide_url>, <attribute1>=<value1>, <attribute2>=<value2>, . . . , <last attribute>=<last value>
```

The software performs the following steps:

- Open the Guide Metadata Database.
- If the guide URL is not already in the database, create a new record containing the guide URL and attribute/value pairs and enter it into the database.
- If the guide URL is already in the database, update the record with the new set of attribute/value pairs.

3.3.2.4 Files

The Guide Translator also contains three files: the Attribute Mapping File, the Collections Mapping File, and the Guide Metadata Database. These files were previously described in sections: 3.2.2, 3.2.3, and 3.2.1 respectively.

3.3.3 ICS Guide Indexer

The ICS Guide Indexer creates and updates the ICS Free-Text and Fielded indexes of ICS guide documents. The software is a specialized Web-crawling index engine. There are many public domain engines that will suffice with some modifications. Let us call this engine, *icsindex*. As you will see, it must be Web accessible for the system to be distributed. The ICS Guide Indexer is composed of the three parts: the *icsindex* indexing engine software, the free-text index, and a fielded index.

3.3.3.1 icsindex Batch Mode

To begin, a site runs the index engine, *icsindex*, pointed at a single starting point, namely the top-level Web page which contains hyperlinks to all ICS Guide Servers at all known agency sites (see section 3.2.5). Let's say that one is available at the following (fake) URL:

`http://cip.org/icsdoc.html`

Recall that this file is mirrored at every site. *icsindex* can be run manually by the ISA simply executing it. Run this way, it creates a free-text index and fielded index of all the ICS guide documents at all sites. General Web search/index engines such as Alta Vista create only free-text indexes. They will perform the same steps as *icsindex* except for creating the fielded index. Here are the steps that *icsindex* follows:

- Contact icsdoc.html - retrieve page of all ICS Guide Servers at all agency sites.
- Contact each of the links on the page. Recall that these links are actually GetDirectory IGP messages to the ICS Guide Servers. They return "directory" documents of all guide documents accessible from their site (see section 3.3.2.1.1 icsdoc Without Path Information Returns a Directory).
- Contact each of the links on the "directory" documents. Recall that these links are actually GetVirtualDocument IGP messages to the ICS Guide Servers. They return virtual documents - the guide documents with additional Meta tags containing the Guide Attribute/value pairs.
- For the free-text index, the indexing engine creates records of every unique word found in all the guide documents, and stores the guide URLs for those words in the index (see section 3.2.6).
- For the fielded index, the indexing engine extracts attribute/value pairs from the Meta tags of the virtual documents, creates records containing the guide URLs and the attribute/value pairs (see section 3.2.7).

A detailed reiteration of the link traversal is presented in Appendix C with examples to clarify the concept.

Note that this process only indexes guide documents that are already "in the system". Because the virtual documents need to have values for all required Guide Attributes, which are stored in the Guide Metadata Database, there is a separate method for adding guide documents to the system. In the following section, we discuss this process.

3.3.3.2 icsindex Add Mode

The indexes could be mirrored at each agency site, but we wish any site to be able to update the indexes as they add new guides to their site. So, we propose that the *icsindex* software at each site can also be used to add guide URLs to the site index, and then notify the *icsindex* at other sites to update their indexes accordingly. This mode of *icsindex* is called the "Add" mode. The details of this Add mode are described in Appendix B, because there are many steps. In an abbreviated manner, here is how it works (see Figure 2):

1. The ICS Site Administrator adds a new guide document to the site database (or the document is made visible to *icsdoc* - it may not necessarily be located at the site). The agency site Guide Translator must be able to access the document when *icsdoc* is invoked with some path information. E.g.:

`http://xyz.gov/icsdoc/newdoc.html`

2. The ICS Site Administrator runs *icsindex* with the IndexNew IGP message (see Table 5b) containing the guide URL for the new guide document (as in the previous step) as an argument. Entered through a browser, the URL is:

```
http://xyz.gov/icsindex?mode=new&guide=  
"http://xyz.gov/icsdoc/newdoc.html"
```

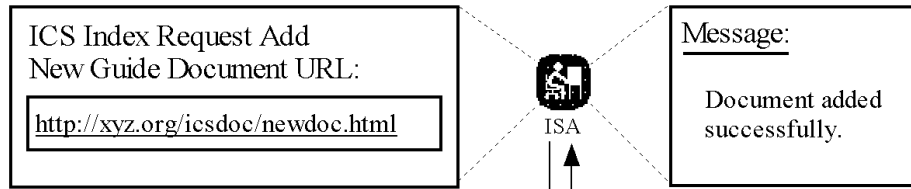
A fancy HTML form page could be used to present a box for the manager to enter the guide URL, but is not required. See Figure 2.

3. *icsindex* uses its Web-crawling index engine on the guide URL. While *icsindex* contacts the URL, it is actually sending a GetVirtualDocument IGP message to the ICS Guide Server. The server sends the message to the Guide Translator, which runs *icsdoc* with the path information for the new guide. The virtual document is generated and sent back to the engine. Note that at this point the guide does not have an entry in the Guide Metadata Database, and will need to undergo attribute mapping and specification by the ICS Site Administrator (see Appendix B for details).
4. When that is done, *icsindex* sends the UpdateMetadata IGP message to the ICS Guide Server, which causes a new entry into the Guide Metadata Database.
5. The index engine then indexes the virtual document that contains the inserted Guide Attribute/Keyword annotations. It updates the site's free-text and fielded indexes with searchable keywords from the virtual document and the guide URL.
6. *icsindex* then contacts the *icsindex* at every other agency site found in the *icsdoc.html* file with the URL from step 2 as a parameter. This is the IndexAdd IGP message. This step is the reason why *icsindex* must be Web accessible. As you can see in Figure 2, the httpd in the ICS Guide Indexer provides the Web access. An alternate design would be to use the httpd in the ICS Guide Server, and have the server pass the message to the ICS Guide Indexer.
7. When the *icsindex* at each other site gets the IndexAdd message, it proceeds exactly as in steps 3-5, updating the site indexes with the new guide URL. This time the software doesn't perform step 6, contacting other sites.

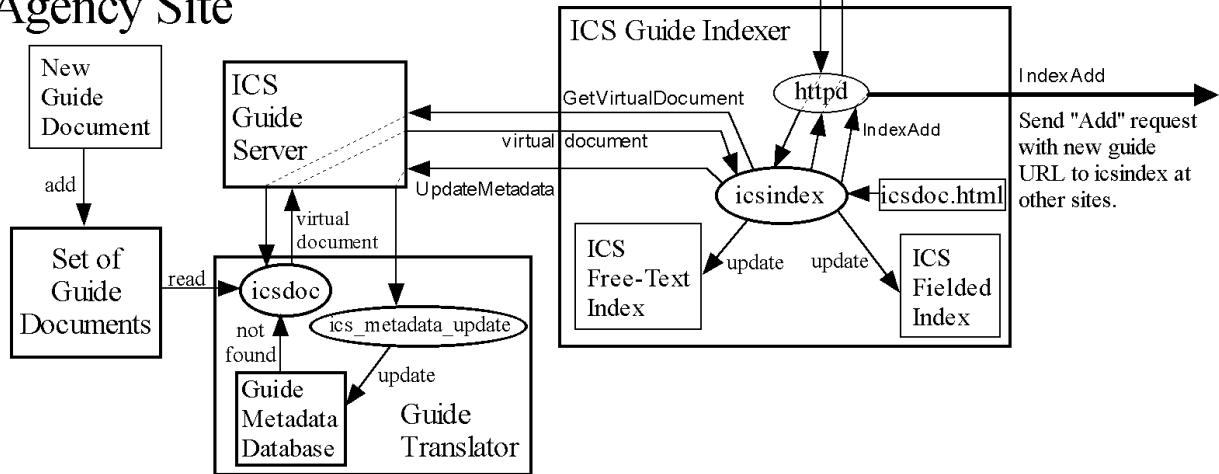
Thus guides can be added to the system at any site, and every site will maintain up-to-date indexes. See Appendix B for details of adding guides to system. Some steps were left out in the description presented here and in Figure 2.

Figure 2. Adding a New Guide Document to the System

Browser



Agency Site



When a new site enters the ICS Guide System, the ICS Site Administrator will first run *icsindex* in "Batch" mode, pointed at the *icsdoc.html* file. This will create indexes of all guides at all the other sites. Then the ICS Site Administrator will run *icsindex* in "Add" mode, adding one "local" guide document at a time to the system.

3.3.3.3 HTTP Daemon

ICS Guide Index needs an *httpd* in order to contact ICS Guide Indexer at other sites with the *IndexAdd* message, sent when new guides are added to the system. It also receives the *IndexNew* message when the ICS Site Administrator enters the guide URL of a new guide. See Figure 2. Note also that the *httpd* provides access to the *icsdoc.html*, site listing file. This file is accessed both by ICS Guide Indexer and general Web crawling indexing engines (such as Alta Vista, Lycos, Infoseek, etc.). As mentioned in step 6 above, an alternate design would be to use the *httpd* in the ICS Guide Server to handle the index messages. It would also need to handle the access to *icsdoc.html*. In that case the ICS Guide Server's *httpd* would have to pass incoming index messages to the local ICS Guide Indexer, and pass the outgoing *IndexAdd* message to the ICS Guide Servers at other sites. The advantage to this alternate design is that only one system port is used for the ICS Guide System. The disadvantage is that the one *httpd* could be quite busy handling both index updates and serving guides.

3.3.3.4 Files

The ICS Guide Indexer also contains three files: the ICS Free-Text Index, the ICS Fielded Index, and the sites listing page *icsdoc.html*. The file formats were discussed previously in sections 3.2.5, 3.2.6, and 3.2.7.

3.3.3.5 Summary of Actions

A summary of ICS Guide Indexer actions based on message is listed in Table 9.

Table 9. ICS Guide Indexer Actions for Messages

IGP Message	ICS Guide Indexer Action
IndexAdd	Run <i>icsindex</i> in Add mode. Update indexes with supplied guide URL.
IndexNew	Run <i>icsindex</i> in Add mode. Update indexes with supplied guide URL and notify ICS Guide Indexers at all other sites.

IndexValues	Send UpdateMetadata message to ICS Guide Server to update Guide Metadata Database.
(no message)	Run <i>icsindex</i> in Batch mode. Update indexes with all guides in the ICS Guide System.

3.4 Related Components

3.4.1 Collection Management Tool

The Collection Management Tool is not a component of the ICS Guide System, but the system depends on it to perform one of its key functions. The ISA uses this tool to update collections in the Collections Database (see Figure 1). The ISA enters guide URLs for the RelatedItemId field in the collection descriptors in the Collections Database using this tool. The Collection Management Tool then contacts the ICS Guide Server with an UpdateCollections message (see Table 5b). The server then has the Guide Translator run *ics_collection_update* with the collection ID and guide URL, which in turn updates the Collections Mapping File with the new information.

3.4.2 ICS Client

The ICS Client provides the user with an integrated environment for utilizing CIP services and the ICS Guide System services. It contains the CIP Client, which enables services such as collection searches and product searches. The ICS Client also contains an IGP Client that provides user interface to the ICS Guide System. This enables guide search and retrieval services. The ICS Client handles the connection between CIP and IGP services using IGP (URL) messages.

3.4.3 IGP Client

The IGP Client provides the user interface to the ICS Guide System. It presents the user with the capability to submit guide searches. This includes free-text and fielded searches for ICS guide documents that contain the specified text or attribute/value pairs. The client must use the messages of the ICS Guide Protocol (see Table 5b). While it presents search forms and HTML pages to the user, it must also form the correct IGP messages to accomplish the user's request.

3.4.4 Guide Documents

The primary purpose of the ICS Guide System is to provide access to ICS guide documents. The guide documents are often stored in collections, located at (or reachable from) the agency sites. These collections may be in databases or could simply be kept in directories. The key issues are:

- The guide documents must be accessible by the *icsdoc* component of the Guide Translator. Files in a directory provide the simplest access. Multiple directories would be ok too, if *icsdoc* knows how to find all of the guides.
- The guide documents must be in a Web presentable form - i.e. HTML. Note that plain text documents are also treated as HTML, since browsers display these documents as text. It may be possible to have the documents in some other format, if the site part of the Guide Translator can convert the document to HTML on the fly. This part of the Guide Translator would be site specific. For instance, PDF guide documents could be included in the system only if the Guide Translator had an additional "plug-in" to convert the format to plain text or HTML. See section 3.3.2.1.

4. Scenarios

The following scenarios explain how parts of the ICS Guide System operate to provide desired functions.

4.1 ICS Guide Search with General Web Search Engine

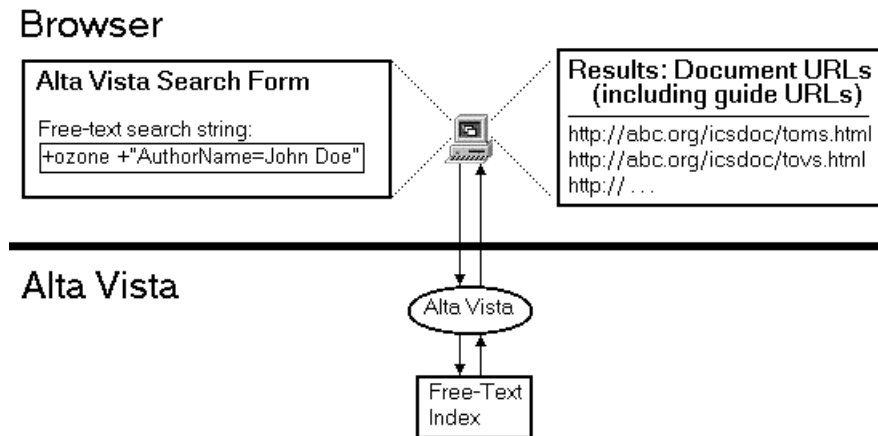
In order for users of general Web search engines, such as Alta Vista, Yahoo, or Lycos, to find ICS guides, the following must take place prior to the search:

- The ICS Guide System must be set up at least at one site.
- Guide documents must have been ingested into the system (see Appendix B).
- The companies hosting Web search engines shall be notified of the URL of one instance of the Sites Listing file `icsdoc.html` (section 3.2.5). Recall that this file will be mirrored at every ICS site.
- The Web search engine must have indexed the ICS guide documents, which it finds from `icsdoc.html`, (see Appendix C).

Thereafter, a person using a Web browser can compose a free-text search query at a general Web search site (such as Alta Vista). If they just enter strings that they expect will be in the ICS guides, they will find them. However, since these general Web search sites index all documents on the WWW, users will likely get matches of documents outside of ICS. They could narrow their search using known Guide Attribute/Keyword pairs. For instance: "AuthorName=John Doe". The search engine simply looks up the URLs for the associated query strings in its index and displays the links for guide documents that contained the free-text. See Figure 3. Note that while the user can use Guide Attribute/Keyword pairs because of the insertion of Meta tags, the search is still a free-text search. These search engines only use free-text. The string "John Doe" will NOT match the string "AuthorName=John Doe" using these engines. The engines don't match word segments. This is why the keyword is inserted twice in the free text Meta tag in section 3.3.2.1.2 (and Appendix A).

Figure 3. Searching for ICS Guide Documents

Using a General Web Search Engine (Alta Vista)



In Figure 3, a user types this string into the Alta Vista search form in their Web browser.

```
+ozone +"AuthorName=John Doe"
```

The + signs require the text following them to be in the matching document. This makes the search look for "ozone" and "AuthorName=John Doe", both in each document. Without the + signs, documents that contain one word or the other word would be listed.

The browser connects to the httpd at Alta Vista, which runs a program that checks its URL database (index) for keys that match strings the user entered. Alta Vista returns to the user's browser, an HTML page containing a list of

hyperlink URLs to documents that contained the strings, in this case guide documents. Note that during indexing the `icsdoc.html` file, the search engine found guide URLs that are actually URLs of the ICS Guide Server with the `GetVirtualDocument` IGP message. The URL returned for an ICS guide will be in the form:

```
site_url + "icsdoc/" + document_path
```

For example, one could look like:

```
http://abc.org/icsdoc/docs/srb.html
```

That is all there is to doing an Alta Vista search. Note that during this process, no part of the ICS Guide System was used.

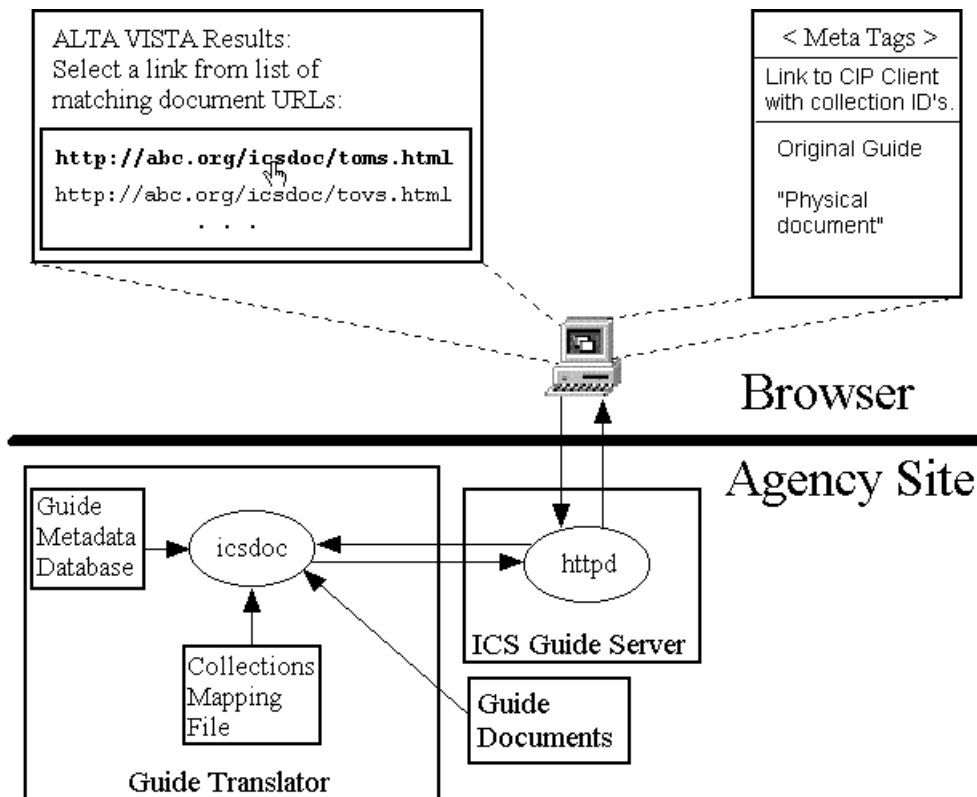
4.2 ICS Guide Retrieval from General Web Search Engine Result

When the user selects one of the links on the page returned from a general Web search engine result, the browser contacts the guide URL. This connects to the `httpd` in the ICS Guide Server at the site indicated in the URL, with the `GetVirtualDocument` IGP message. For example:

```
http://abc.org/icsdoc/docs/srb.html
```

Figure 4 shows the process described here.

Figure 4. Retrieving ICS Guide Documents from Alta Vista Search Results



The server invokes `icsdoc` in the Guide Translator with path information of the guide document. `icsdoc` looks up the guide in the Guide Metadata Database, retrieves the metadata for that guide, and constructs Meta tags in the format given in section 3.3.2.1.2. `icsdoc` then reads the guide using the path information supplied in the message. It then inserts the Meta tags into the header section of the guide. Next, `icsdoc` looks up the guide in the Collections

Mapping File. If it finds it there, it extracts the collection IDs for the guide. *icsdoc* constructs links to the data collections. The URLs of the links would look like (ignore line wrapping):

```
z39.50s://pdq.org/zclient?user="guest"&passwd="guest"&
collection="CID_srbmonth"& collection="CID_srbdaily"
```

The appearance of the link on the page could either be text or a button with text indicating that selecting it will take the user to data collections.

Note that for ICS guide retrieval from Alta Vista results, Alta Vista is not contacted. The browser connects directly to the httpd in the ICS Guide Server at the agency site referenced in the link.

4.3 ICS Client Free-Text Guide Search

An ICS Client user can perform a free-text guide search in the following way. The IGP Client presents a simple form with a single text box. The user enters text that must be in the list of guides that will be returned. See Figure 5. The user submits the form (click a submit button on form). The IGP Client constructs and contacts a URL, which is a SearchFreeText IGP message, containing the text. The URL for the ICS free-text search request is of the form:

```
site_url + "icssearch?free_text=" + text
```

For example:

```
http://abc.org/icssearch?free_text="OZONE and GSFC"
```

Note that the free text can contain Boolean elements such as "and", "or", and "not".

The IGP Client connects to the httpd in the ICS Guide Server at the same site as the client. The server responds to the SearchFreeText message by invoking *icssearch*, the ICS search engine, with the free text as a parameter. *icssearch* parses the string for Boolean elements, and then queries the ICS Free-Text Index. Recall that the index contains guide URLs for all guides at all sites. The query matches guides that contained the specified free-text, and returns the list of guide URLs. *icssearch* constructs an HTML page containing the list of guide URLs and passes it back to the IGP Client through the httpd. The IGP Client displays the page of matching guide URLs to the user.

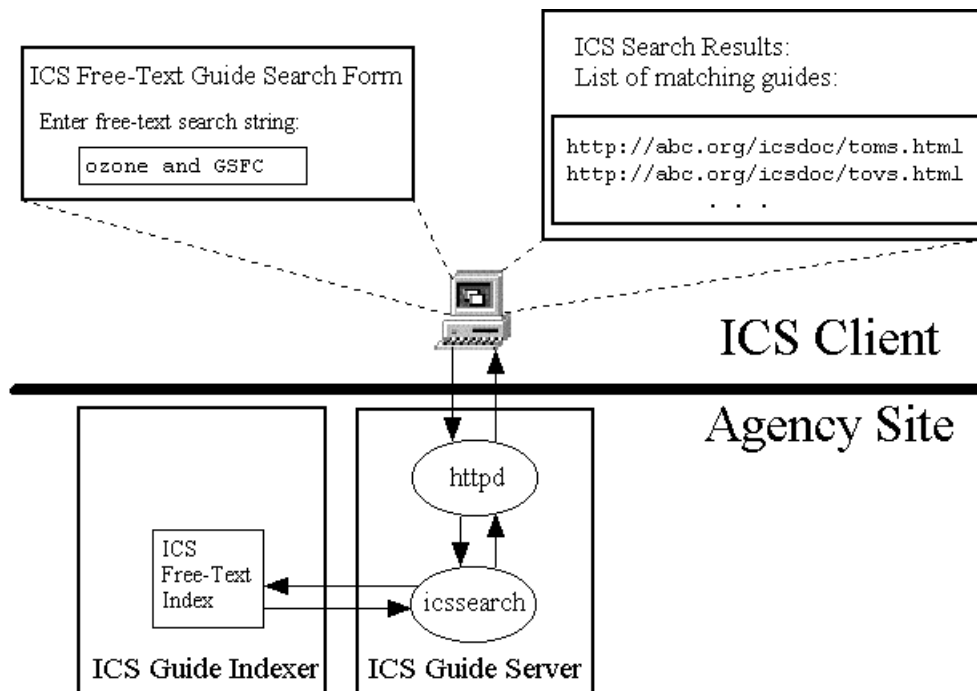
The guide URLs that *icssearch* places in the HTML page are modified by appending the string "&ICS_CLIENT" to them. This is so that when a user of an ICS Client retrieves a guide from the results page, the ICS Guide Server contacted will know the connection is from an ICS Client. The links will therefore be in the form:

```
site_url + "icsdoc/" + document_path + "&ICS_CLIENT"
```

For example, one could look like:

```
http://abc.org/icsdoc/docs/srb.html&ICS_CLIENT
```

Figure 5. ICS Client Free-Text Guide Search



4.4 ICS Fielded Guide Search

An ICS fielded guide search allows the user of an ICS Client to specify values for guide attributes, which will be in the list of guides returned. When a user of the ICS Client wishes to perform a fielded guide search, the user selects a button or link on the IGP Client interface that offers this service. The IGP Client sends the FieldSearchForm IGP message to the ICS Guide Server. This is accomplished through a URL of the form:

```
site_url + "icssearch/searchform"
```

For example (fake URL):

```
http://ijk.gov/icssearch/searchform
```

When the IGP Client contacts this URL, it connects to the httpd in the ICS Guide Server at the ijk.gov site. The httpd invokes the *icssearch* CGI script with the argument "searchform". See Figure 6. *icssearch* reads the Attributes Default File, generates a fielded search form HTML page, and sends it back to the IGP Client to display. The form contains Guide Attribute names for labels for text boxes that the user can fill in. There is an additional text box with the label "Free Text" which the user can use to enter free-text to search. The form will have a submit button on it. The text, labels, and button are contained within a set of HTML <FORM> </FORM> tags.

When the form containing any filled in fields is submitted, the IGP Client connects to the httpd in ICS Guide Server at the same site as the client, and sends the SearchFields IGP message. This message is in the form:

```
site_url + "icssearch?" + attribute1=value1&attribute2=value2& . . .
```

Suppose the user entered the following values in these fields:

```
AuthorName: "John Doe"  
PublicationDate: "June 1997"
```

The search URL would look like (ignoring line wrapping):

```
http://ijk.gov/icssearch?AuthorName="John Doe"&
PublicationDate="June 1997"
```

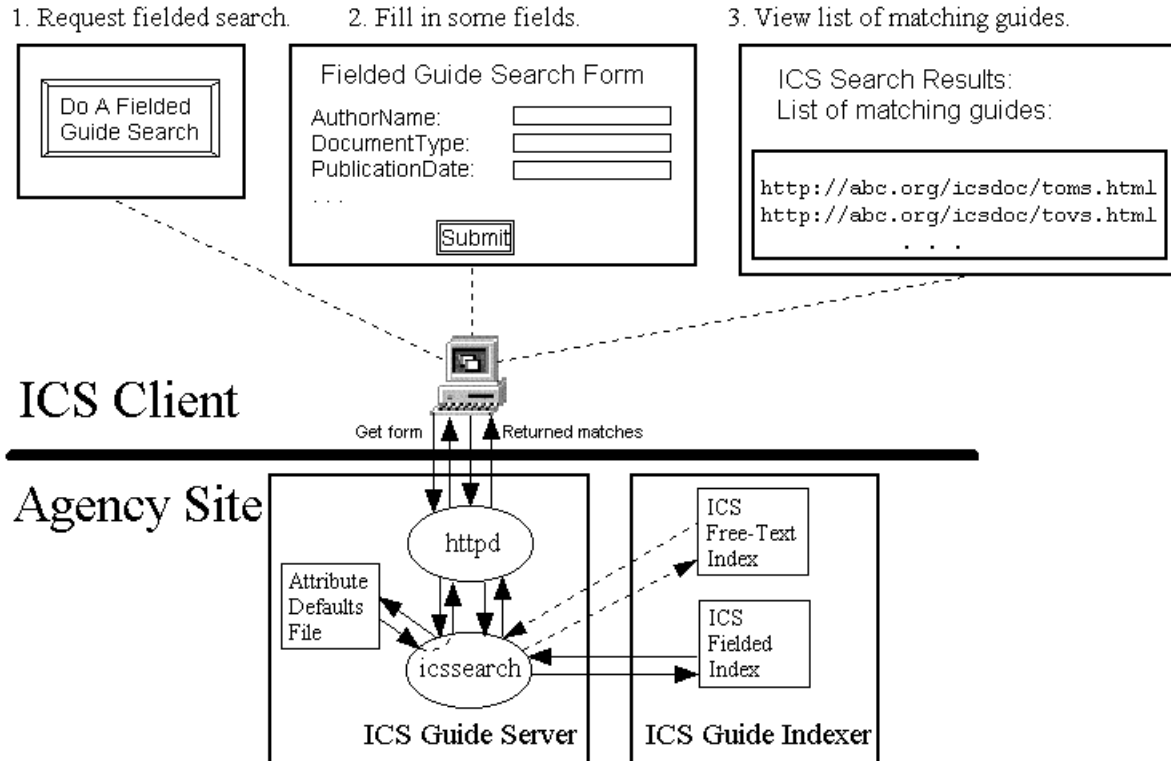


Figure 6. ICS Client Doing Fielded Guide Search

The server responds to the SearchFields message by invoking *icssearch*, the ICS search engine, with the parameter value list. In the example, that list would be:

Parameter	Value
AuthorName	"John Doe"
PublicationDate	"June 1997"

Note that a fielded search can also contain free-text. One of the attributes would then be "free_text". For example:

```
http://ics.gov/icssearch?AuthorName="John Doe"&
PublicationDate="June 1997"&free_text="Oceans"
```

This would result in *icssearch* being invoked with the following parameter value list:

Parameter	Value
AuthorName	"John Doe"
PublicationDate	"June 1997"
free_text	"Oceans"

icssearch queries the ICS Fielded Index. If the parameter "free_text" is supplied, *icssearch* also queries the ICS Free-Text Index. Recall that the indexes contain guide URLs for all guides at all sites. The query to the fielded index matches guides that contained the specified values for the parameters. A query to the free-text index matches guides that contained the specified text, anywhere in them. The queries return a list of guide URLs. *icssearch* constructs an HTML page containing the list of guide URLs and passes it back to the IGP Client through the httpd. The IGP Client displays the page of matching guide URLs to the user as shown in Figure 6. As in the free-text search described in section 4.4, the guide URLs that *icssearch* places in the HTML page are modified by appending the string "&ICS_CLIENT" to them. This is so that when a user of an ICS Client retrieves a guide from the results page, the ICS Guide Server contacted will know the connection is from an ICS Client. For example, one link could look like:

```
http://abc.org/icsdoc/docs/srb.html&ICS_CLIENT
```

An advanced service of the fielded search would be to allow some additional Boolean logic such as "greater than" and "less than". For example, it could provide the capability of requesting guides published after a specific date by specifying the PublicationDate value as ">1 January 1997". The *icssearch* search engine must know how to parse and query this form, and the index must provide the capability, in order for this service to be available.

4.5 ICS Guide Retrieval from ICS Client

After the user has performed a free-text or fielded guide search from an ICS Client, the user can then select one of the guide document links on the results page. When a link is selected, the IGP Client contacts the guide URL. This connects to the httpd in the ICS Guide Server at the site indicated in the URL, with the GetVirtualDocument IGP message. Recall from Table 5b and the preceding two sections, that the string "&ICS_CLIENT" will have been appended to the guide URL. This lets the ICS Guide Server know the connection is from an ICS Client. For example, one such link could be:

```
http://xyz.org/icsdoc/docs/erbe.html&ICS_CLIENT
```

Figure 7 shows the process described here. Note that it is nearly the same process as when guides are retrieved from a general Web search engine results page. The only difference is in the collection links, which will be described shortly.

The ICS Guide Server invokes *icsdoc* in the Guide Translator with path information of the guide document. *icsdoc* looks up the guide in the Guide Metadata Database, retrieves the metadata for that guide, and constructs Meta tags in the format given in section 3.3.2.1.2. *icsdoc* then reads the guide using the path information supplied in the message. It then inserts the Meta tags into the header section of the guide. Next, *icsdoc* looks up the guide in the Collections Mapping File. If it finds it there, it extracts the collection IDs for the guide. *icsdoc* constructs links to the data collections. Because this connection is from an ICS Client, the URLs of the links would look like (ignore line wrapping):

```
z39.50s://pdq.org/zclient?collection="CID_erbe_001"&  
collection="CID_erbe_002&collection="CID_erbe_003&  
collection="CID_erbe_004&collection="CID_erbe_005"
```

The appearance of the link on the page could either be text or a button with text indicating that selecting it will take the user to data collections. Notice that there is no need for user name and password since the ICS client already has that information. Note that the retrieval is from the ICS Guide Server at the site of the matched guide. This may not be the same site where the search was performed. Recall that the indexes contain guide URLs from all sites.

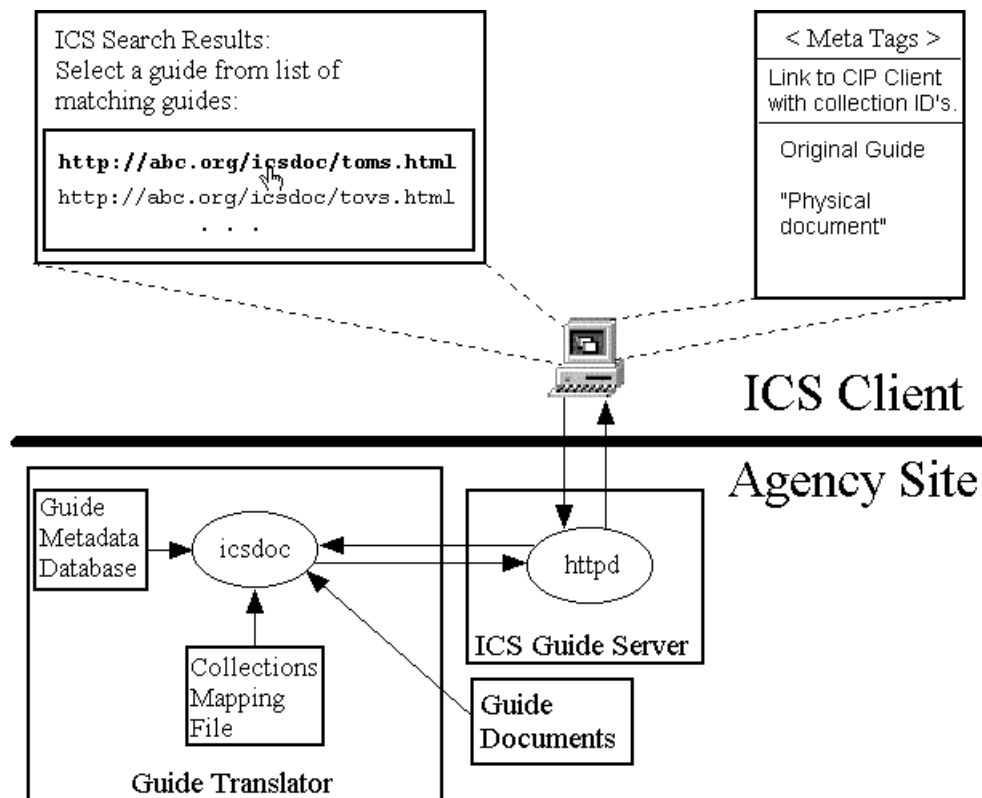


Figure 7. Retrieving ICS Guide Documents from ICS Search Results

4.6 Linking to Collections from Guides

As was described in sections 4.2 and 4.5, links to data collections are inserted into the guide documents in one of two formats by *icsdoc* in the Guide Translator.

4.6.1 Linking to Collections from Guides using ICS Client

If the guide is being served to an ICS Client the link is in the form of the GetCollection IGP message:

```
cip_url_prefix + cip_client + "?" + ("collection=" + collection_id + ["&"])*
```

For example:

```
z39.50s://pdq.org/zclient?collection="CID_erbe_001"&  
collection="CID_erbe_002&collection="CID_erbe_003&  
collection="CID_erbe_004&collection="CID_erbe_005"
```

When the user selects this link, the CIP Client must run a collection search using the specified collection IDs. The search will use the broadest values for time range and geographic area. The collection descriptors will be listed in the user's client.

4.6.2 Linking to Collections from Guides using General Browser

If the guide is being served to a general browser the link is in the form of the GetCollectionGuest IGP message:

```
cip_url_prefix + cip_client + "?user=guest&passwd=guest&" + ("collection=" + collection_id + ["&"])*
```

For example:

```
z39.50s://pdq.org/zclient?user=guest&passwd=guest&  
collection="CID_erbe_001"&collection="CID_erbe_002&  
collection="CID_erbe_003&collection="CID_erbe_004&  
collection="CID_erbe_005"
```

When the user selects this link, a CIP Client must begin a session initialized for the guest user, and then run a collection search using the specified collection IDs. The search will use the broadest values for time range and geographic area. The collection descriptors will be listed in the CIP Client. This system feature will enable bringing new users into the system.

4.7 Linking to Guides from Data Collections

The ISA enters associated guide URLs into the RelatedItemID field of the collection descriptors in the site's Collections Database with the Collection Management Tool. CIP Client users can select these guide URLs to perform guide retrieval. The guide URLs are in the form of the GetVirtualDocument IGP message with the "&ICS_CLIENT" string at the end. When the guide URL is selected, the ICS Guide Server at the appropriate site is contacted with the GetVirtualDocument message. The guide document is modified by the Guide Translator and returned to the user's ICS Client for display.

4.8 Adding Guides to the System

The process of adding new guide documents to the ICS Guide System is detailed in Appendix B.

APPENDIX A. Generating Virtual Documents

The Guide Translator converts guide documents from their on-line form to the virtual documents presented to users and indexing engines. As was specified in section 3.3.2.1, the primary software component that does this conversion is called *icsdoc*.

Assumptions

We assume two things about the guide documents with regard to *icsdoc*.

- They are accessible to *icsdoc*.
- They are in HTML (or plain text).

As was stated in section 3.4.4, guide documents, if the guides are not in HTML, a site-specific additional software layer is required to convert the guides on-the-fly. For this discussion of virtual document generation we will continue with these two assumptions.

Summary

The primary purpose of generating the virtual documents is to embed into all ICS guide documents the Guide Attributes and Keywords and links to data collections. The embedded attributes are what enable fielded searches using uniform specification of guide attributes. There are two methods for determining the Guide Attributes for each guide document. First, the authors of the guides can specify values for these attributes in invisible Meta tags in their guides. Second, when the guides are ingested into the system, the ICS Site Administrator (ISA) enters values for mandatory Guide Attributes (see Table 4). To reduce the burden on the ISA, *icsdoc* will attempt to find attribute/value pairs in the guide documents during ingest. It will map any site specific attributes to ICS Guide Attributes. The Guide Attribute/Keyword pairs will be stored in the Guide Metadata Database during ingest. After this point, *icsdoc* merely needs to look the guide up in the Guide Metadata Database to get the Guide Attribute/Keyword pairs to insert into the guide.

Meta Tags

Invisible tags are a method in HTML for storing information in a document that is not displayed to a user in a browser. There are two methods of doing this:

Any text placed inside a <META> tag will not be displayed. For example:

```
<META This text won't be shown >
```

Any text placed inside a comment tag will not be displayed. For example:

```
<!-- This also will not be displayed -->
```

Inside the guide documents, the authors can flag attribute and values in their agency schemes via these invisible tags. If there are no such flags, software and the ISA will take care of this requirement during guide ingest, without physically modifying the guide (see Appendix B).

The method used in EOSDIS V0 is to place an invisible HTML comment section near the end of the HTML body. The comments are not visible to the user when a Web browser displays the guide. The format of the comments in the section is:

```
<!-- ATTRIBUTE=VALUE -->
```

For example:

```
<!-- Dataset=CZCS_LEVEL_1 -->
```

Another method, which would enable the developers to use simpler generic software, would be to use the invisible HTML META tag:

```
<META name=ATTRIBUTE value=VALUE>
```

For example:

```
<META name="date" value="June 1997">
```

This second method will be used in the virtual documents generated in the ICS Guide System. These tags will be placed in the <HEAD> . . . </HEAD> section, and are not visible to browsers.

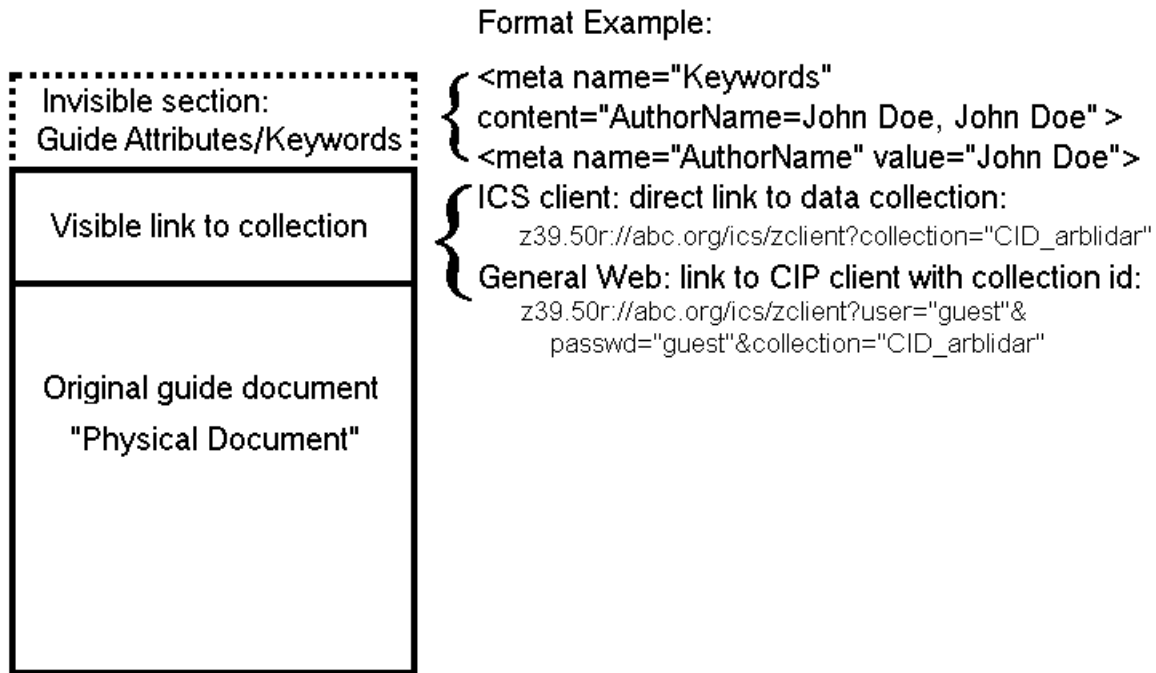
icsdoc Guide Processing

When *icsdoc* is run with the path information to a guide, *icsdoc* will generate a virtual document and return it. Virtual documents are HTML documents generated on-the-fly by software, from the physical documents. They contain a reproduction of the physical documents HTML, with two additions:

- An invisible Meta tag section containing Guide Attribute/Keyword pairs identifying each guide.
- A visible link to data collections related to the guide document. If there are multiple collections related to the guide, then multiple links will be displayed and one link for all the related collections. The HTML of the link will be in one of two formats:
 - If the virtual document is being sent to an ICS Client, the link will be a direct link to the collection.
 - If the virtual document is being sent to a general Web browser, the link will be to a CIP Client, requesting a guest access, and with the collection information. This will bring people into the ICS system while giving them access to the data collection.

The virtual document is shown schematically in Figure 8.

Figure 8. Virtual Document



icsdoc needs the following things at the agency site in order for the ICS Guide System to work.

- Direct access to the guide documents in HTML (or plain text) format. They may actually be in a database at the site, or there may simply be a file of URLs or URNs to the guides.

- Guide Metadata Database. This database contains an entry for each guide accessed through the agency site. For each entry, this contains values for all of the mandatory Guide Attributes and any optional ones found in the guide or specified during ingest.
- Attribute Mapping File. This file contains a mapping from agency attributes to Guide Attributes.
- Collections Mapping File. This file contains an entry for each guide accessed at the agency site. For each entry there is a list of related collection identifications.

When *icsdoc* is run with the path information to a guide, *icsdoc* will generate the virtual document through the following steps.

1. *icsdoc* looks up the guide in the Guide Metadata Database. If *icsdoc* is being called during the guide ingest process (see Appendix B), then *icsdoc* will not find an entry. If *icsdoc* is being called during normal guide retrieval (or indexing) it will find an entry. If an entry is found, *icsdoc* extracts the Guide Attributes/Keywords from the entry.
2. *icsdoc* reads the guide. This may be from a database or URL (or URN).
3. If the guide **was not** found in the Guide Metadata Database in step 1 (ingest phase), then the following steps are performed:
 - i. *icsdoc* will scan through the guide for site-specific metadata (attributes and keywords). If the physical documents are annotated with META tags as described above, then this module is quite simple to implement. Alternative schemes may be simple to handle if used consistently across the entire site. During this step, *icsdoc* will extract all of the agency attributes and keywords from the Meta tags or comment lines of HTML in the guide.
 - ii. *icsdoc* will read the Attribute Mapping File. *icsdoc* will attempt to map each of the found agency attributes to Guide Attributes. It may also attempt to map each of the found agency keywords to Guide Keywords, if it is decided that this mapping is to be done. The mappings can be done as simple look-up tables.
4. If, on the other hand, the guide **was** found in the Metadata Database in step 1, then *icsdoc* simply gets the list of Guide Attributes and Keywords from the database for the guide. Note that this much faster method is used during normal guide retrieval.
5. *icsdoc* will generate invisible Meta tags for the Guide Attributes/Keywords and insert them into the virtual document's header. The format will be:

```
<meta name="Keywords"
      content="attribute1=keyword1, keyword1,
              attribute2=keyword2, keyword2,
              attribute3=keyword3, keyword3,
              . . .">
<meta name="attribute1" value="keyword1">
<meta name="attribute2" value="keyword2">
<meta name="attribute3" value="keyword3">
. . .
```

Note that if the physical document already contains a `<meta name="Keywords">` then the inserted Meta tag above will have to contain whatever was in the original content section. Remember that these strings are not visible in the user's browser, but are used by the indexing engines to index the guide.

Note that each of the keywords is placed three times to allow proper indexing by general Web index engines like Alta Vista, and by our ICS index engine. This is why: generally at the top of documents index engines look for a Meta tag with the specific name "Keywords". Note that this has nothing to do with CIP, ECS, or anybody

else's meaning of "Keywords". These keywords are used for quick indexing of the documents by general search engines. For example, the V0 Web Gateway's home page has a Meta tag that begins:

```
<meta name="Keywords"
      content="aerosols, air quality, atmosphere, . . .">
```

In order to add a Guide Attribute/Keyword pair, such as "AuthorName=John Doe", the string is added to the CONTENT list. However, the user may enter two different strings into a general search engine to find the guide. They may enter "AuthorName=John Doe", or they may enter "John Doe". In order to catch both cases, both strings can be added to the CONTENT list. Remember that with search engines, segments of strings don't match, e.g. "book" will not match "booking". In addition to the two cases, in order to use a fielded index search from an ICS Client, we need an additional Meta tag for the attribute/keyword pair. Thus, if just this one example pair were inserted, the HTML is:

```
<meta name="Keywords"
      content="AuthorName=John Doe, John Doe">
```

```
<meta name="AuthorName" value="John Doe">
```

6. *icsdoc* looks up the guide in the Collection Mapping File. It will generate one or more links to the collection(s) that are related to the guide. As was mentioned above, the file contains links to collections for each guide. For example, one guide and associated collection IDs might be listed like this:

```
erbe.html
CID_erbe_001
CID_erbe_002
CID_erbe_003
```

Then, for the given guide, *icsdoc* will insert **visible** links to these collections. The link could appear as a button or hyperlink text. *icsdoc* inserts the links in one of two ways, depending on whether it was contacted by an ICS Client or not. It knows whether it was or not because the GetVirtualDocument message will have the string "&ICS_CLIENT" appended to it if the message is sent from an ICS Client. The formats of the HTML for the links are:

- If the guide is being sent to an ICS Client, the link will be the GetCollection IGP message. The HTML would look like this (ignore line wrapping):

```
z39.50s://pdq.org/zclient?collection="CID_erbe_001"&
collection="CID_erbe_002"&collection="CID_erbe_003"
```

- If the guide is being sent to a general Web browser, the link will be to a CIP Client, with the collection information. This would bring people into the ICS system while giving them access to the data collection. This HTML would look like (ignore line wrapping):

```
z39.50s://pdq.org/zclient?user="guest"&passwd="guest"&
collection="CID_erbe_001"& collection="CID_erbe_002"&
collection="CID_erbe_003"
```

This is a standard CGI method of passing information, parameters, to a CGI script. These links provide access from guide documents to data collections through the ICS. If a person finds a guide through a general Web search engine, they will link to a new CIP session pre-loaded with the corresponding collection identification. The CIP client will probably do a broad collection search using the collection IDs. Note that the CIP session is started as a guest user.

7. *icsdoc* now streams the virtual document it created to the ICS Guide Server, which passes it back to whatever contacted it through the httpd, via HTTP.

See Figure 4 in section 4.2 for an illustration of guide retrieval with a general browser. See Figure 9 in Appendix B for an illustration of guide retrieval during the ingest process.

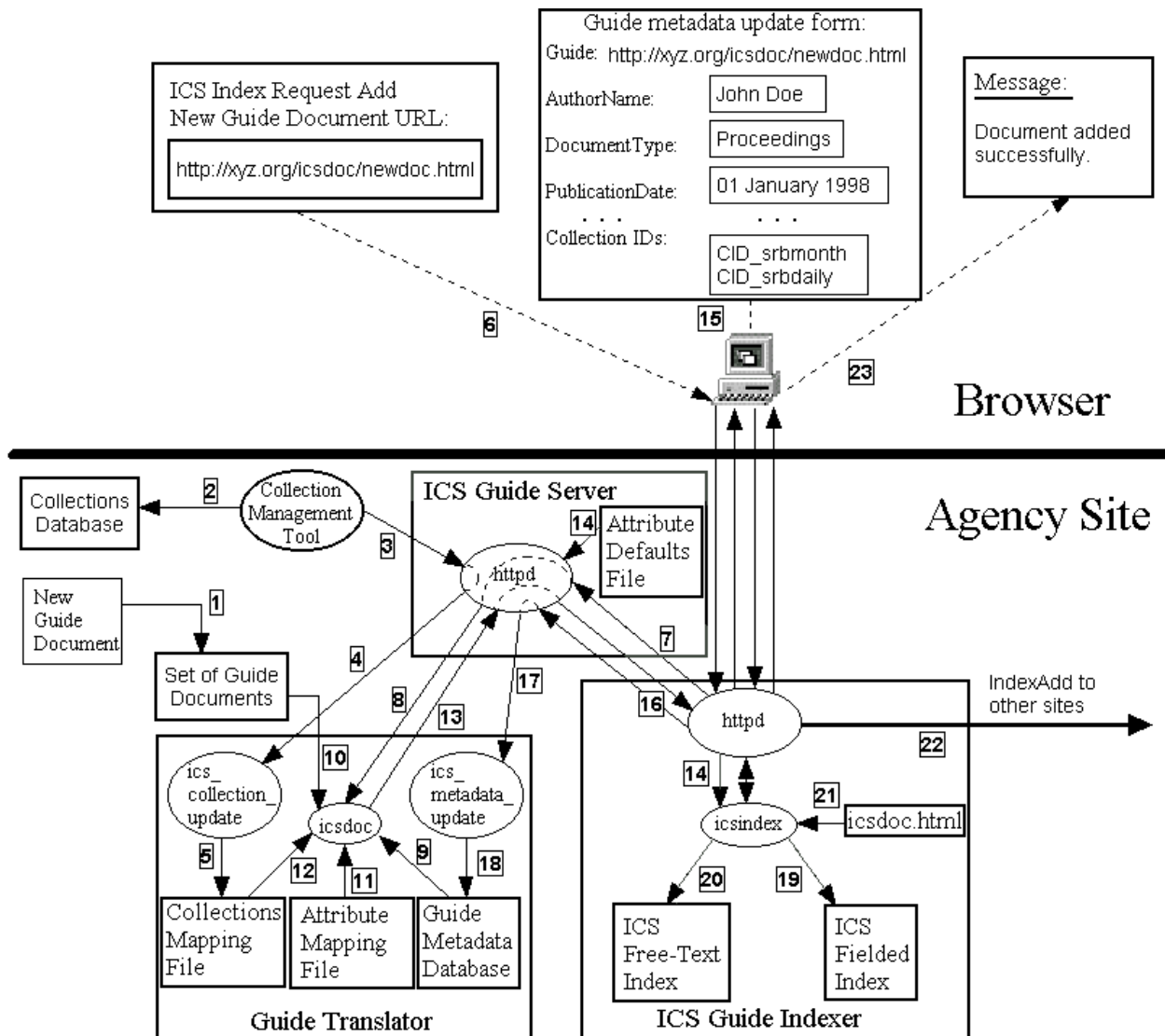
APPENDIX B. Adding New Guide Documents to System

Adding new guide documents is a critical part of the system. The method was described in an abbreviated form in section 3.3.3.2, icsindex Add Mode. We will cover the details of all the steps here. The critical factors to consider are:

- The new guide may not have values for all mandatory Guide Attributes. For fielded searches, these values must be found in the virtual documents.
- The Collection Database will have to be updated with guide URLs.
- The Collection Mapping File will have to have the collection IDs for the guide.
- Indexes at other agency sites will need to be updated to match the modified one.

The complete set of steps required for adding guide documents to the system follow. While there are many steps, most of them are simple. The software performs most of these steps with a few simple steps for the ICS Site Administrator (ISA). The following complex diagram shows all steps and the parts of the system involved. Reference it while studying the steps. You may also want to refer to the abbreviated version of the diagram in section 3.3.3.2, Figure 2.

Figure 9. All Steps for Adding a New Guide to the System



To help you locate the steps on the diagram, preceding each step will be the general location.

1. (left) The ICS Site Administrator (ISA) adds the guide to the site's guide documents. This may involve simply putting the file into a special directory, or could require adding it to a database, or could simply involve adding the guide URL (or URN) to a specified guide list file. The guide documents do not need to be stored at the agency site. This step just makes the guide accessible by the site's *icsdoc* script.
2. (left) The ISA adds the new guide URL (GetVirtualDocument IGP message) to the RelatedItemID field of related collection descriptors in the Collections Database using the Collection Management Tool.
3. (left) The Collection Management Tool contacts the ICS Guide Server with the UpdateCollections IGP message, containing the guide URL and collection ID. The connection is to the httpd in the server. For example:

```
http://abc.org/ics_collection_update?
guide="http://abc.org/icsdoc/localpath/newdoc.html"&
colletion="CID_AM1_0001"
```
4. (left) The server runs *ics_collection_update* program in the Guide Translator with the guide URL and collection ID.
5. (bottom left) *ics_collection_update* updates the Collections Mapping File. If there is already an entry for this guide URL, the new collection ID is added to the list for the URL. If not, a new entry is created.
6. (top left) The ISA starts the ICS Guide Indexer program *icsindex* with a Web browser. This presents a form to the ISA asking for the URL of the new guide. The URL entered must actually be the GetVirtualDocument message. For example:

```
http://abc.org/icsdoc/localpath/newdoc.html
```

The browser sends the IndexNew message to the ICS Guide Indexer. For example:

```
http://abc.org/icsindex?mode=new&
guide="http://abc.org/icsdoc/localpath/newdoc.html"
```
7. (center) The httpd in the ICS Guide Indexer runs *icsindex* with the mode and guide parameters. *icsindex* then contacts the ICS Guide Server using the guide URL through the httpd.
8. (left) The ICS Guide Server runs *icsdoc* in the Guide Translator with the path information to the new guide.
9. (bottom left) *icsdoc* looks for an entry for the guide in the Guide Metadata Database. At this point the guide will NOT be found there. An entry will be added there later, but *icsdoc* always looks there for Guide Attributes for the requested guide. (See Appendix A).
10. (left) *icsdoc* reads the new guide from the site's guide documents (or access point).
11. (bottom left) *icsdoc* reads the Attribute Mapping File. This contains the map from agency attributes to Guide Attributes (and, possibly, agency keywords to Guide Keywords).
12. (bottom left) *icsdoc* looks up the guide in the Collections Mapping File. From this, *icsdoc* finds collection IDs to which this guide is related.
13. (center) *icsdoc* creates the virtual document as outlined in Appendix A. Note that at this point the virtual document may be missing some Guide Attributes in its Meta tags, but has the collection link information inserted. The virtual document is sent through the ICS Guide Server httpd, to the ICS Guide Indexer httpd, and into *icsindex* for analysis.
14. (center) *icsindex* reads the Attribute Defaults File from the ICS Guide Server. This contains a site dependent default value for each of the mandatory Guide Attributes (see Table 4). Having this file frees the ISA from having to re-enter constant information for every guide added.
15. (top center) *icsindex* generates an HTML "add guide" form and sends it via the httpd to the ISA's browser. This form displays all of the attributes with values set by either being found in the guide and mapped by *icsdoc*, or found in the defaults file in step 14, or blank if not specified in either location. It also displays the collection IDs found in step 12. The ISA now has the opportunity and responsibility to fill in the form with any missing values, or modifying defaulted values, for the new guide. Once everything is acceptable and there is a value for every mandatory attribute, he can click a "Submit" button. The contents of the form are submitted by CGI and HTTP to *icsindex* through the httpd in the ICS Guide Indexer. This is the IndexValues IGP message.
16. (center) *icsindex* sends the UpdateMetadata message through the httpd to the ICS Guide Server.
17. (center) The ICS Guide Server runs *ics_metadata_update* in the Guide Translator with the parameters of the UpdateMetadata message.
18. (bottom left) *ics_metadata_update* updates the Guide Metadata Database with values for all entered Guide Attributes, keyed by the new guide's URL.
19. (bottom right) *icsindex* adds or updates entries in the fielded index database for the new guide.
20. (bottom center) *icsindex* adds or updates entries in the free-text index database for the new guide.

21. (bottom right) *icsindex* reads the list of all ICS agency sites from the *icsdoc.html* file.
22. (right) *icsindex* sends the IndexAdd message through the *httpd* to the ICS Guide Indexer at each of the other agency sites. This instructs them to index the new guide. Note that when *icsindex* at the other sites retrieve the guide from this site for indexing, this site's *icsdoc* will now find all the mandatory Attributes in the Guide Metadata Database. The returned virtual document will have everything needed for *icsindex* at the other sites to update their fielded and free-text databases.
23. (top right) Finally, notification of success is sent or shown to the ISA.

These are the steps required to add or update a guide in the ICS Guide System. *icsindex* shall also have a similar feature to remove a guide and notify other sites. *icsindex* will also have the previously mentioned "batch mode". In batch mode, *icsindex* re-indexes all the guide documents at all of the ICS agency sites. This is the function mentioned in section 3.3.3.1. This general re-indexing could take place once a week, or once a month.

APPENDIX C. Indexing Guide Documents

In section 3.3.3.1 icsindex Batch Mode, we introduced the concept of how the Web indexing engines traverse Web pages for indexing. Here we reiterate the explanation with greater detail and examples for clarification.

Note: the following explains the process of what the Web indexing engines will do. This is NOT what happens when a user searches for or retrieves guide documents. The process happens whenever the Web indexing engines are run to update their indexes. The indexing must be done at least once prior to any searching or retrieving is possible.

General Web search/indexing engines such as AltaVista, Lycos, Yahoo, etc., will perform the following steps in the exact manner as *icsindex*. The difference is that they index all documents on the Web and create only free-text indexes. *icsindex* only creates an index of ICS guide documents and creates both a free-text index and specialized fielded index. The general Web search engines will be notified of the location of one instance of the top-level, site listing Web page. *icsindex* also starts there. For example, let's say one can be found at this (fake) URL:

```
http://cip.org/icsdoc.html
```

Recall that this page contains links to the ICS Guide Servers at all of the agency sites, one per line. For example: this document could look like this (where . . . implies more entries):

```
http://abc.org/icsdoc
http://def.org/ics/icsdoc
http://ghi.org/cip/scripts/icsdoc
. . .
http://pdq.gov/icsdocserver/icsdoc
http://xyz.org/icsdoc
```

When an indexing engine contacts the site listing page, the engine will simply traverse the ICS guide document tree.

1. First, the engine reads in the page above.
2. Next, the engine contacts each of the links on the page; like so:

```
http://abc.org/icsdoc
http://def.org/ics/icsdoc
. . .
http://xyz.org/icsdoc
```

When it contacts these links, the GetDirectory IGP message is sent to the ICS Guide Server. That's because the links are GetDirectory messages (see Table 5b).

3. The ICS Guide Server runs *icsdoc* in the Guide Translator.
4. Since there is no path information following "icsdoc" in the link, *icsdoc* returns a "directory" page (see section 3.3.2.1.1). These will actually be virtual HTML documents containing links to all of the guide documents reachable from those sites through the ICS Guide Server. For example:

```
http://abc.org/icsdoc/docs/erbe.html
http://abc.org/icsdoc/docs/fire.html
http://abc.org/icsdoc/docs/isccp.html
http://abc.org/icsdoc/docs/nvap_tovs.html
. . .
http://def.org/ics/icsdoc/srb_monthly.html
http://def.org/ics/icsdoc/srb_daily.html
http://def.org/ics/icsdoc/topex_poseiden.html
. . .
http://xyz.org/icsdoc/landsat7.html
http://xyz.org/icsdoc/nalc.html
```

Recall that these guide URLs are actually GetVirtualDocument IGP messages.

5. The indexing engine contacts these links. The GetVirtualDocument messages are sent to the ICS Guide Servers.
6. The servers run *icsdoc* in the Guide Translator with the guide path information. *icsdoc* will run for each guide, modifying it by inserting Meta tags containing Guide Attribute/Keyword pairs and links to related data collections. This creates a virtual document.
7. The virtual document is returned to the indexing engine through the httpd.
8. The engine scans the guide for words and adds the guide URL to its database, with the words as keys. The guide is not stored at the index, just the URL.

In this way, the indexing engines access and process every virtual ICS guide document at all ICS sites. The guides will thus be indexed and can be searched for and retrieved by users of general Web search engines and the ICS Client. Since the virtual documents contain links to CIP collections, new people can be brought into the ICS system and learn of its features when they find an ICS guide through a general Web search engine.

As mentioned in step 6, *icsdoc* will insert Guide Attribute/Keyword pairs invisibly into the guide documents it returns. For example, one invisible pair could be inserted like so (see Appendix A):

```
<meta name="Keywords"
      content="AuthorName=John Doe, John Doe">
```

Because of these tags, the fielded index generated by the ICS Guide Indexer will contain the Guide Attribute/Keyword pairs. An ICS Client user can therefore find guides by specifying values for the Guide Attributes in a fielded search.

Figure 10 shows how Alta Vista connects to all agency sites by finding them in the *icsdoc.html* document. *icsdoc* runs at each site, first returning a list of all guide documents reachable from the site. *icsdoc* then runs again, returning each guide modified with invisible Guide Attributes tags, and visible links to collections. Alta Vista then adds words, from the guide documents, as keys and guide URLs to the Alta Vista free-text index.

Figure 10. General Web Indexing and Agency Sites

