

ISBN 0-000000-00-0
ISSN 0000-000X

NORTH PACIFIC MARINE SCIENCE ORGANIZATION

PICES TECHNICAL REPORT No. 1 2007



Metadata Federation of PICES Member Countries



PICES

**PICES Technical Report No. 1
2007**

Metadata Federation of PICES Member Countries

Edited by

Bernard A. Megrey, S. Allen Macklin,
Kimberly Bahl, and P. Daniel Klawitter

1 February 2007

Secretariat / Publisher

North Pacific Marine Science Organization (PICES)

c/o Institute of Ocean Sciences, P.O. Box 6000, Sidney, B.C., Canada. V8L 4B2

E-mail: secretariat@pices.int Home Page: <http://www.pices.int>

ISSN 1993-4661 ISBN

ISBN 1-897176-06-6 (web version)

TABLE OF CONTENTS

Table of Contents	iii
Executive Summary	vii
1. Introduction	1
1.1. Data sharing, international boundaries and large marine ecosystems	1
2. Objectives	3
3. Background	5
3.1. North Pacific Ecosystem Metadatabase	5
3.2. First federation effort: NPEM and the Korea Oceanographic Data Center	7
3.3. Continuing effort: Adding Japan’s Marine Information Research Center	9
4. Metadata Standards	11
4.1. Directory Interchange Format	11
4.2. Ecological Metadata Language	12
4.3. Dublin Core	12
4.3.1. Elements of DC	12
4.4. Federal Geographic Data Committee	13
4.5. The ISO 19115 Metadata Standard	13
4.6. Metadata stylesheets	13
4.7. Crosswalks	14
4.8. Tools for creating metadata	14
5. Communication Protocols	15
5.1. Z39.50	15
5.1.1. What does Z39.50 do?	15
5.1.2. Isite	15
6. Clearinghouses	17
7. Methodology	19
7.1. FGDC metadata	19
7.1.1. Main sections	21
7.1.2. Supporting sections	21
7.1.3. Metadata validation	24
7.2. Getting a copy of Isite	24
7.3. NSDI Clearinghouse	24
8. Server Configuration and Technical Issues	25
8.1. Hardware recommendations	25
8.2. Operating system – Red Hat Linux Fedora	25

8.3.	Web services – Apache HTTP Server version 2.2.3	25
8.4.	Create and validate FGDC-compliant Metadata in XML format	25
8.5.	Obtaining, installing and configuring Isite for UNIX/Linux	26
8.5.1.	Download the appropriate Isite software	26
8.5.2.	Untar the file	27
8.5.3.	Name your database	27
8.5.4.	The <code>zserver.ini</code> file	27
8.5.5.	The <code>sapi.ini</code> file	27
8.5.6.	Indexing metadata	28
8.5.7.	Start the Clearinghouse Server process	29
8.5.8.	Testing the <code>zserver</code> installation	29
8.6.	Registering with NSDI Clearinghouse	29
8.7.	Security issues	38
9.	Search Tutorial and Examples	41
9.1.	Legacy NSDI Clearinghouse search interface	41
9.2.	New GeoNetwork search interface	47
10.	Challenges	51
11.	Emerging Standards	53
12.	Future Activity	55
13.	Acknowledgments	57
14.	References	59
15.	Acronyms	61
16.	Appendices	63
16.1.	KODC-NPEM meeting agendas and minutes	63
16.1.1.	Seattle meeting agenda, August 22–23, 2005	63
16.1.2.	Seattle meeting minutes, August 22–23, 2005	63
16.1.3.	Busan meeting agenda, October 10–11, 2005	65
16.1.4.	Busan meeting minutes, October 10–11, 2005	66
16.2.	MIRC-NPEM meeting agendas and minutes	67
16.2.1.	Seattle Meeting agenda, August 14–15, 2006	67
16.2.2.	Seattle meeting minutes, August 14–15, 2006	67
16.2.3.	Tokyo meeting agenda, October 19–20, 2006	69
16.2.4.	Tokyo, meeting minutes, October 19–20, 2006	69
16.3.	XML stylesheet conversion crosswalks	71
16.3.1.	FGDCI to DIF stylesheet converter	71
16.3.2.	DIF to FGDCI stylesheet converter	85
16.3.3.	String-modified stylesheet	103
16.4.	FGDC Metadata Standard	126
16.4.1.	Overall structure	126
16.4.2.	Section 1: Identification information	127
16.4.3.	Section 2: Data quality information	128
16.4.4.	Section 3: Spatial data organization information	129

16.4.5. Section 4: Spatial reference information	130
16.4.6. Section 5: Entity and attribute information	131
16.4.7. Section 6: Distribution information.....	132
16.4.8. Section 7: Metadata reference information	133
16.4.9. Sections 8, 9 and 10: Citation information, time period information, and contact information	134
16.5. Images of the Isite server directory structure and the files contained in each subdirectory after Isite installation	135
16.6. Listing of NPEM's Isite configuration files	139
16.6.1. zserver.ini	139
16.6.2. sapi.ini	142
16.7. Java program to extract records from the NPEM metadatabase and write one XML file for each record.....	143
16.8. Java program to execute the metadata extraction program	151

EXECUTIVE SUMMARY

This report describes the history of the PICES Metadata Federation project, the planning and activities that took place to establish four federated PICES member countries, and the ongoing challenges and future developments. The greater intention of this project is to federate the marine metadata holdings of all PICES member countries. Recently Canada and the People's Republic of China have expressed interest in joining the PICES Federation.

The main body of the report includes a detailed description of specific technical instructions and guidance for anyone wishing to join the Federation, including hardware requirements, software requirements and installation, and steps required for configuring and registering a node at the National Spatial Data Infrastructure Clearinghouse.

We conclude with some comments on future directions and up-to-date information on the

emerging changes expected to take place in the Clearinghouse interface.

Perhaps the most important part of the document is that the Federation Clearinghouse technology is quickly evolving to new and improved standards and capabilities (See Sections 9 and 11 for more details.). As a consequence, PICES chose to publish this living document as an on-line PDF PICES Technical Report with the goal to keep it as current as possible as new Federation partnerships make themselves available, as well as new developments in communication and search technologies come on-line. Thus, expect to see that this report will be updated frequently as the Clearinghouse search interface moves from the legacy interface (NSDI legacy interface) to the new GeoNetwork interface. Please refer to the revision date on the title page to make sure you have the most recent copy.

1. INTRODUCTION

1.1. Data sharing, international boundaries and large marine ecosystems

Marine science and marine management organizations have a need to know, first-hand and quickly, pertinent details of marine ecosystems to facilitate planning and management. The North Pacific Marine Science Organization (PICES), for example, produces reports on the status of North Pacific marine ecosystems that are intended to periodically review and summarize the status and trends of the marine ecosystems in the North Pacific, and to consider the factors that are causing, or are expected to cause, change in the near future. The first report (PICES, 2004) served as a pilot project for what might be possible. This report was based largely on geographic locations and subjects for which time series data or information were readily available. The report also identified locations and subjects where data were collected but were not yet available.

For the most part, marine ecosystem data are

contained in archives of national ocean data centers, academic institutions and other agencies. Some of these bodies serve some or all of their archived information through the World-Wide Web. Many data, however, remain hidden from public search and use. Even discovering the data that are available on line can be an onerous task. For example, obtaining information on the East China Sea requires access, at least, to marine data of the People's Republic of China, Republic of Korea and Japan. The situation is repeated for many other marginal seas of the North Pacific Ocean.

More and more, ecosystem management focuses on Large Marine Ecosystems (LMEs, Fig. 1.1). LMEs are regions of ocean space encompassing coastal areas from river basins and estuaries to the seaward boundaries of continental shelves and the outer margins of the major current systems. They are relatively large regions on the order of 200,000 km² or greater, characterized by distinct (1) bathymetry, (2) hydrography, (3) productivity,

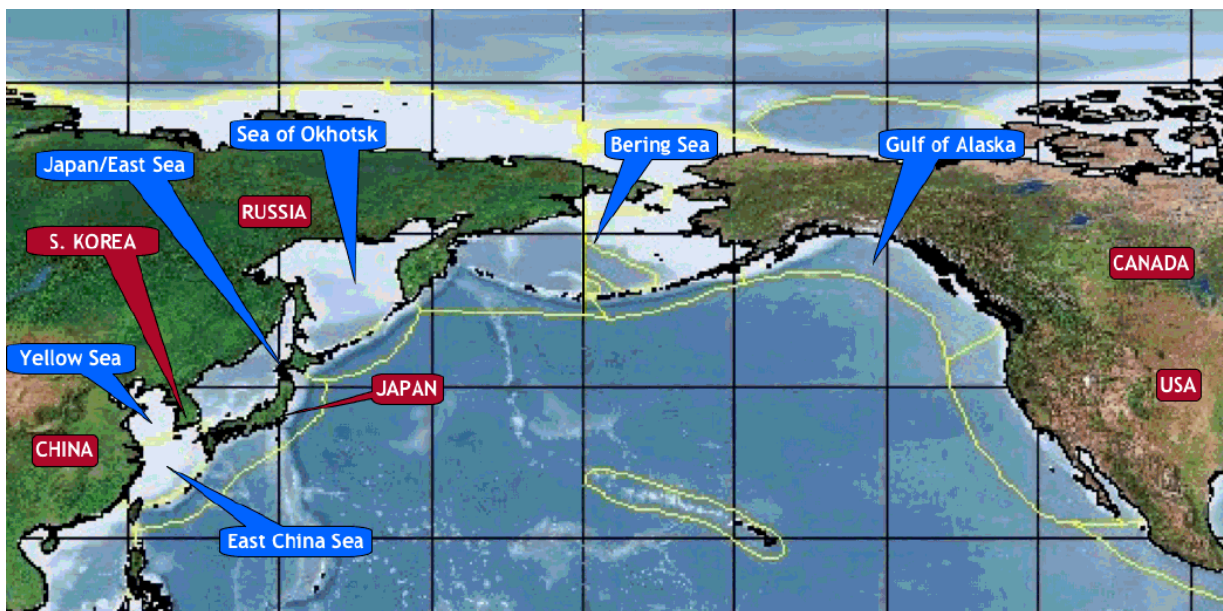


Fig. 1.1 The North Pacific Ocean and its marginal seas (blue labels), PICES member nations (red labels), and Large Marine Ecosystems (yellow boundaries).

and (4) trophically dependent populations. On a global scale, 64 LMEs produce 95% of the world's annual marine fishery biomass. Within their waters, most of the global ocean pollution, overexploitation, and coastal habitat alteration occur. Yet, vital information on these processes is not centrally available, nor even easily available, for public scrutiny, let alone international management efforts.

The PICES Technical Committee on Data Exchange (TCODE) is working to address this problem. TCODE has among its terms of reference: (1) *to identify the data management requirements of PICES* and (2) *to develop strategic plans to meet these requirements*. TCODE has established the PICES Long Term Time Series and endorsed the North Pacific Ecosystem Metadatabase (NPEM). The latter is the precursor to the project documented in this report, and NPEM is discussed further in section 3.1. Part of TCODE's working plan for the last

several years has been the PICES Metadata Federation project. This project attempts to alleviate the shortcomings of international ecosystem data sharing by creating a "federation" of marine ecosystem data servers, using standard data cataloging and communication protocols. This is not new technology. Similar systems have been in existence for many years, used mainly by library and museum systems. Scientific clearinghouses exist, as well, and the method described in this report makes use of such a clearinghouse.

The remainder of this report presents the objectives of the project, its history, a general discussion of metadata and communication standards, and the specific application selected to support the PICES metadata federation. The report concludes with a discussion of challenges to implementation and presentation of future activities, as this project is not yet completed.

2. OBJECTIVES

The objectives of the PICES Metadata Federation project are to create standardized metadata descriptions of national, institutional and agency databases and to serve those descriptions in a World-Wide Web-based, one-stop environment with search and delivery capability. Such a federated or distributed system has many benefits.

Federation is a process of joining for mutual benefit. For example, suppose Provider 1 produces bread, and Provider 2 produces butter. In a non-federated system, a consumer wanting bread would have to get it from Provider 1, and butter

would only be available from Provider 2. However, if Providers 1 and 2 are willing and able to cooperate, each provider can maximize the distribution of its own product by also offering it through the other provider. This is a federation.

This federation promotes efficiency for the provider and the consumer. Each provider effectively boosts its product line by having available more products without actually having to produce them. The consumer benefits by being able to locate more products without having to know more providers.

3. BACKGROUND

The foundation of the PICES Metadata Federation effort was the creation and development of the North Pacific Ecosystem Metadatabase (NPEM, originally called the Bering Sea Ecosystem Biophysical Metadatabase). When faced with the decision of how to enlarge the contents of the NPEM, the directors sought a solution that would allow NPEM users access to new metadata without having to host those metadata in the NPEM. Prior interaction with the founders of Alaska's Cook Inlet Information Management and Monitoring System (CIMMS) had demonstrated the value of a distributed system.

3.1. North Pacific Ecosystem Metadatabase

The NPEM (Macklin and Megrey 2004, Fig. 3.1, <http://www.pmel.noaa.gov/np/mdb>) is an Internet utility to aid in the understanding, management,

stewardship and utilization of North Pacific Ocean ecosystems. The utility is a browsable, searchable, on-line inventory of data and other information. NPEM is dynamic, *i.e.*, it undergoes continuous development to keep its contents up to date so that users can access current information from which to make decisions. NPEM's goal is to provide free and open access to information that ordinarily would be unavailable to researchers. This practice enables collaborations between investigators and makes the exchange and use of marine science data more efficient.

NPEM began in 1996 as the Bering Sea Ecosystem Biophysical Metadatabase (Megrey and Macklin, 1998) with 3-year support from the National Oceanic and Atmospheric Administration (NOAA) Environmental Services Data Information Management (ESDIM). The

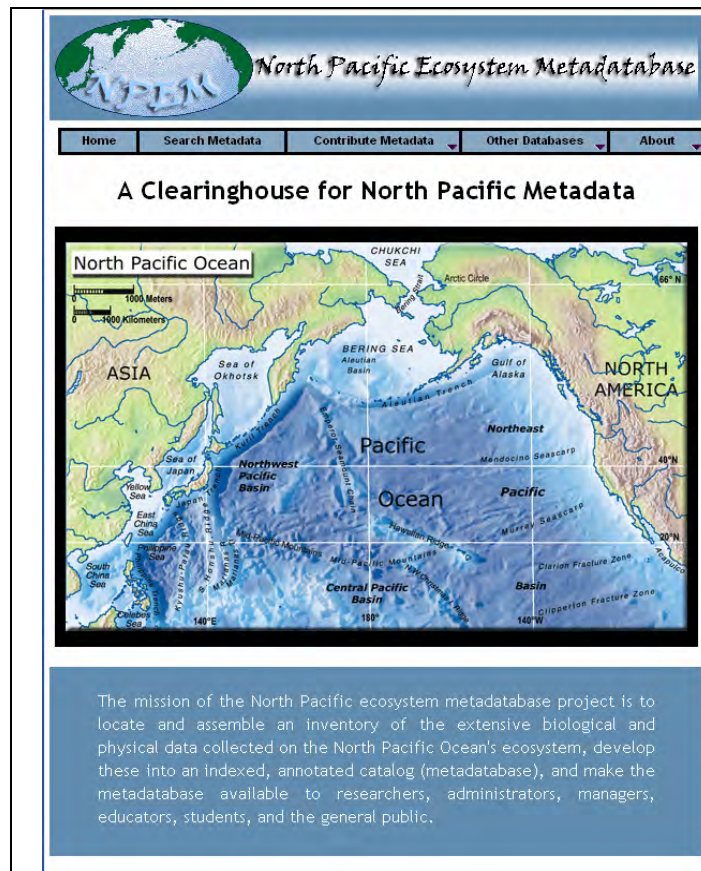


Fig. 3.1 Home page of the North Pacific Ecosystem Metadatabase (<http://www.pmel.noaa.gov/np/mdb>).

metadatabase addressed a deficiency identified by the National Research Council (1996). In its report on the Bering Sea ecosystem, the council concluded that a directory of data and information sources relevant to the Bering Sea, cataloged in one place, was a critical need. Furthermore, the council cited the lack of such a database as the one major impediment to studying the Bering Sea. It was clear that scientists had little appreciation of metadata or their importance at that time.

We first developed a schema from the minimum set of Federal Geographic Data Committee (FGDC) descriptive elements, designed the database in Microsoft Access, and procured a Windows server as a public interface using Active Server Pages (ASP) scripts. In mid-1997, we published the first call for metadata. We solicited information from scientists, advertised in science newsletters, made national and international presentations and, through PICES, developed contacts with Canadian, Chinese, Japanese, South Korean, and Russian marine science institutes. We educated the scientific community on the importance of metadata and indicated the benefits that would accrue to scientists and science as a result of proper metadata specification. We mailed thousands of metadata entry forms to scientists, requesting their metadata. From these efforts, the metadatabase grew to 70 records within a few months. By the end of the third year of funding, there were more than 1000 records populating the metadatabase. Also in those first years, the metadatabase earned support and endorsement from Fisheries–Oceanography Coordinated Investigations (FOCI), the North Pacific Marine Science Organization (PICES), the *Exxon Valdez* Oil Spill Trustee Council and the North Pacific Marine Research Program, for which the metadatabase was granted funds to be the official program metadata repository.

In March 2001, the metadatabase directors attended a PICES-sponsored, international workshop on “Impact of climate variability on observation and prediction of ecosystem and biodiversity changes in the North Pacific”. Workshop participants from Canada, People’s Republic of China, Japan, Republic of Korea, Russia, the United Kingdom, the United States, and 11 international science organizations

nominated existing time series and predictions for determining the status of North Pacific ecosystems. Attendees were amazed at the diversity and quantity of the many data series that were brought forward. Data from western Pacific nations has been particularly difficult to identify and obtain, as much of it is known only locally. The nominated time series from all around the North Pacific rim, basin, and marginal seas, have sufficient historical length, accuracy, and likelihood of continuance to be important indicators of climate and climate response. Participants of the workshop recommended that the time series information and scientific contacts identified be recorded and updated in the North Pacific (*i.e.*, Bering Sea) Ecosystem Metadatabase. With this impetus, we again applied to ESDIM, successfully, to expand the Bering Sea Ecosystem Biophysical Metadatabase to the NPEM, and that work began in late 2002. Search and display capabilities have been upgraded with this funding, and the metadatabase is now housed in MySQL and served from a Linux platform.

As of December 2006, the NPEM contains 3921 records referencing physical and biological datasets, model output, museum samples, publications, reports, proposals, atlases, and audio and video programs. The regional distribution of these records is shown in Figure 3.2. Most records are from coastal locations. In terms of metadata density, most contributions pertain to the eastern North Pacific Ocean. We suspect that there has been a similar quantity of research performed in the western North Pacific Ocean, however, results from this research are not as readily available to us. For example, although more than ten Asian institutes have contributed to the metadatabase,



Fig. 3.2 Regional distribution of metadata records.

these records make up less than 11% of the holdings. Holdings span all biological and physical scientific disciplines, including historical and present information ranging from atmosphere to open ocean to inter-tidal areas.

The North Pacific Ocean Theme Page <http://www.pmel.noaa.gov/np/> is the Internet gateway to the metadatabase. The theme page and the metadatabase offer a rich suite of environmental information to scientists, students, teachers, managers, and casual users. Since their inception, both the theme page and the metadatabase have increased in popularity as shown in Figure 3.3. Peaks in user activity correspond to important announcements of availability of research funds or other resources. Note the drop in summertime theme page usage when U.S. public school is not in session. On average, the metadatabase is exercised about 3500 times a month. This represents about 3% of all theme page use. A growth trend from 1997 to about 2002 seems to be leveling.

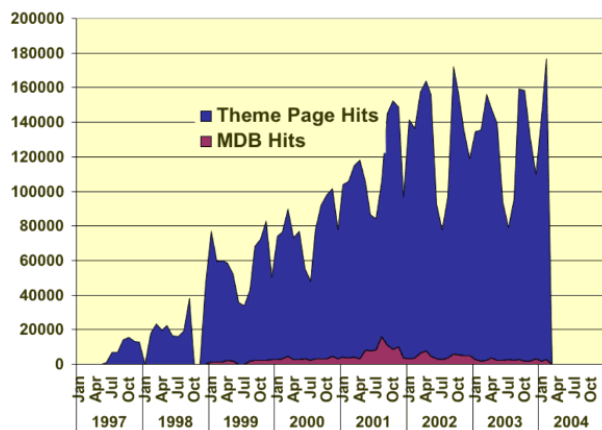


Fig. 3.3 Time history of theme page and metadatabase user activity.

The metadatabase is found through the Theme Page's DATA link or can be accessed directly at <http://www.pmel.noaa.gov/np/mdb/>. Once online, a user can learn about the metadatabase, contribute metadata, or search for metadata by time, location, keyword, country of origin, etc. Spatial searches are accomplished through an interactive map display or by direct specification of latitude and longitude. A user is able to build

compound searches using any two or more search techniques. Search results are returned according to user specification. Presently, the options are to return information as a list of metadata record titles or as dynamically linked icons on a regional map, such as in Figure 3.2. Clicking on a metadata title or clicking on an icon will display the complete metadata record selected. Complete metadata records display to the user all FGDC descriptive elements captured within NPEM. These are contributor, citation, description, status, ecosystem components, keywords, spatial domain, temporal domain, source, and constraints. There is also a link to the PICES Metadata Federation, and that link is exercised using the principles and instructions contained in this report.

We continue to archive all metadata associated with the North Pacific Ocean. In particular, we want to increase holdings of Asian metadata to enrich our references to the western North Pacific Ocean and bordering regions. Toward this end, NPEM established partnership in an existing federation sponsored by FGDC called the National Spatial Data Infrastructure (NSDI) Clearinghouse. The NSDI Clearinghouse requires metadata to be coded using the FGDC standard, and it uses Isite, an instance of the Z39.50 communication protocol, for queries and exchanges. Late in 2003, we began the necessary work, and by September 2005, NPEM became a registered node of the Clearinghouse.

Also late in 2003, we launched plans to implement with other North Pacific marine data centers (e.g., KODC and JODC (Japan Oceanographic Data Center)) "federated searches" or queries that search all metadata sets in separate data locations in a manner that is completely transparent to the user. Using this technique, a user of any of the aforementioned data facilities or of NPEM will be able to search the collection of all subscribing data facilities in a single session.

3.2. First federation effort: NPEM and the Korea Oceanographic Data Center

Representatives (Fig. 3.4) of the Korea Oceanographic Data Center (KODC) and the NOAA-PICES NPEM exploited a communications technique allowing public

Internet search of their combined metadata collections in a single session. The approach requires that each metadata provider establish English-language XML (Extensible Markup Language) metadata records in the FGDC standard format. The XML records are served using the Z39.50 communications protocol. Access is through a metadata clearinghouse that supplies search and delivery scripts to the user. Presently, the federation uses FGDC's National Spatial Data Infrastructure Clearinghouse (<http://clearinghouse3.fgdc.gov/>), in which KODC and NPEM each have registered nodes.



Fig. 3.4 The KODC–NPEM Federation Team outside the National Fisheries Research and Development Institute, Busan, Korea; top row, left to right: Bernard Megrey, Allen Macklin, Dan Klawitter, and Hae-Seok Kang; bottom row, left to right: Joon-Yong Yang, Kyu-Kui Jung, Kimberly Bahl, and Hee-Dong Jeong.

With this understanding of requirements to build a federation in hand, NPEM members approached KODC personnel at the Twelfth Annual PICES meeting in Seoul, Republic of Korea, October 2003, with an invitation to cooperate on a joint federation project. KODC expressed interest in federating with NPEM. Informal communications between parties that year culminated in the submission of a proposal from TCODE to the PICES Science Board at the Thirteenth Annual PICES meeting in Honolulu, U.S.A., the following October. PICES agreed to fund, in part, two meetings of KODC and NPEM principals over the coming year to establish the federation, and to promulgate information to other PICES members about joining the federation.

Using partial support from PICES, KODC and NPEM personnel developed the application over a year, with major progress coming from joint meetings held in Seattle, U.S.A., during August 2005 (Fig. 3.5) and Busan, Republic of Korea, in October 2005 (Fig. 3.6). At the latter meeting, KODC joined NPEM as a registered node of the Clearinghouse. To increase its presence in the Clearinghouse, KODC is expanding the information that it serves through prioritized translation of metadata records from Korean to English and their subsequent conversion to the FGDC standard. To facilitate the conversion of DIF metadata records to FGDC, we obtained Excel Visual Basic routines from NASA (National Aeronautics and Space Administration). The routines require modification to work with KODC's modified Directory Interchange Format (DIF) metadata. A listing of the Seattle and Busan meeting minutes and agendas can be found in Appendix 15.1.



Fig. 3.5 KODC and NPEM representatives working at the Alaska Fisheries Science Center in Seattle in August 2005; left to right: Allen Macklin, Bernard Megrey, Hae-Seok Kang (mostly hidden by laptop), Kyu-Kui Jung, and Kimberly Bahl.



Fig. 3.6 KODC becomes a registered node of the NSDI Clearinghouse; left to right: Joon-Yong Yang watches on the big screen, Dan Klawitter makes the final keystroke, Hae-Seok Kang talks with the server site in Seoul, and Kyu-Kui Jung looks on.

3.3. Continuing effort: Adding Japan's Marine Information Research Center

NPEM personnel recently began working actively with Japan's Marine Information Research Center (MIRC). MIRC provides quality control and value-added product development for the Japan Oceanographic Data Center. The NPEM–MIRC federation is underwritten by FGDC and PICES. A work plan similar to that developed for NPEM–KODC federation is being used.

In August 2006, Dr. Toru Suzuki (MIRC) traveled to Seattle for the first MIRC–NPEM planning meeting (Figs. 3.7 and 3.8). The meeting began with an overview of NPEM, Isite (an application of the Z39.50 protocol) and a history of the NPEM and PICES Federation project. The overview was based on a presentation given at the Fourteenth Annual PICES meeting in Vladivostok, Russia, October 2005. An overview of MIRC's data holding and metadata needs followed. Dr. Suzuki informed participants of the hierarchical structure of MIRC, JODC, and the Japan Hydrographic Association (JHA). He then discussed the varied types of data holdings available through JODC. JODC's data holdings are extremely valuable to scientists working in the North Pacific. They maintain data from several million stations dating back to the early 1800s. JODC Cruise Summary Reports (CSR) provide information for each observational cruise including date/time, research area, abstract, purpose, and contact information. Therefore, the CSR contains much of the core metadata elements that will serve as the basic source of PICES–MIRC metadata base.

The first requirement for federation is to produce FGDC-compliant metadata. Kimberly Bahl, who received training from FGDC last spring, introduced the FGDC metadata content standard and its sections and elements. This gave Dr. Suzuki the rules to write FGDC-compliant metadata records from MIRC information. Ms. Bahl also demonstrated two open-source metadata creation and validation tools, Metavist 2005 and Metadata Parser (MP). These tools allow easy creation of individual metadata records in XML file format (required for any clearinghouse node) and validation that they are FGDC-compliant. Participants used Metavist and MP to create and

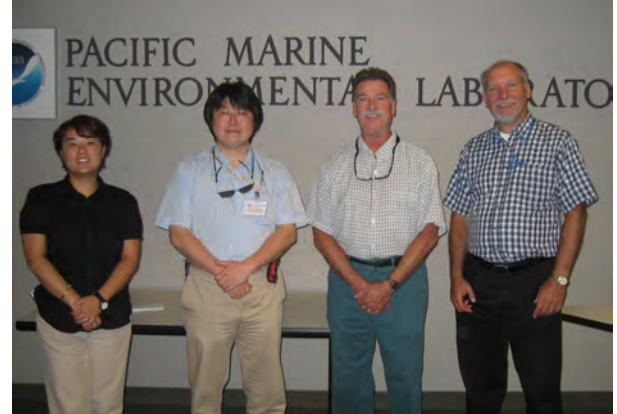


Fig. 3.7 The MIRC–NPEM Federation Team at the Pacific Marine Environmental Laboratory, Seattle, August 2006; left to right: Kimberly Bahl, Toru Suzuki, Allen Macklin, and Bernard Megrey.



Fig. 3.8 MIRC and NPEM representatives working at the Alaska Fisheries Science Center in Seattle, August 2006; left to right: Bernard Megrey, Kimberly Bahl, Toru Suzuki, and Allen Macklin.

validate an XML metadata record from a JODC CSR.

The second requirement for federation is to supply a common communication protocol: Z39.50. Ms. Bahl provided specific instructions of how to install and configure the Isite application that allows the use of the Z39.50 protocol. The Isite software suite is a free, open-source application available from the FGDC website.

The remainder of the meeting was spent discussing strategies for implementing a Japanese clearinghouse node and dealing with the problems and challenges of locating ongoing funding for the PICES Federation. PICES has been very supportive but has limited resources. At present, funding from within NOAA is unlikely. Despite

numerous efforts, attracting money from international funding organizations has not been successful. There is a possibility that NOWPAP (Northwest Pacific Action Plan) may be able to provide support for a federation. This year, MIRC will request proposals for a three-year project to begin in April 2007. Participants of this meeting will work with Dr Suzuki to develop a MIRC proposal to their funding agency, the Nippon Foundation. The proposal will provide support for ongoing MIRC participation in the PICES federation, primarily through development of a MIRC metadatabase. The meeting ended with a presentation of MIRC plans to build a demonstration site using Isite and the XML record created at the meeting and to register the node at the Clearinghouse.

The second planning meeting was held in Japan in October 2006 in conjunction with the Fifteenth Annual PICES meeting. Meetings were held at the Redbrick Warehouse in Yokohama and continued at the MIRC offices in Tokyo (Figs. 3.9 and 3.10). Mr. Norio Baba of NOWPAP also joined the discussions.

Participants reviewed issues raised at the TCODE meeting which took place a day earlier. These included the advantages of promoting the metadatabase in NOWPAP DINRAC (NOWPAP's Data and Information Network Regional Activity Center). Participants discussed the relationship between PICES TCODE and NOWPAP DINRAC activities and new opportunities for capacity building, and investigating the utility of an Asian-side metadatabase mirror server. Norio Baba said that NOWPAP has worked on metadata capacity building and might be able to invite a specialist from NPEM to collaborate. Dr. Suzuki suggested that representatives from Republic of Korea and Japan may also assist with the DINRAC activity.

Dr. Suzuki introduced the new PICES-MIRC node registered to the NSDI Clearinghouse and reported that Isite had been installed on MIRC's site and registered as 'PICES-MIRC metadatabase' on October 18. He stated that some small problems had been encountered during installation and configuration of the site.



Fig. 3.9 Group photo at the Marine Information Research Center; left to right: Toru Suzuki, Norio Baba, Kimberly Bahl, Allen Macklin, and Bernard Megrey.



Fig. 3.10 Working session at MIRC headquarters in Tokyo, Japan, October 2006; left to right: Allen Macklin, Toru Suzuki, Norio Baba, and Bernard Megrey. Kimberly Bahl (photographer, not shown).

Technical issues related to resolving these problems were discussed.

Participants reviewed the progress on the Seattle meeting action plan and amended the plan based on the Japan meeting discussions. Dr. Suzuki planned to submit a proposal by October 23, 2006, to the Nippon Foundation for metadata translation. A listing of the Seattle and Tokyo meeting minutes and agendas can be found in Appendix 15.2.

As the federation grows, the accumulated experience will make it easier for other PICES partners to join. The Pacific Institute of Geography of the Russian Academy of Sciences Far Eastern Branch has already joined the PICES Metadata Federation of its own volition. This report will provide technical guidance for anyone wishing to become a partner.

4. METADATA STANDARDS

Metadata, or data about data, describe the content, quality, condition, and other characteristics of data. For example, metadata for vertical profiles of ocean properties obtained from hydrographic casts might describe, minimally, the locations and times of the casts, the inclusive depths, the variables measured, the location of the data, and the name of the person to contact to request access to the data. In general, metadata include thematic, semantic and syntactic descriptors of the data they reference.

- Thematic metadata describe the context of the study that produced the data. Such descriptors can include, for example, principal investigator, species association and study hypothesis.
- Semantic metadata describe contextual information about the data. Candidate descriptors are measurement type, measurement device, units of measurement, calibration information, *etc.*
- Syntactic descriptors define the way the data are packaged, *e.g.*, file size, file format, storage mechanism and location.

Metadata are coded using standard transcriptions developed over the years for different purposes. For example, metadata in NPEM are described using a common set of terminology and definitions provided by the FGDC metadata standard. The data themselves are not part of the metadatabase and continue to reside with their owner. Each metadata record provides a dynamic link to the data or to the contributor. The metadata standard determines what thematic, semantic and syntactic descriptors are catalogued and how they are presented. There are a number of metadata standards, *e.g.*, Directory Interchange Format (DIF), Ecological Metadata Language (EML), Dublin Core (DC), and FGDC.

The standard coding provides a basis for search methods. When seeking a certain kind of data, a scientist or manager uses a search procedure to examine a broad body of metadata. The search procedure examines all metadata records,

eliminating those that do not satisfy the search criteria. For example, when searching for vertical profiles of ocean properties obtained from hydrographic casts, a manager might specify the locations and times of the casts, the inclusive depths and the variables measured. The successful search will reiterate these parameters and tell the searcher the location of the data, and how those data can be obtained.

Committing to a metadata standard is not a trivial decision. In this section, we review the major metadata standards so that potential Federation participants can evaluate the benefits and disadvantages of adopting alternative metadata standards.

4.1. Directory Interchange Format

Directory Interchange Format (DIF, <http://gcmd.gsfc.nasa.gov/difguide/difman.html>) is a *de facto* standard used to create directory entries that describe a group of data. A short history of DIF can be found at <http://gcmd.nasa.gov/User/difguide/whatisadif.html>.

DIF is used to create directory entries that describe a group of data. DIF consists of a collection of fields that detail specific information about the data. Eight fields are required in DIF; the others expand upon and clarify the information. Some of the fields are text fields; others require the use of controlled keywords (sometimes known as “valids”). The eight required fields are:

- Entry_ID,
- Entry_Title,
- Science_Keywords,
- ISO_Topic_Category,
- Data_Center,
- Summary,
- Metadata_Name, and
- Metadata_Version.

As with most metadata standards, DIF allows users of data to understand the contents of a data set, and contains those fields that are necessary for users to decide whether a particular data set would be useful for their needs.

The most well-known and widely used source for DIF metadata format is that employed by the Global Change Master Directory (GCMD, <http://gcmd.nasa.gov/>).

4.2. Ecological Metadata Language

Ecological Metadata Language (EML) is a metadata specification developed by the ecology discipline and supported by The Knowledge Network for Biocomplexity (<http://knb.ecoinformatics.org/home.html>). It is based on prior work done by the Ecological Society of America and associated efforts (Michener *et al.*, 1997). EML is implemented as a series of XML document types that can be used in a modular and extensible manner to document ecological data. Each EML module is designed to describe one logical part of the total metadata that should be included with any ecological dataset.

EML is one of the newer metadatabase standards. The benefit of using EML is that it is a superset of FGDC (eliminating potential incompatibility) and includes a means to describe biological metadata.

To access the EML specification on-line, read it in HTML (Hypertext Markup Language) format, or download the entire specification, including both the HTML documentation and the XML Schema files, browse the EML 2.0.1 Specification at <http://knb.ecoinformatics.org/software/eml/eml-2.0.1/index.html>. The EML modules can be downloaded from the Internet at <http://knb.ecoinformatics.org/software/download.html#eml>; EML XML Schema can be found at <http://www.w3c.org/XML/Schema>, and general information on EML can be found at <http://knb.ecoinformatics.org/software/eml/>.

4.3. Dublin Core

The Dublin Core (DC) may be considered to be the best-known metadata project today. From its start in 1995, DC evolved into 'the leading initiative for improving resource discovery on the Web' (Weibel, 2000). During a series of metadata workshops, DC was created and further refined. It is a minimal (15 fields) generic metadata scheme for virtually any kind of document.

DC is a metadata format determined on the basis of international consensus that has defined a minimal information resource description, generally for use in a World-Wide Web environment. The description of the information resources is created using DC elements. An element is a pre-defined string or label, which is paired to a value. In its simplest form it looks like this: 'Author = Ronald Snijder'. A list of fifteen elements is defined, which will be discussed in the next subsection. The syntax of the elements is simple: just the name (or 'Identifier') of the element, and the value of the element. The value may consist of free text or it may be taken from a standardized resource. The description may reside in a separate file or it may be a part of the information resource itself. While no formal syntax rules are defined, several syntax recommendations have been created for generic text files, HTML and the Resource Description Framework (RDF). These are discussed in (Hillmann, 2000).

4.3.1. Elements of DC

The DC element set originally consisted of thirteen elements. Below are the elements, as defined in the first DC metadata workshop held in Dublin, Ohio, March 1995. For a detailed discussion of those elements, see Weibel and Godby *et al.* (1995).

- Subject – Topic addressed by the work,
- Title – Name of the object,
- Author – Person(s) primarily responsible for the intellectual content of the object,
- Publisher – Agent or agency responsible for making the object available,
- OtherAgent – Person(s), such as editors and transcribers, who have made other significant intellectual contributions to the work,
- Date – Date of publication,
- ObjectType – Genre of the object, such as novel, poem, or dictionary,
- Form – Data representation of the object, such as Postscript file or Windows executable file,
- Identifier – String or number used to uniquely identify the object,
- Relation – Relationship to other objects,

- Source – Objects, either print or electronic, from which this object is derived, if applicable,
- Language – Language of the intellectual content,
- Coverage – Spatial locations and temporal durations characteristic of the object.

Information on the Dublin Core Metadata Initiative can be found at <http://www.geocities.com/ronaldsnijder/#2>. General information can be found at http://alexandria.sdc.ucsb.edu/public-documents/metadata/md_imp.html. The Dublin Core Metadata Glossary can be found at <http://library.csun.edu/mwoodley/dublincoreglossary.html>

4.4. Federal Geographic Data Committee

The Federal Geographic Data Committee (FGDC, <http://www.fgdc.gov/metadata>) Content Standards for Digital Geospatial Metadata (CSDGM) specify the information content of metadata for a set of digital geospatial data. The purpose of the standard is to provide a common set of terminology and definitions for documentation related to these metadata. The standard provides a common set of terminology and definitions for the documentation of geospatial data. The standard establishes the names of data elements and groups of data elements to be used for these purposes, the definitions of these data elements and groups, and information about the values that are to be provided for the data elements. Information about terms that are mandatory, mandatory under certain conditions, and optional (provided at the discretion of the data provider) also is provided by the standard. Description of the metadata elements can be found at the link above.

The FGDC standard has recently been updated to include the Biological Data Profile (<http://www.fgdc.gov/standards/projects/FGDC-standards-projects/metadata/biometadata/biodatap.pdf>). The profile broadens the application of the CSDGM so that it is more easily applied to data that are not explicitly geographic (laboratory results, field notes, specimen collections, research reports) but can be associated with a geographic location. The profile changes the conditionality and domain of CSDGM elements, requires the use

of a specified taxonomical vocabulary, and adds elements.

Information on FGDC can be found at National Geospatial Data Clearinghouse. (<http://www.fgdc.gov/metadata>). The FGDC Metadata Glossary can be found at <http://www.fgdc.gov/metadata/csdgm/glossary.html>.

4.5. The ISO 19115 Metadata Standard

The international community, through the International Organization for Standardization (ISO), has developed and approved an international metadata standard, ISO 19115. As a member of ISO, the United States is required to revise the CSDGM in accord with ISO 19115. Each nation can craft their own profile of ISO 19115 with the requirement that it include the 13 core elements. The FGDC is currently leading the development of a U.S. profile of the (ISO) international metadata standard, ISO 19115.

Table 4.1 ISO 19115 Core Metadata Elements.

Mandatory Elements	Conditional Elements
Dataset title	Dataset responsible party
Dataset reference date	Geographic location by coordinates
Dataset language	Dataset character set
Dataset topic category	Spatial resolution
Abstract	Distribution format
Metadata point of contact	Spatial representation type
Metadata date stamp	Reference system
	Lineage statement
	On-line Resource
	Metadata file identifier
	Metadata standard name
	Metadata standard version
	Metadata language
	Metadata character set

The status of U.S. Implementation of ISO Metadata can be found at <http://www.fgdc.gov/metadata/geospatial-metadata-standards#isometadata>.

4.6. Metadata stylesheets

Three XML stylesheets have been developed to convert DIF to FGDC and FGDC to DIF. These are listed in Appendix 15.3 and are available from the PICES web site.

4.7. Crosswalks

To reach the broadest community of information workers, metadata must be made available in accordance with a number of popular content metadata standards. As the number, size, and complexity of content metadata standards continues to grow, supplying the metadata for each standard becomes more and more repetitious, time consuming, and tedious. In order to minimize the amount of time needed to create and maintain the metadata and to maximize its usefulness to the widest community of users, there is a need for the metadata created and maintained in one standard to be accessible via related content metadata standards.

A crosswalk is a specification for mapping one metadata standard to another. Technically, a crosswalk is a set of transformations applied to the content of elements in a source metadata standard that results in the storage of appropriately modified content in the analogous elements of a target metadata standard. Crosswalks provide the ability to make the contents of elements defined in one metadata standard available to communities using related metadata standards.

Several crosswalks are available to convert from one metadata standard to another.

- See http://gcmd.gsfc.nasa.gov/Aboutus/standards/fgdc_to_dif.html to convert from the FGDC to the GCMD DIF standard.
- See http://gcmd.gsfc.nasa.gov/Aboutus/standards/dublin_to_dif.html to convert from DC to the GCMD DIF standard.
- See http://www.fgdc.gov/metadata/documents/FGDC_Sections_v40.xls to convert from CSDGM to ISO 19115. This tool was developed by Intergraph as part of the NSDI Cooperative Agreement Program (CAP) (<http://www.fgdc.gov/grants/index.html>).
- See http://gcmd.gsfc.nasa.gov/Aboutus/standards/esri_to_dif.html to convert from GCMD DIF

format to the Environmental Systems Research Institute, Inc. (ESRI) profile of FGDC.

4.8. Tools for creating metadata

There are many useful tools to assist with creating metadata, some of which are listed below.

- ARC/INFO GIS Metadata Generator AMLs (ftp://moon.cecer.army.mil/pub/metadata/arc_meta/)
- ASCII template (ftp://moon.cecer.army.mil/pub/metadata/ascii_template)
- Another ASCII template (<ftp://ftp.nbs.gov/pub/metadata/mptemplt.txt>)
- CORPSMET (<http://sco.wisc.edu/wisclinc/metatool/corpsmet.htm>) is a U.S. Army Corps of Engineers (USACE) program to create metadata.
- MP (<http://geology.usgs.gov/tools/metadata/tools/doc/mp.html>), a compiler for formal metadata by developed by Peter Schweitzer of the United States Geological Survey (USGS).
- NOAA's FGDC Metadata Toolkit Software (http://www-orca.nos.noaa.gov/cgi-bin/titledetails.pl?2_SEA_MetadataToolkit)
- XTME (<http://geology.usgs.gov/tools/metadata/>) is a Metadata Entry System.
- Metavist 2005 (<http://ncrs.fs.fed.us/pubs/viewpub.asp?key=2737>) is a stand-alone metadata creation and editing tool developed by the USDA Forest Service North Central Research Station. Metavist 2005 creates metadata compliant with the CSDGM (FGDC) 1998 metadata standard and the National Biological Information Infrastructure (NBII) 1999 Biological Data Profile for the FGDC standard. The software runs under the Microsoft Windows 2000 and XP operating systems, and requires the presence of Microsoft's .Net Framework version 1.1. The metadata are output in XML format.

5. COMMUNICATION PROTOCOLS

Communication protocols establish pre-defined methods whereby hosts and applications can exchange information through the World-Wide Web.

5.1. Z39.50

Z39.50 is a communication protocol used to integrate information resources in a distributed environment. The current version of Z39.50 is more properly known as North American standard ANSI/NISO Z39.50-1995, Information Retrieval (Z39.50): Application Service Definition and Protocol Specification (<http://lcweb.loc.gov/z3950/agency/1995doce.html>), or as the matching international standard ISO 23950:1998, Information and Documentation/Information Retrieval (Z39.50): Application service definition and protocol specification. The current release is version 3 of the American National Standards Institute/National Information Standards Organization (ANSI/NISO) standard, and dates back to 1995. Version 3 is the dominant version of Z39.50 utilized in Europe, although a number of North American sites continue to use the earlier version 2.

The standard's formal home on the Worldwide Web is the Z39.50 Maintenance Agency (<http://lcweb.loc.gov/z3950/agency/>) hosted by the United States' Library of Congress (<http://lcweb.loc.gov/>). Despite a common misconception to the contrary, Z39.50 is *not* simply used by libraries, although the library sector is one with a clear and long-held need for Z39.50-type functionality. In the world of government and community information, too, utilities such as the Government Information Locator Service (GILS) Profile (http://www.gils.net/prof_v2.html) makes use of Z39.50 to link a wide range of resources internationally.

There is a huge amount of information available on the web relating to Z39.50. A basic search for the term "Z39.50" on 03 December 2006,

produced 1,450,000 hits from Google (<http://www.google.com>), 699,000 hits from YAHOO (<http://www.yahoo.com/>) and 530,000 returned hits from FAST (<http://www.alltheweb.com/>). Two sites prove to be useful first stops for information. These are the Maintenance Agency's own web site (<http://lcweb.loc.gov/z3950/agency/>), and Dan Brickley's Z39.50 Resources page at the Institute for Learning and Research Technology (ILRT) <http://www.ilt.bris.ac.uk/discovery/z3950/resources/>.

5.1.1. What does Z39.50 do?

Z39.50 is designed to enable communication between computer systems. This communication could be between a scientist's PC and a database server at a National Oceanographic Data Center running on a UNIX server in the basement, and equally it could be between a user browsing the web from Plymouth UK, seeking data residing on a server in NODC and a JODC database in Tokyo. This transparency is undeniably useful to those individuals who have to update data repositories. But it is the latter application and others like it that represents much of the potential of Z39.50 in today's distributed network environment.

5.1.2. Isite

Isite is a software client that relies on the Z39.50 client for communication protocol. It was developed by the Center for Networked Information Discovery and Retrieval.

The Isite software is described at <http://www.csdl.tamu.edu/DL95/papers/nebert/nebert.html> and can be downloaded from <http://www.fgdc.gov/dataandservices/getisite> or <http://clearinghouse4.fgdc.gov/ftp/>.

The configuration of the Isite software suite is illustrated in Fig. 5.1.

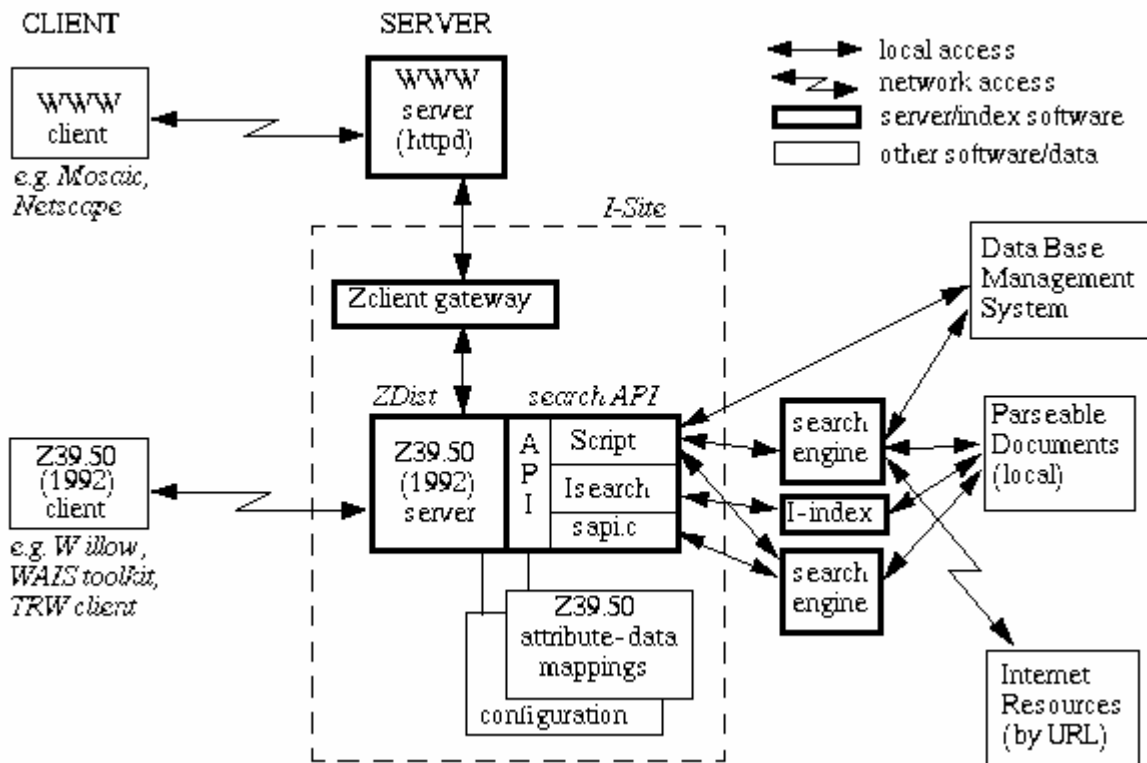


Fig. 5.1 Configuration of I-site Z39.50-compliant software developed by the Clearinghouse for Networked Information Discovery and Retrieval.

6. CLEARINGHOUSES

A clearinghouse is an institution that collects and distributes information. Clearinghouses began with financial, medical and transportation industries. With the advent of the World-Wide Web, it became possible and advantageous for many organizations to belong to central clearinghouses. Thus, they proliferated. With the popularization of geospatial data and the infrastructure afforded by Geographic Information Systems (GIS) applications, clearinghouses for geographic data began to appear.

The Geospatial Data Clearinghouse (<http://registry.fgdc.gov/serverstatus/>) is a collection of over 450 spatial data servers that have digital geographic data primarily for use in GIS, image processing systems, and other modeling software. These data collections can be searched through a single interface based on their descriptions, or “metadata”.

7. METHODOLOGY

To enable an Internet browsing client to search and discover information through a federated metadata database, four elements must be in place.

1. The client must be able to communicate with the clearinghouse using normal web communications.
2. The clearinghouse must offer an Internet communication protocol and associated utilities that permit search and discovery of federated metadata records.
3. Each federated partner must maintain metadata records using a standard that is supported by the communication protocol.
4. Each federated partner must serve the Internet through a server that is conversant in the communication protocol and hosts metadata records in the proper standard. Figure 7.1 shows these elements.

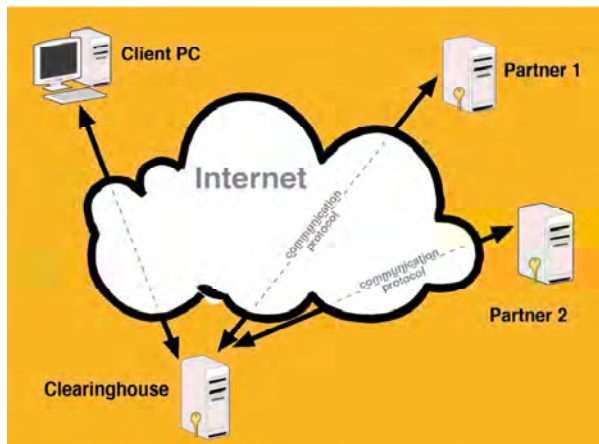


Fig. 7.1 An Internet metadata federation requires a clearinghouse and partners sharing a communication protocol and serving metadata using the same standard.

There are three main steps to creating a Clearinghouse node. Figure 7.2 illustrates the required components and steps to developing a node.

1. Create a FGDC metadata or take existing metadata and make them compliant with the FGDC's CSDGM.

2. Provide a server with the ISite software suite installed to allow the NSDI Clearinghouse server to connect.
3. Register the server node with the NSDI Clearinghouse to allow metadata to be indexed, searched and retrieved by remote users.

The following list describes the processes shown in Fig. 7.2.

- Obtain or create new metadata,
- Download a pre-parser (CNS) and the metadata parser (MP),
- Download ISite,
- Configure ISite,
- Index metadata,
- Test index,
- Server index,
- Test server,
- Register server.

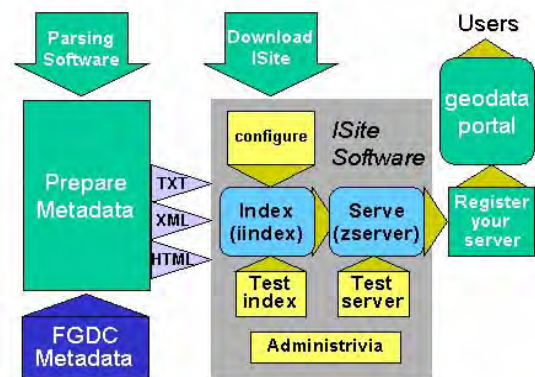


Fig. 7.2 A Clearinghouse node component and development diagram, <http://www.fgdc.gov/dataand services/isite tutorial>.

7.1. FGDC metadata

In order to develop a clearinghouse node, the PICES Metadata Federation partner must first have, or convert to, metadata that comply with the U.S. Federal Geographic Data Committee's workbook CSDGM. The current standard is in version 2 (FGDC-STD-001-1998; Federal Geographic Data Committee, 1998).

The CSDGM consists of seven main sections and three supporting sections. Within in each section there are various component elements. Figure 7.3 depicts a color-coded diagram that provides a graphical representation of the ten sections. Refer

to Appendix 15.4 for diagrams of all sections. Each section diagram provides color-coded element boxes contained within the section and the conditionality of the elements.

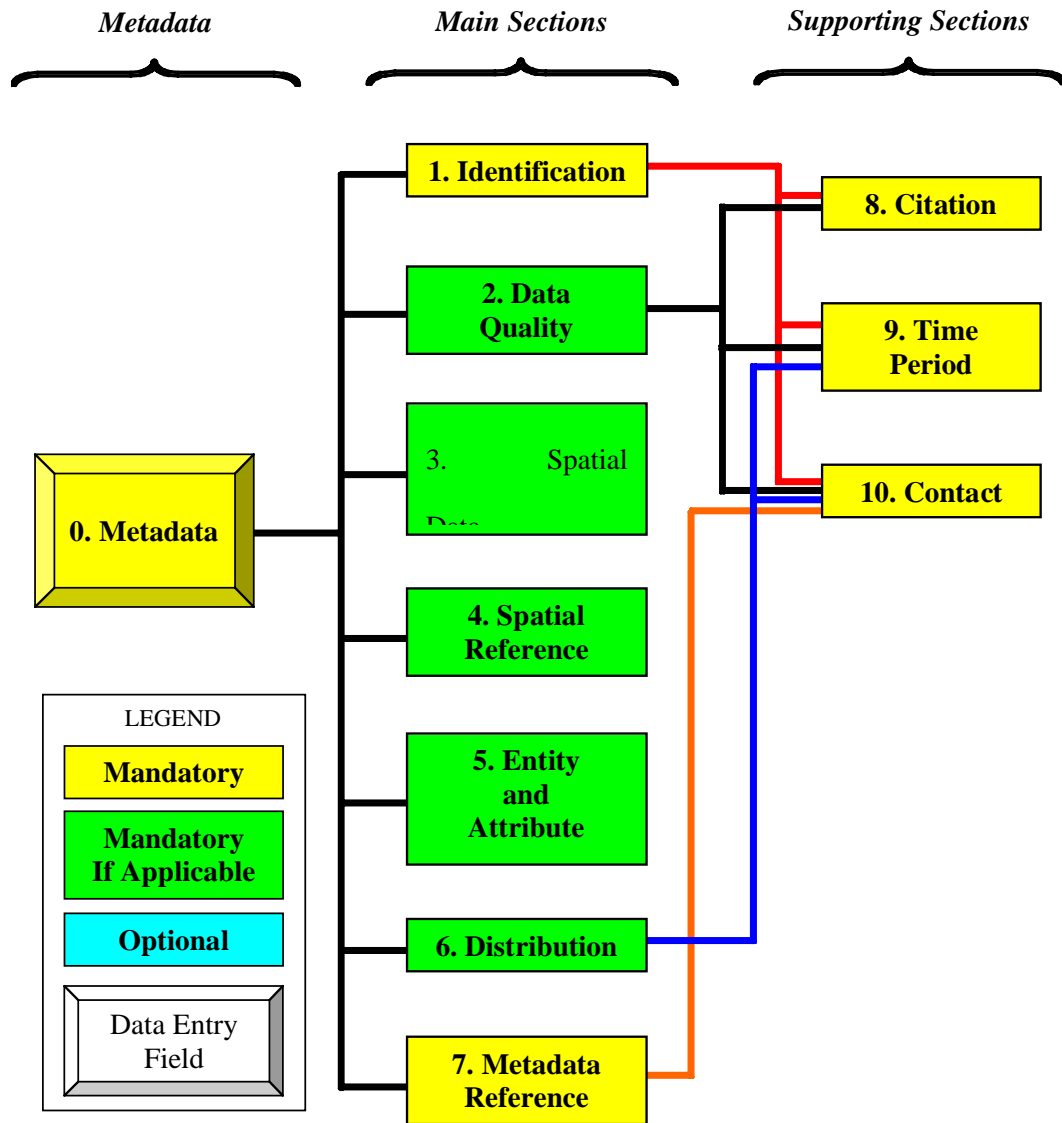


Fig. 7.3 FDGC Metadata Sections of the Standards.

For a basic but still complete metadata record, sections one and seven are required. With the two main required sections, supporting sections: 8, 9, and 10 must be included. Each section is made up of compound elements, which are made up of data elements. All sections and elements are either mandatory, mandatory if applicable, or optional. Review the CSDGM workbook for a complete and sample description of all sections and elements.

Sections are the main chapters of the standards. Each section is composed of a section definition; list of elements, definitions, types, and values; and information about what is mandatory and repeatable.

Metadata contain the following sections and elements. Section 0, “Metadata” provides the starting point. It is composed of the main sections of the standard.

7.1.1. Main sections

Sections 1 through 7 are the main sections of the standard. Providing sections 1, 7, 8, 9, and 10 of metadata, basically answers the “who”, “what”, “where”, “when”, “why”, and “how” of a data set. If more information about the data set is to be provided, review the additional “mandatory if applicable” sections.

1. Identification Information (*mandatory*)
2. Data Quality Information (*mandatory if applicable*)
3. Spatial Data Organization Information (*mandatory if applicable*)
4. Spatial Reference Information (*mandatory if applicable*)
5. Entity and Attribute Information (*mandatory if applicable*)
6. Distribution Information (*mandatory if applicable*)
7. Metadata Reference Information

7.1.2. Supporting sections

Sections 8, 9, and 10 support other main sections and are never used alone.

8. Citation Information (*mandatory*)
9. Time Period Information (*mandatory*)
10. Contact Information (*mandatory*)

See Appendix 15.4 or the CSDGM workbook to view all metadata sections and elements.

Figure 7.4 provides a diagram of the CSDGM Section 1 Identification Information (FGDC-STD-001-1998; Federal Geographic Data Committee, 1998). Provide at least the first 10 compound elements.

Figure 7.5 is the Section 7 Metadata Reference (FGDC-STD-001-1998; Federal Geographic Data Committee, 1998). This section also requires a supporting section within. Provide elements 7.1, 7.4, 7.5, and 7.6 to meet the requirements.

Figure 7.6 is a diagram of the metadata Supporting Sections 8, 9, and 10 (FGDC-STD-001-1998; Federal Geographic Data Committee, 1998).

With the understanding of the FGDC metadata standards, metadata can now be created. To create metadata, use one of the many metadata creation tools written for different user communities. A list of tools can be found at <http://www.fgdc.gov/metadata/geospatial-metadata-tools>. The tools may be freely or commercially available. Most metadata creation tools provide a user input form to enter, update and output metadata. Metadata should be in one of the following formats: TXT, HTML, SGML, or XML. The XML form is the recommended format and the only required form by the FGDC Isite use. For more information regarding the FGDC metadata standard structure and content, refer to the FGDC CSDGM Document Type Definition (DTD) available through the Internet at <http://geochange.er.usgs.gov/pub/tools/metadata/tools/doc/metadata.dtd> and an XML Schema Document (XSD) representation available on-line at http://www.csc.noaa.gov/metadata/xml/fgdc_std-001-1998-ann.zip.

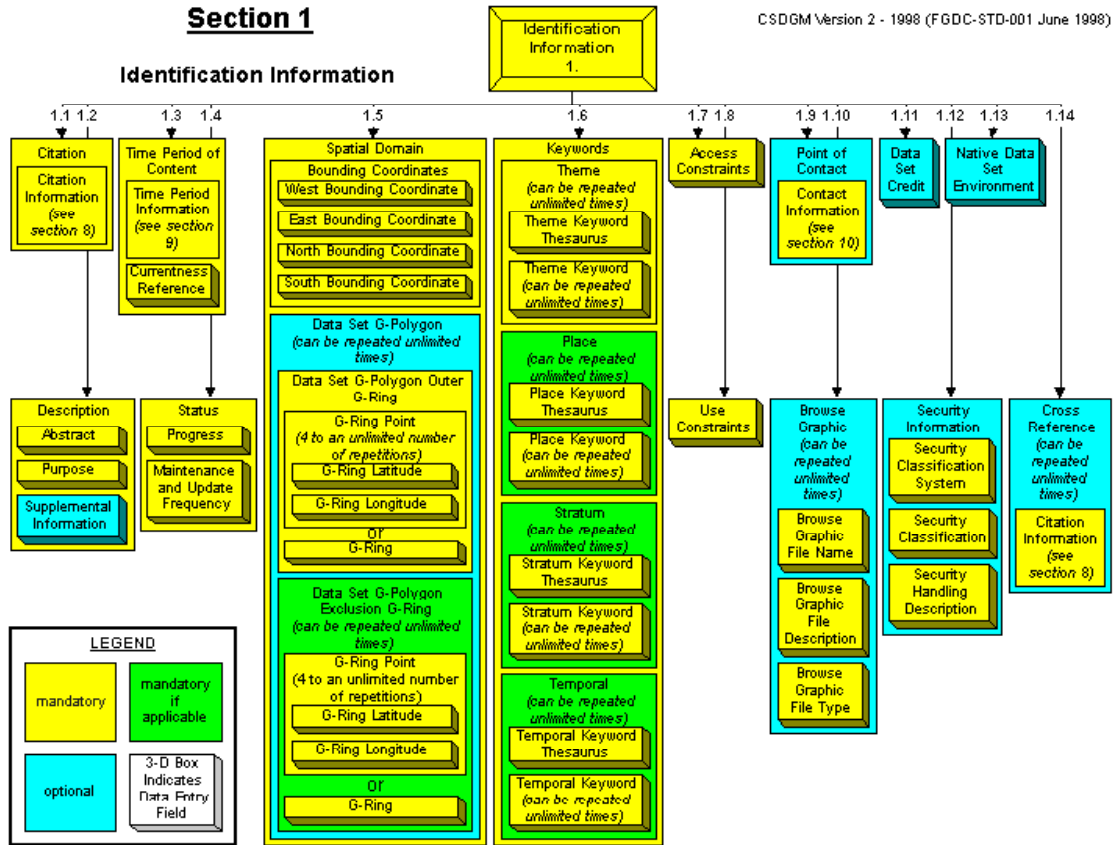


Fig. 7.4 CSDGM Section 1 Identification Information (http://www.fgdc.gov/metadata/documents/workbook_0501_bmk.pdf).

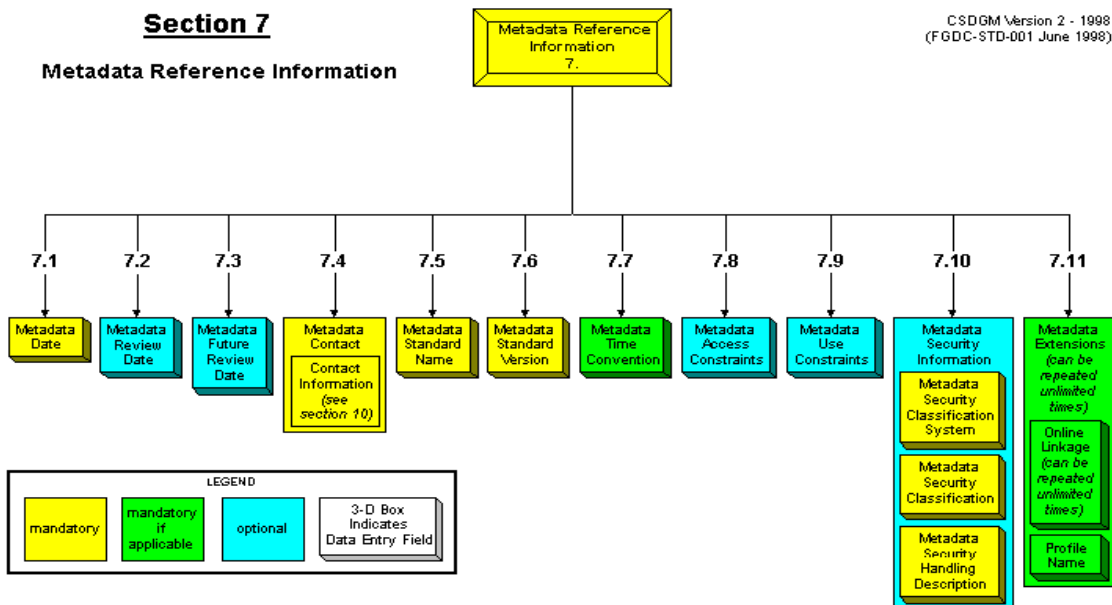
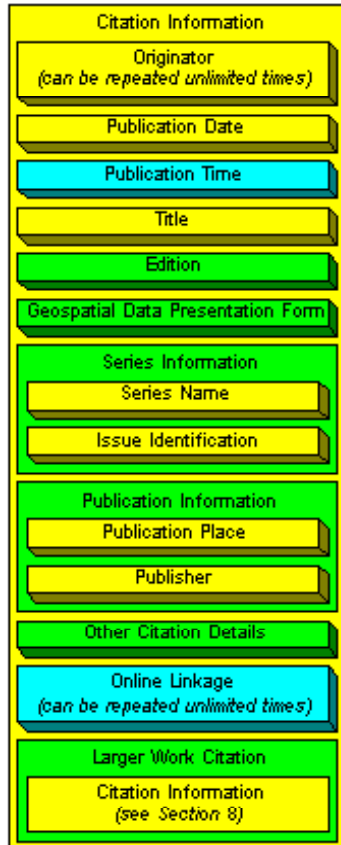


Fig. 7.5 CSDGM metadata Section 7 Metadata Reference Information (http://www.fgdc.gov/metadata/documents/workbook_0501_bmk.pdf).

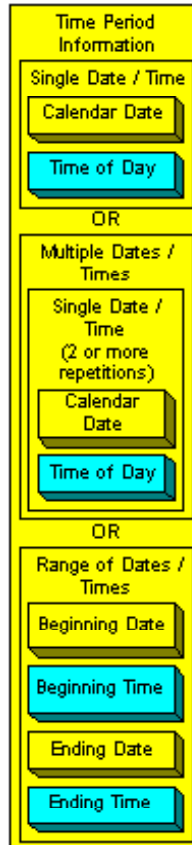
Section 8

Citation Information



Section 9

Time Period Information



Section 10

Contact Information

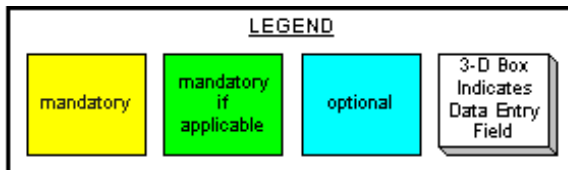
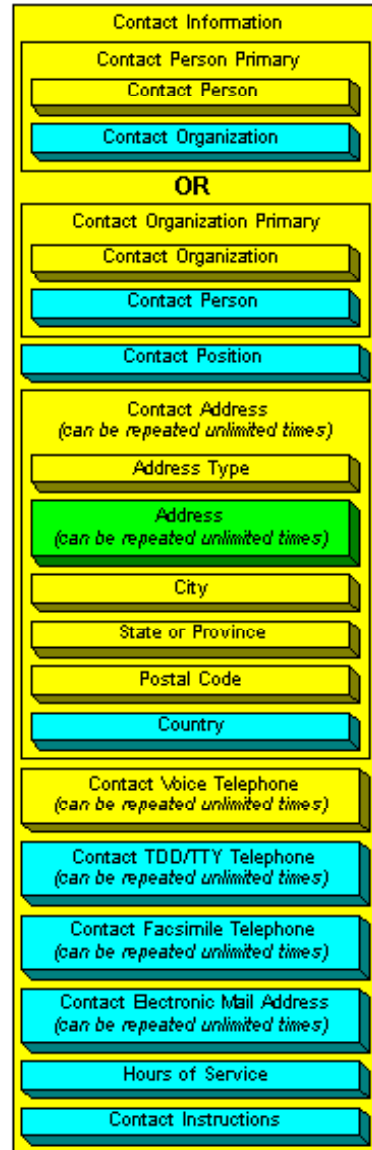


Fig. 7.6 A diagram of CSDGM Sections 8, 9, and 10 (http://www.fgdc.gov/metadata/documents/workbook_0501_bmk.pdf).

7.1.3. *Metadata validation*

For Isite to index metadata properly, all metadata files must be validated. Download metadata parser (MP) and pre-parser (CNS) to validate and reformat FGDC metadata. The MP program is a compiler for formal metadata that checks the structure of each metadata record against the FGDC standard. The CNS is a pre-parser for formal metadata. This validation is important because in order for metadata entries to be queried, correct tag names must be assigned to the search engine when the metadata documents are loaded; CNS and MP software can be obtained from <http://geology.usgs.gov/tools/metadata>.

If the existing metadata have been created using a different metadata standard, XSLT (XSL Transformations) style sheets created by different organizations maybe obtained to convert to the FGDC standard.

The MP and CNS software may be installed in a common-use directory. It is recommended that the executable file be renamed to its prefix (*e.g.*, MP) after download.

7.2. **Getting a copy of Isite**

Isite is available for free use from the FGDC website. To acquire and use Isite, download the latest version of the Isite software from the FGDC

FTP webpage at <http://clearinghouse4.fgdc.gov/ftp/>.

7.3. **NSDI Clearinghouse**

This section provides a tutorial on how to set up the Clearinghouse node server. The Isite software suite will be installed, run to tag the metadata files, and tested locally and externally. Once the server and software configuration is complete, the server can be registered with the NSDI Clearinghouse Network.

The NSDI Clearinghouse Network allows a repository of data servers or clearinghouse nodes to be searched for metadata and to retrieve those metadata. Registering on the NSDI Clearinghouse Network is very simple. Just fill out an on-line form on the NSDI Clearinghouse website at <http://registry.gsdi.org/>. During registration, the agency or the project responsible for the Clearinghouse node and its metadata will be allowed to provide a description of the project and its metadata. To conform to the PICES Metadata Federation project and its Clearinghouse node naming convention, name your node as “PICES – [agency name or project name]”. This will allow searching for metadata among the PICES members Clearinghouse nodes easier. See Section 9 of this report to view a search tutorial and examples of metadata.

8. SERVER CONFIGURATION AND TECHNICAL ISSUES

The FGDC Metadata Clearinghouse is a decentralized system of Internet servers you can use to search for available geospatial data. The PICES Metadata Federation (Fig. 8.1) has successfully connected servers in Japan, Republic of Korea, Russia, and the U.S.

An overview of the steps required to prepare metadata and make it available on an Internet server is summarized in Figure 8.2. In this section we will go through each step in detail.

8.1. Hardware recommendations

A typical computer (*i.e.*, 2.3 GHz Intel CPU, 512 MB RAM and 120GB HD) is all that is required to act as a server node. In the U.S. the cost is approximately \$1,200.

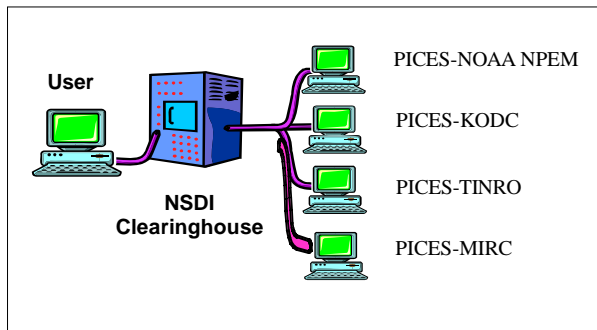


Fig. 8.1. Schematic of the PICES Metadata Federation.

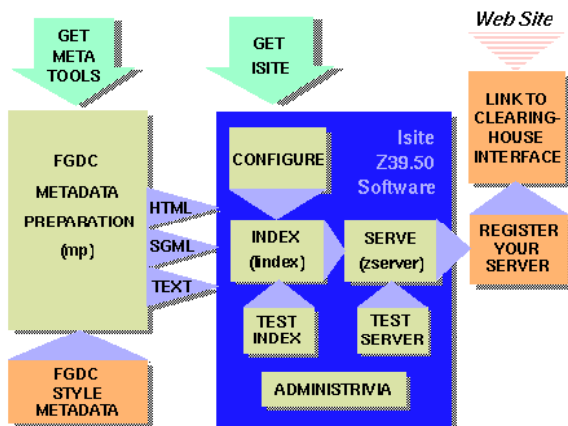


Fig. 8.2 Steps required to prepare metadata and make it available on an Internet server.

8.2. Operating system – Red Hat Linux Fedora

We recommend using the freely available Linux operating system by Red Hat. The Fedora Project (<http://fedoraproject.org/>) is a Red Hat sponsored and community supported open-source project. The goal of the Fedora Project is to work with the Linux community to build a complete, general-purpose operating system exclusively from free software.

Red Hat Fedora can be downloaded from <http://fedoraproject.org/wiki/Distribution/Download> or from the Red Hat Fedora Linux Operating System website found at <http://fedora.redhat.com/>.

8.3. Web services – Apache HTTP Server version 2.2.3

Web services such as HTTP (Hypertext Transfer Protocol), *etc.* are provided by Apache HTTP Server. Apache HTTP Server version 2.2.3 (sometimes called Apache httpd) is a project of the Apache Software foundation aimed at creating a robust, commercial-grade, feature-filled, and freely-available source code implementation of an HTTP (Web) server. For software, installation instructions and documentation, browse <http://httpd.apache.org/download.cgi>.

8.4. Create and validate FGDC-compliant Metadata in XML format

Your clearinghouse server (node) will need a copy of every metadata record in XML format stored in a predetermined subdirectory on the server.

You will need to decide if you want to store your metadata in a relational database or in discrete XML metadata documents. Deciding between holding your metadata in a database or producing discrete metadata documents for each data set is somewhat dependent on the variety and volume of your data sets, as well as how often they (and the metadata) are updated. If you store your metadata in a relational database, then you will need a script of some kind to extract each metadata record from

the database and prepare it as a FGDC record in XML format.

Several tools are available to create and validate FGDC metadata. Free products include:

- Tk Metadata Editor (tkme)
<http://geology.usgs.gov/tools/metadata/tools/doc/tkme.html>
- Xt Metadata Editor (xtme)
<http://geology.usgs.gov/tools/metadata/tools/doc/xtme.html>
- MetaScribe
<http://www.csc.noaa.gov/metadata/metascibe/>
- NOAA's Metadata Enterprise Resource Management Aid (MERMAid)
<http://www.ncddc.noaa.gov/Metadata/tools>
- Metavist 2005
<http://doc.nprb.org/web/metadata/workshop/MetaVist/?C=S;O=A>.

Some capable commercial products (not free) are:

- ArcCatalog
<http://www.esri.com/software/arcgis/about/desktop.html>
- GeoMedia Catalog
<http://www.intergraph.com/geomediacatalog/default.asp>
- SMMS (Spatial Metadata Management System)
<http://marinemetadata.org/tools/refs/SMMS/>.

We prefer Metavist as it is easy to use and produces compliant XML metadata records. For documentation, see the website http://ncrs.fs.fed.us/pubs/gtr/gtr_nc255.pdf#search=%22Metavist%22.

Metadata can be checked with any of the metadata validation utilities listed below.

- Chew 'n Spit (CNS)
<http://marinemetadata.org/tools/refs/CNS/>
- Metadata Parser (MP)
<http://geology.usgs.gov/tools/metadata/tools/doc/mp.html>

- Enumerated Domain Helper
<http://geology.usgs.gov/tools/metadata/tools/doc/ctc/edom.shtml>
- MP Batch Processor
<http://support.intergraph.com/Geospatial/Downloads/Tools.asp?ID=48&SORT=Title>
- Metadata Validation Service
<http://geo-nsdi.er.usgs.gov/validate.php>.

MP and MP Batch Processor do a good job for individual records or a library of records.

8.5. Obtaining, installing and configuring Isite for UNIX/Linux

8.5.1. Download the appropriate Isite software

Download the appropriate Isite software by going to <http://clearinghouse4.fgdc.gov/ftp/>. Figure. 8.3 shows an image of the Isite web site listing versions of Isite for different operating systems.

For this example, we will be using the Linux 2.6.8



Fig. 8.3 Image of the Isite web site showing versions of Isite for different operating systems.

version of Isite. We will also be using GNU tar (available from the server at <http://www.gnu.org/software/tar/tar.html>) to extract it.

Conventions: Commands should be entered all on one line unless specified otherwise. We will also assume that your database name will be in **bold**, and the string you are searching for will be in **bold italics**. The \$ at the beginning of each command is the Linux shell prompt.

8.5.2. *Untar the file*

For this example, we will be using the /opt directory to untar Isite using the command

```
$ tar xzvf clearinghouse2-  
v2.3_Linux_2.6.8-24.10-i386-  
2005010602-static.tar.gz
```

This should will generate an Isite2 directory and output something like the following:

```
/opt/Isite2/bin/GetScript  
/opt/Isite2/bin/Key1  
/opt/Isite2/bin/SearchScript  
/opt/Isite2/bin/Iindex2  
/opt/Isite2/bin/Isearch2  
/opt/Isite2/bin/Iutil2  
/opt/Isite2/bin/Iget2  
...
```

Isite2 contains two directories of documentation and software

The /opt/Isite2/bin subdirectory contains the following executables:

- Iindex2 – indexes metadata entries
- Isearch2 – provides local search test
- zclient – provides command-line query of remote or local Z39.50 service
- Izclient – interactive Z39.50 client using ANSI terminal node
- zserver – the Z39.50 server process
- zping – checks the availability of a given Z39.50 service (local or remote)

In the /opt/Isite2/bin subdirectory, there are two initialization files, the Search Application Program Interface initialization file (sapi.ini) and the zserver process initialization file (zserver.ini). The sapi.ini file contains specific configuration information for your indexes. The zserver.ini file

specifies the port number to be used to connect to the Internet, and the DBList parameter lists the names of the databases described in further detail in the sapi.ini file.

Images of the server directory structure and a listing of files contained in each subdirectory are shown in Appendix 15.5.

8.5.3. *Name your database*

Pick a name for your database, preferably one without spaces in it. It should be a unique name (not, for example, FGDC) that is indicative of your data content. It *does* matter how you capitalize this name, so be consistent.

Examples: **arizona**, **FrameworkData**, **ABCCounty**. For this example, we will use our DB name of **NPEM**.

8.5.4. *The zserver.ini file*

Open and edit the file zserver.ini. Confirm that the following parameters are set as they are below:

```
ServerType=STANDALONE  
Port=6668  
SAPI=sapi.ini  
AccessLog=zserver_access.log  
DBList=NPEM
```

Note that the port number is the communication link with the Internet. If you have a firewall installed then this port must be explicitly open to the Internet in order for the Clearinghouse node to communicate with your Isite server.

You will need to create a location to store your database. For this example we created the directory /opt/Isite2/index.

A listing of the zserver.ini file used for the NPEM node can be found in Appendix 15.6.

8.5.5. *The sapi.ini file*

Edit the sapi.ini file to look like the following structure:

```
[DEFAULT]  
DBList=NPEM
```

```
[NPEM]
type=ISEARCH
location=/opt/Isite2/index
FieldMaps=/opt/Isite2/bin/bib1_fgdc.m
ap,\
  /opt/Isite2/bin/gils_fgdc.map,\
  /opt/Isite2/bin/geo_fgdc.map
```

In this example, the database is called “**NPEM**”, and this must match the DBList parameter in zserver.ini. The database is stored in the /opt/Isite2/index directory. The type is ISEARCH.

Verify that Fieldmaps are in the right location. The pathnames to Fieldmaps *must* be set to the location where you just installed Isite on your system. Be sure that you have no spaces in your Fieldmaps declaration line. Verify that these files are in the right locations by typing the following commands:

```
$ file /opt/Isite2/index
$ file /opt/Isite2/bin/bib1_fgdc.map
$ file /opt/Isite2/bin/gils_fgdc.map
$ file /opt/Isite2/bin/geo_fgdc.map
```

Each command should return either “directory” or “ASCII text”. If it returns “(No such file or directory)”, check your pathnames.

A listing of the sapi.ini file used for the NPEM node can be found in Appendix 15.6.

8.5.6. Indexing metadata

To index all metadata files, first create the directory /opt/Isite2/index. Then go to the /opt/Isite2/bin directory and use the Iindex2 command with the following arguments from the Isite2/bin directory. The lines below are all one command:

```
$ /opt/Isite2/bin/Iindex2 -d
/opt/Isite2/index/NPEM -t fgdc -o
fieldtype=/opt/Isite2/bin/fgdc.fields
/opt/Isite2/data/*.xml
```

Note: For this example we will assume your metadata is located in /opt/Isite2/data. We have saved the indexing commands above in a script file, DoIndex, which we placed in the /opt/Isite2/bin/ subdirectory.

We execute the script from the /opt/Isite2/bin/directory with the command

```
$ sh DoIndex
```

The database **NPEM** will be written to the /opt/Isite2/index directory, using FGDC metadata format, using FGDC search fields, and using as input all files in the /opt/Isite2/data directory that end in *.xml. You will need to adapt these pathnames to match your system.

You can find advanced indexing options by browsing:

<http://clearinghouse4.fgdc.gov/ftp/advancedindexing.html>.

If all is successful, you should receive a message similar to the following:

```
Index v2.2
Building document list ...
Creating new cache
[/opt/Isite2/index/NPEM.db]
Building database
/opt/Isite2/index/NPEM:
Parsing files ...
  Parsing
/opt/Isite2/data/npem993.xml
key=276712911580
```

(This line will repeat for every XML file that is indexed.)

```
Indexing 21624 words ...
Writing compressed blocks to sequence
table 1
.....
Wrote 36 blocks of compressed
strings.
Done with sequence table 1
Dumping sparse table 1...
Done with sparse table 1
Database files saved to disk.
```

It is recommended that you save this command line as a script that can be run over and over again and edited if changes are warranted.

8.5.7. Start the Clearinghouse Server process

Start your Z39.50 Clearinghouse Server process by changing directory to `Isite2/bin` and invoking the `zserver` command. You may wish to start the process in the background with the Linux command:

```
$ ./zserver &
```

You could also consider including it in startup scripts typically stored in the `/etc/rc.d` directory so it will get restarted after the system reboots. Otherwise, the `zserver` process will have to be manually started after every reboot.

8.5.8. Testing the `zserver` installation

Once you have the `zserver` running (on port 6668 and with a database named “**NPEM**”), you can test it by confirming that it is running by executing the following command:

```
$ ./zping localhost 6668
```

You should get the response:

```
Z39.50 server at localhost on port
6668 is alive.
```

The following title search for the string “*data*” (or some other word you know will be in title records within your database) should return some hits:

```
$ ./zclient localhost 6668 "NPEM"
"data[1,4]"
```

Confirm that a full text search for the string *data* (or some other word you know will be in your database) returns some hits:

```
$ ./zclient localhost 6668 "NPEM"
"data [1,1016]"
```

Finally a global spatial search should return every record in the database:

```
$ ./zclient localhost 6668 "NPEM" "90
-180 -90 180[1,2060]"
```

8.6. Registering with NSDI Clearinghouse

Registering your node is a five-step process. Screen-shots of each step are given below.

For on-line documentation, see:

- Guide to Clearinghouse Node Creation http://www.fgdc.gov/dataandservices/clearinghouse_qanda,
- Install and configure Isite <http://clearinghouse4.fgdc.gov/ftp/unixinstall1.html>.

To begin the registration process, go to <http://registry.gsdi.org/>, and click on the “Add Your Node” icon (see Fig. 8.4).

For the first step, supply either a valid DNS name or an IP address (Fig. 8.5). Click on the Next-> Perform Server Check button.

Next, a test is performed to validate your server name and compatibility. Enter the communication port number for your Z39.50 server (Fig. 8.6). Click on the Next-> Perform Z39.50 Check button.



Fig. 8.4 Entry point for registering your node with the Clearinghouse.

[Add Your Node](#)
[Browse Registry](#)
[Browse Diagnostics](#)
[Modify Your Node](#)
[International Server Status](#)
[Domestic Server Status](#)
[Canadian Server Status](#)
[Home](#)

THE CLEARINGHOUSE REGISTRY

Federal Geographic Data Committee (FGDC)

[Add Your Node Tutorial](#)

Step 1 of 5: Add Your Node to the Clearinghouse (Server Check)

This form is used to collect nominations of Z39.50 Version 2 or 3 servers which support the GEO profile. Entries made via this form are made accessible to the Clearinghouse Search Forms in HTML and Java.

Enter the Host Name of your Z39.50 server:
 (Example: registry.gsdi.org) Host Name:

If you do not have a Host Name, select this option and enter your Machines IP Address in the Host Name field.

No Host Name:

Fig. 8.5 Step 1: Enter the name or IP address of the Z39.50 server.

[Add Your Node](#)
[Browse Registry](#)
[Browse Diagnostics](#)
[Modify Your Node](#)
[International Server Status](#)
[Domestic Server Status](#)
[Canadian Server Status](#)
[Home](#)

THE CLEARINGHOUSE REGISTRY

Federal Geographic Data Committee (FGDC)

[Add Your Node Tutorial](#)

Step 2 of 5: Add Your Node to the Clearinghouse (Server Check)

This form is used to collect nominations of Z39.50 Version 2 or 3 servers which support the GEO profile. Entries made via this form are made accessible to the Clearinghouse Search Forms in HTML and Java.

Status: Your server at codline.afsc.noaa.gov has been located and tested via the internet.
 Please continue the clearinghouse registration by providing the following information.

Enter the Port Number of your z39.50 server
 (example: 210)

Port Number:

Fig. 8.6 Step 2: Submit server port number.

[Add Your Node](#)
[Browse Registry](#)
[Browse Diagnostics](#)
[Modify Your Node](#)
[International Server Status](#)
[Domestic Server Status](#)
[Canadian Server Status](#)
[Home](#)

THE CLEARINGHOUSE REGISTRY

Federal Geographic Data Committee (FGDC)

[Add Your Node Tutorial](#)

Step 3 of 5: Add Your Node to the Clearinghouse (Search Test)

This form is used to collect nominations of Z39.50 Version 2 or 3 servers which support the GEO profile. Entries made via this form are made accessible to the Clearinghouse Search Forms in HTML and Java.

Status: Your server at codline.afsc.noaa.gov and Port 6668 has been successfully tested. Please continue the clearinghouse registration by providing the following information.

Enter the Database Name of the z39.50 server
 Database Name:

Enter a valid word contained within the title of one of the metadata records
 Title Search:

Enter a valid word contained within the full text of one of the metadata records
 Full Text Search:

Perform Spatial Search Test:

Un-check this box ONLY if your collection has ZERO records with bounding coordinates in them. Unchecking this box will skip a spatial test on this collection.

Fig. 8.7 Step 3: Test search function.

Now add the name of your database and two search words, as shown in Fig. 8.7. Click on the Next->Perform Z39.50 Search Tests button.

Inspect the status report (Fig. 8.8) of the search test made using your database name and your two search phrases or words. Click on Next-> Continue Registration.

For the final step, supply information that describes your server (Figs. 8.9–8.13). Please fill out all fields. Add “PICES” to the beginning of the title of the server. In this way all PICES servers will be seen at the same time in the alphabetized list of active servers connected to the Clearinghouse.

To continue, click on the Submit Node Entry to Registry button and you will be asked to enter a password for future modifications or updates.

That completes the registration process. Optionally, you can register your node with Geodata.gov by clicking on the Modify Your Node icon at the top of the screen, entering your node name, contact name and password (Fig. 8.13) and clicking on the Register on Geodata.gov button.

Status of nodes can be determined by browsing <http://registry.fgdc.gov/serverstatus/> (Fig. 8.14).

The Clearinghouse Registry
Federal Geographic Data Committee (FGDC)

[Add Your Node](#) [Browse Registry](#) [Browse Diagnostics](#) [Modify Your Node](#) [International Server Status](#) [Domestic Server Status](#) [Canadian Server Status](#) [Home](#)

[Add Your Node Tutorial](#)

Step 4 of 5: Add Your Node to the Clearinghouse (Test Results)

This form is used to collect nominations of Z39.50 Version 2 or 3 servers which support the GEO profile. Entries made via this form are made accessible to the Clearinghouse Search Forms in HTML and Java.

Status: Your z39.50 server at codline.afsc.noaa.gov on Port [6668](#) with the [NPEM](#) has successfully passed the required clearinghouse search tests with the following results:

- ◆ **fish** Title Search: [74](#) Metadata Record(s) Found.
- ◆ **data** Full Text Search: [2775](#) Metadata Record(s) Found.
- ◆ Geospatial Coordinate Search: [2775](#) Metadata Record(s) Found.
- ◆ Implementation Name: [CNIDR zserver](#)
- ◆ Implementation Version: [2.2.4-SUSE92-Linux, Release 2005042001](#)
- ◆ GEO Value: [GEO22](#)

Please continue the clearinghouse registration by continuing to the next page.

[Next -> Continue Registration](#)

Fig. 8.8 Step 4: Status report.

[Add Your Node](#)
[Browse Registry](#)
[Browse Diagnostics](#)
[Modify Your Node](#)
[International Server Status](#)
[Domestic Server Status](#)
[Canadian Server Status](#)
[Home](#)

THE CLEARINGHOUSE REGISTRY

Federal Geographic Data Committee (FGDC)

Step 5 of 5: Add Your Node to the Clearinghouse (Registration)

This form is used to collect the remaining required information for Clearinghouse registration. Entries made via this system are made accessible to the Clearinghouse Search Forms in HTML and Java.

Server Description

- ◆ Long Title of Server: (to be used on select list)
- ◆ Short Title of server: (to be displayed on selection map)
- ◆ Abstract: (Describe the collection criteria, data coverage, themes, and formats of information served by this 239.50 service. This field may be used by clients to evaluate the types of information and coverage to perform a selective query or to evaluate query results.)

Fig. 8.9 Step 5a: Supply title of server (long and short versions) and provide an abstract that describes the service.

◆ Cost information: (Enter cost basis description)

◆ Server Categories:

Agriculture and Farming:	<input type="checkbox"/>	Atmospheric and Climatic Data:	<input checked="" type="checkbox"/>
Biologic and Ecologic Information:	<input checked="" type="checkbox"/>	Environmental Monitoring and Modeling:	<input checked="" type="checkbox"/>
Business and Economic Information:	<input type="checkbox"/>	Administrative and Political Boundaries:	<input type="checkbox"/>
Earth Surface Characteristics and Land Cover:	<input checked="" type="checkbox"/>	Elevation and Derived Products:	<input type="checkbox"/>
Geologic and Geophysical Information:	<input checked="" type="checkbox"/>	Health and Disease:	<input type="checkbox"/>
Ocean and Estuarine Resources and Characteristics:	<input checked="" type="checkbox"/>	Cultural and Demographic Information:	<input type="checkbox"/>
Images and Photographs:	<input type="checkbox"/>	Fresh Water Resources and Characteristics:	<input type="checkbox"/>
Cadastral and Legal Land Descriptions:	<input type="checkbox"/>	Utility Distribution Networks:	<input type="checkbox"/>
Facilities, Buildings and Structures:	<input type="checkbox"/>	Geodetic Networks and Control Points:	<input type="checkbox"/>
Transportation Networks and Models:	<input type="checkbox"/>	Base Maps, Scanned Maps and Charts:	<input type="checkbox"/>
Tourism and Recreation:	<input type="checkbox"/>		

◆ This Clearinghouse Node contains metadata that provide the following access to data through the Online Linkage field through a WebURL:

Data download via ftp:	<input type="checkbox"/>
Data download via http:	<input checked="" type="checkbox"/>
Data ordering via http form:	<input type="checkbox"/>
Data information via http form:	<input type="checkbox"/>
Data information via offline access:	<input checked="" type="checkbox"/>
OpenGIS Web Mapping Service:	<input type="checkbox"/>
OpenGIS Web Feature Service:	<input type="checkbox"/>

Fig. 8.10 Step 5b: Provide cost basis, server categories and methods you use to link metadata to data.

Server Host Information

This block of information is required to enroll your server in the list of GEO servers. The hostname, port, and database name information is used to automatically generate select lists for the Java and HTML query interfaces.

- Full Internet Host Name: **codline.afsc.noaa.gov**
- Full Internet IP Number: **161.55.120.200**
- TCP/IP Port Number for Z39.50 Service: **6668**
- Data index name (Z39.50 Database Name): **NPEM**
- Valid word in a database record title: **fish**
- Valid word found in a database record text: **data**
- Perform Spatial Search: **Yes**
- GEO Value: **GEO22**
- Software Implementation: **CNIDR zserver**
- Software Version: **2.2.4-SUSE92-Linux, Release 2005042001**

- Platform:
- Full WebURL of related Web site:
- Server latitude in Decimal Degrees (DD.D):
- Server longitude in Decimal Degrees (-DDD.D):
- What Country best describes the significant coverage of your database? Note: If your database has significant coverage for multiple countries you may need to select either a continent or global selection from the pulldown list:
- Data Coverage:
- Does this collection include data that covers the United States in part or full? Yes
- Collection Scope ([What is this?](#))

Predominant Geographic Extent of Data Served

This rectangular footprint defines the general area for which you manage geographic metadata and data. This information may be used in a preemptive query to identify suitable servers from a long list of servers.

min longitude **max longitude**
min latitude **Max latitude**

Fig. 8.11 Step 5c: Provide server host information and geographic coverage summary.

Server Contact Information

Contact information to permit clients and servers to contact you with service problems or queries.

- Name:
- Organization:
- Address:
- City:
- State or Province:
- Postal Code:
- Country:
- E-mail address:
- User Support Hours of Service:
- Telephone:
- Fax:

Fig. 8.12 Step 5d: Provide contact information.

THE CLEARINGHOUSE REGISTRY

Federal Geographic Data Committee (FGDC)

[Modify Your Node Tutorial](#)

Modify the Registry

Select the name of the node, contact name and password for the node that you wish to modify. If you provide the correct information, you will be able to modify the node registration information.

Node:

Contact:

Password:

Fig. 8.13 Step 5e: Optional registration at Geodata.gov.

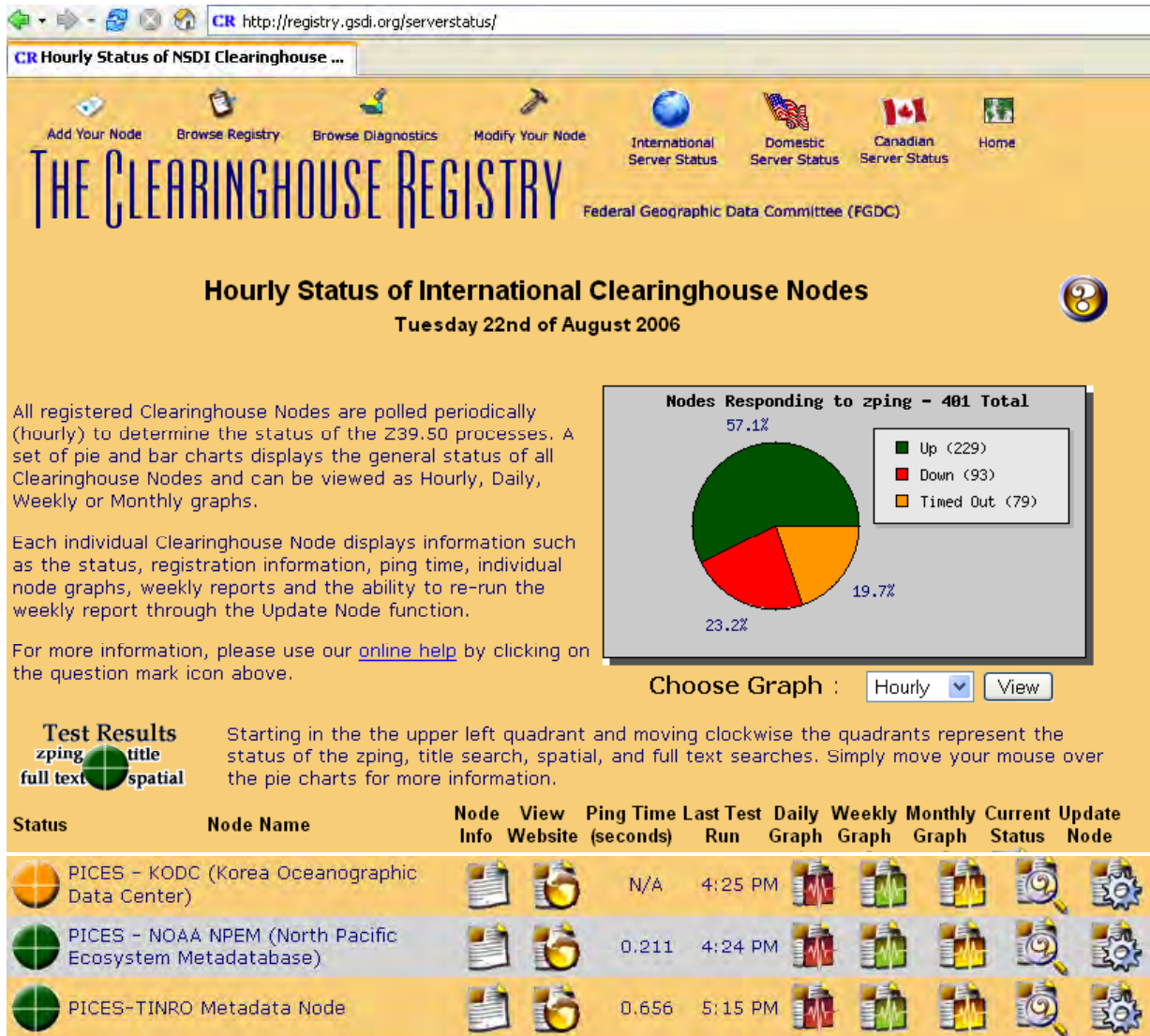


Fig. 8.14 Hourly status of international clearinghouse nodes (PICES nodes shown).

8.7. Security issues

The diagram in Fig. 8.15 shows how we have arranged our computer hardware for the PICES–NPEM node. Recall that the communication port on the NPEM node Isite server has to be open to the Internet. This presents a serious security problem. Our solution was to open the port (6668) on the firewall, but place the NPEM node server outside our firewall but inside the DMZ (Demilitarized zone), still under control of the firewall router.

Our metadata reside in a relational database on a server inside the firewall. Records from the metadata relational database are read, one-at-a-time, and converted into an XML record with a

Java script (listed in Appendix 15.7). At the end of this process, there is an XML file for each record in the metadata base. These XML files are transferred to the NPEM node server via FTP as updates as additions are made. After each update, the XML files are re-indexed on the NPEM node. The NPEM node and the NSDI Clearinghouse Gateway communicate via Z39.50 protocols. Users can access the NPEM metadata base server through our custom search interface <http://www.pmel.noaa.gov/np/mdb/> using HTTP or they can search the same metadata records through the NSDI clearinghouse Gateway along with other shared Gateway holdings via HTTP and Z39.50.

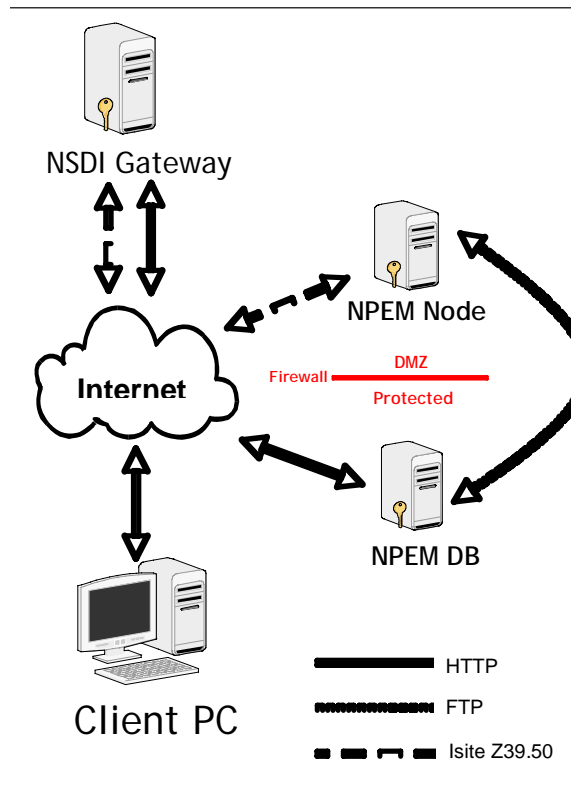


Fig. 8.15 Arrangement of computer hardware for a secure PICES–NPEM node.

9. SEARCH TUTORIAL AND EXAMPLES

Two search examples are given. In the first, an example search using the legacy NSDI Clearinghouse search interface is presented. This eventually will be replaced by the GeoNetwork search interface which is the second example.

9.1. Legacy NSDI Clearinghouse search interface

Begin a search for metadata by opening an Internet browser and connecting to the FGDC search page <http://clearinghouse3.fgdc.gov/> (Fig. 9.1).

There are six entry points or gateways that provide exactly the same lists of servers or Clearinghouse nodes. For this example, we will choose the NOAA CSC Clearinghouse Gateway.

In Fig. 9.2 are the three different types of search interfaces you can use to search for geospatial data: the search wizard, the map interface with place names, and the place names without the use of Java applet. We will use the NSDI Search Wizard to “smart select” servers and data.

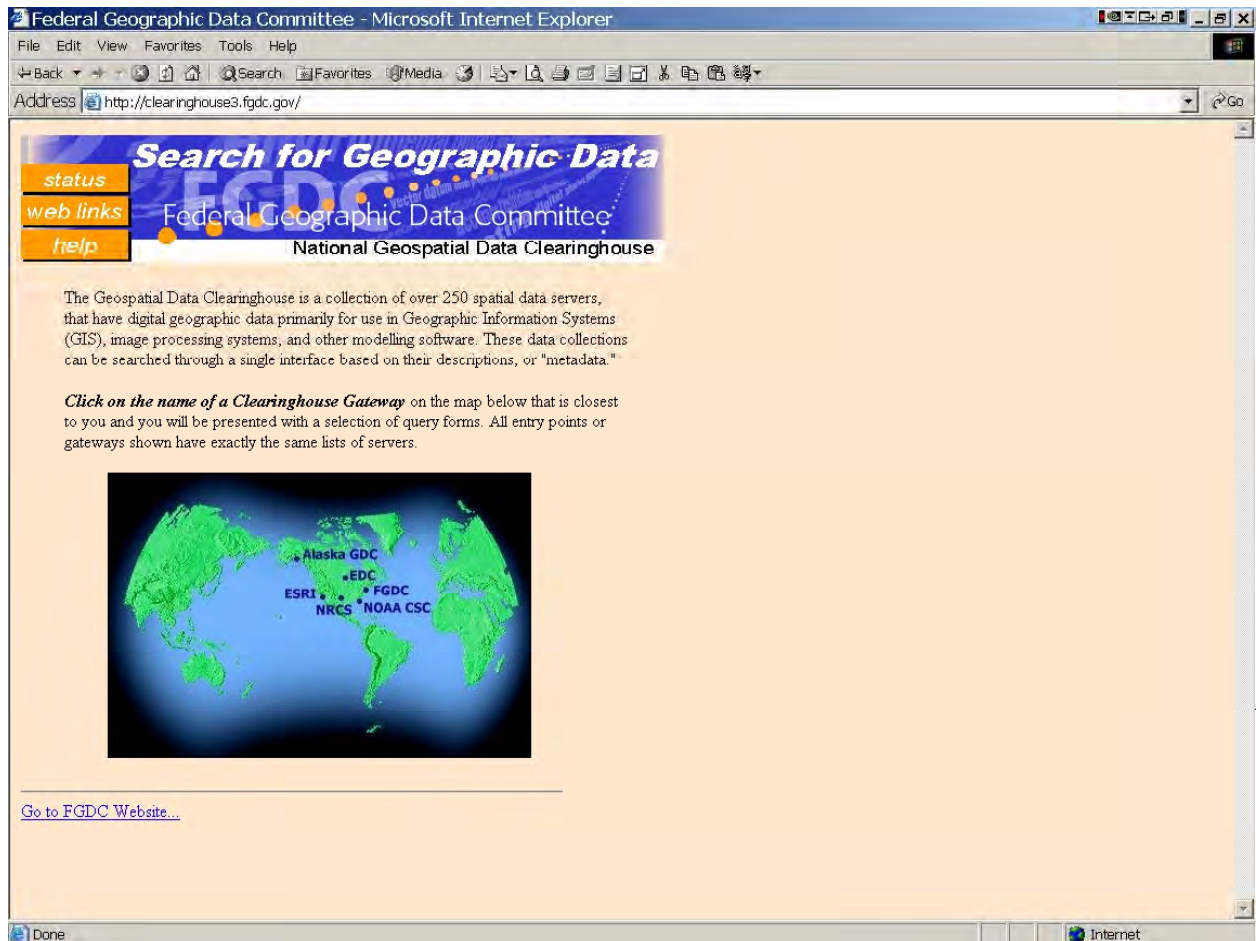


Fig. 9.1. FGDC NSDI Clearinghouse Network page.

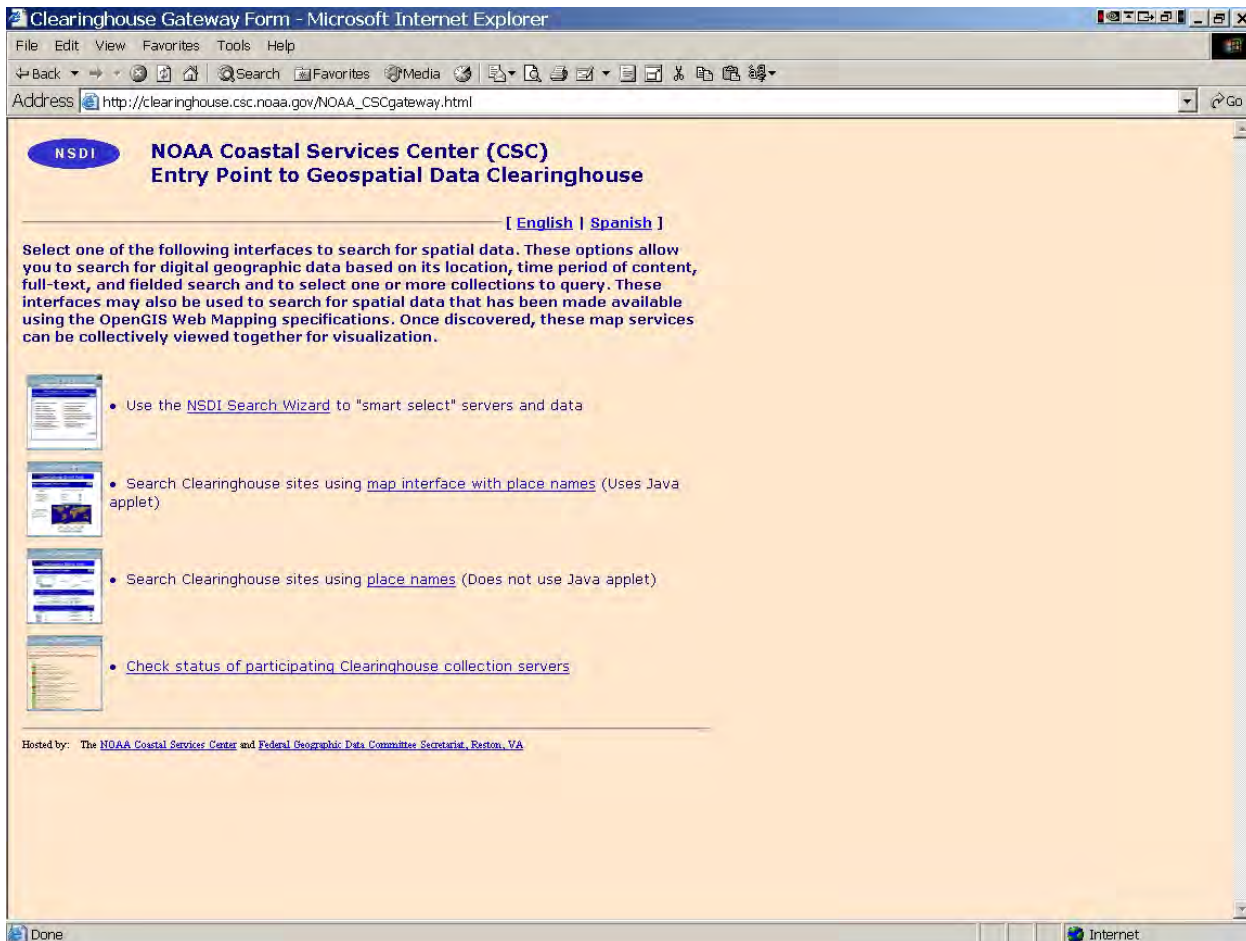


Fig. 9.2 NOAA Coastal Services Center (CSC) Entry Point to Geospatial Data Clearinghouse.

In Fig. 9.3, the Clearinghouse Search Wizard provides three methods to find servers with data. Choose all topics, any one of up to four topics, or all of up to four topics that are listed. We will choose the topic “Ocean and Estuarine Resources and Characteristics” by checking the provided check box next to the topic and then click the “Next” button to continue with the search.

The window shown on Fig. 9.4 allows you to define the geographic area of coverage. This search page is also the second search option on the Gateway page. It allows search of the Clearinghouse sites using the map interface with place names. For PICES metadata, enter the bounding coordinates of the area that is of interest to you. The general northern North Pacific bounding coordinates are 90° North, 30° South, 110° West, and 110° East. You have the option of

using the interactive map or entering the coordinates directly. Once you have identified your search area, click the “Next” button.

Figure 9.5 lists all servers that contain metadata within the area you identified in your geographic area option. Select all the data servers to search for metadata. Click on the “Clear All” button to uncheck all the servers. Since we want to search for and through PICES member country metadata, choose the PICES servers or Clearinghouse nodes listed on the page by checking the boxes next to the nodes: “PICES-KODC”, “PICES-NOAA NPEM”, and “PICES-TINRO”. To learn more about the data server in which you are interested, click on the “More Information” link next to the data server name. When you are finished selecting the server and ready to continue, click “Next”.

Clearinghouse Search Wizard

Select a Topic of Interest Help...

Select all topics or choose up to four topics of interest from the categories listed. This will help select appropriate Clearinghouse Servers to query.

Find servers that have data on: All topics Any one of up to four topics All of up to four topics

<input type="checkbox"/> Administrative and Political Boundaries	<input type="checkbox"/> Facilities, Buildings and Structures
<input type="checkbox"/> Agriculture and Farming	<input type="checkbox"/> Geodetic Networks and Control Points
<input type="checkbox"/> Atmospheric and Climatic Data	<input type="checkbox"/> Geologic and Geophysical Information
<input type="checkbox"/> Base Maps, Scanned Maps, and Charts	<input type="checkbox"/> Human Health and Disease
<input type="checkbox"/> Biologic and Ecologic Information	<input type="checkbox"/> Imagery and Aerial Photographs
<input type="checkbox"/> Business and Economic Information	<input type="checkbox"/> Inland Water Resources and Characteristics
<input type="checkbox"/> Cadastral and Legal Land Descriptions	<input checked="" type="checkbox"/> Ocean and Estuarine Resources and Characteristics
<input type="checkbox"/> Earth Surface Characteristics & Land Cover	<input type="checkbox"/> Society, Cultural and Demographic Information
<input type="checkbox"/> Elevation and Derived Products	<input type="checkbox"/> Tourism and Recreation
<input type="checkbox"/> Environmental Monitoring and Modeling	<input type="checkbox"/> Transportation Networks and Models
	<input type="checkbox"/> Utility Distribution Networks

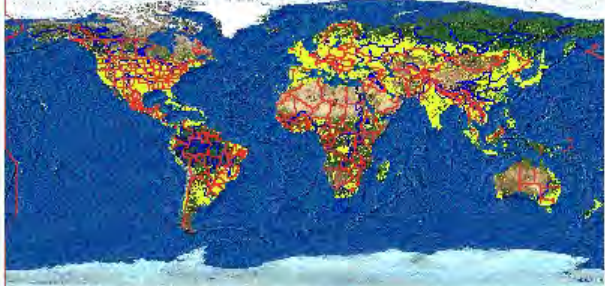
Find All Data
 Only Find Data With Interactive Web Mapping Services (OGC)

Fig. 9.3 Clearinghouse Search Wizard. Begin your search.

Define the Geographic Area of Coverage

Help...

Select either a placename from the list, drag a rectangle on the map (zoom in if you like), or enter coordinates, until the region you are interested in is shown as a yellow rectangle in the map frame. This will help limit your geographic area of search across the Clearinghouse collections.



United States
 International

Afghanistan
 Africa
 Albania
 Algeria
 Angola
 Antarctica
 Antarctica
 Arctic
 Argentina

North
 West East
 South

Fig. 9.4 Specify the geographic area of coverage.



Fig. 9.5 Select the Data Server(s) to search.

Specify Time Period of Content Help...

Specify a date or date range by selecting one of the methods below.

Don't search based on time period

Get data whose date is the date

Get data from through

Search in Full-Text (Any) or by Field Help...

If you selected search topics earlier, sample search words are provided below to assist in finding data. You may modify these words, the fields that will be searched, or the AND/OR condition between the words.

Search for: in the field

in the field

in the field

in the field

Additional Configuration Settings Help...

Maximum number of records to show on each page:

Fig. 9.6 More search options: time period of content and field text search.

The window shown by Fig. 9.6 presents the option of identifying the time period of content and entering text to search for. With the time period of content, you can choose to look for data that begins at a certain date or provide a range of dates.

With the “Full-Text” or “by Field” search, enter the words that you want to search for in your metadata. You have the option of using the And/Or conditions to use more than one word in multiple fields.

You can also choose the maximum number of search result records to see on a page. You are now done entering search options. Your next move is to click the “Search” for metadata button at the bottom of the page.

On the search results page, e.g., Fig. 9.7, each Clearinghouse node is listed with its search status and number of metadata record-matched results. Each Clearinghouse node is clickable when the search is successful and has returned a number of results. Just click on the node to view a list of matched metadata.

Clicking on a node brings up a list of metadata titles (Fig. 9.8). You can review each metadata record either by its summary or full description. Click on the “Summary” link to view a metadata summary page (Fig. 9.9) or click on the “Full” link to view a metadata full description page (Fig. 9.10).

Done with search!

Select the links below to view matches by database.

Database	Status	# Results
PICES - KODC (Korea Oceanographic Data Center)	Search Successful	0
PICES - NOAA NPEM (North Pacific Ecosystem Metadatabase)	Search Successful	223
PICES-TINRO Metadata Node	Search Successful	5

Having troubles using this software to find what you need?
[Click here to send us a message.](#)

[<< Back](#)

Fig. 9.7 Metadata search results page.

Brief Metadata Information and Links

You are currently viewing 1-30 out of 223 matches from database 'PICES - NOAA NPEM (North Pacific Ecosystem Metadatabase)'.

[All](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [Next](#)

Velocities of outlet glaciers, ice streams, and ice shelves, Antarctica, from satellite images
[\[Summary\]](#) [\[Full\]](#)

Sea Ice and Polar Climate in the NCAR CSM [\[Summary\]](#) [\[Full\]](#)

Passive microwave remote sensing of thin sea ice using principal component analysis [\[Summary\]](#)
[\[Full\]](#)

Monthly Averaged Sea Ice Concentration Grids [\[Summary\]](#) [\[Full\]](#)

Sea Ice Concentration Grids for the Polar Regions on CD-ROM [\[Summary\]](#) [\[Full\]](#)

Sea ice prediction modeling [\[Summary\]](#) [\[Full\]](#)

Atmospheric boundary layer structure and drag coefficients over sea ice [\[Summary\]](#) [\[Full\]](#)

Response of sea ice drift to wind forcing on the northeastern Newfoundland shelf [\[Summary\]](#) [\[Full\]](#)

Arctic and Antarctic Research Institute (AARI) 10-day Sea Ice Charts for the Arctic Ocean
[\[Summary\]](#) [\[Full\]](#)

Modeling of Antarctic Sea Ice in a General Circulation Model [\[Summary\]](#) [\[Full\]](#)

Modeling Sea Ice as a Granular Material, Including the Dilatancy Effect [\[Summary\]](#) [\[Full\]](#)

Polar Microwave Sea Ice Data from Nimbus - 5 [\[Summary\]](#) [\[Full\]](#)

Bering AIR-SEA-ICE Study (Basics) [\[Summary\]](#) [\[Full\]](#)

Tracking of Ice Edges and Ice Floes by Wavelet Analysis of SAR Images [\[Summary\]](#) [\[Full\]](#)

Monthly Averaged Sea Ice Concentration Grids for the Polar Regions [\[Summary\]](#) [\[Full\]](#)

Fig. 9.8 A list of metadata titles with summary and full metadata description links.

As shown by Fig. 9.9, the summary page will provide you with the metadata title, time period of content, and the bounding coordinates. You also have the option of seeing the full description page by clicking on the “Full” link at the top corner of the page, or view previous or next summary metadata page with the navigation links at the top corner of the page.

Figure 9.10 illustrates a full metadata description page. Scroll down the page to see the whole page. You also have the option of choosing the viewable format of the metadata record with links at the top of the page under the metadata title.

9.2. New GeoNetwork search interface

The NSDI Clearinghouse has begun a transition from their old proprietary legacy interface (See Section 11) to a new opensource interface called GeoNetwork. GeoNetwork is similiar to the

legacy interface but offers more functionality, yet it is still based on the tried and true Z39.50 communication protocol. Users can download the GeoNetwork software via the Internet at <http://geonetwork-opensource.org/>.

To search with GeoNetwork’s interface, go to <http://edclxs25.cr.usgs.gov:8080/geonetwork/srv/en/main.home?extended=on&remote=on&attrset=geo&hitsPerPage=10> (Fig 9.11), enter search criteria, select a server or group of servers, and put your mouse over the search gear on the right edge of the page and click.

This brings up a listing of records found (Fig. 9.12). Click on one of these and more deailed information is displayed in a default view (Fig. 9.13), an advanced view (Fig. 9.14) or an XML view (Fig. 9.15).

Summary Metadata Information

You are currently viewing record 5 out of 223 from database 'PICES - NOAA NPEM (North Pacific Ecosystem Metadatabase)'. [Full](#) [Previous](#) [Next](#)



Title: Sea Ice Concentration Grids for the Polar Regions on CD-ROM
Time Period of Content: 1987-07-09 to 1991-12-31
West: 163.000000 **East:** -157.000000 **North:** 65.500000 **South:** 52.000000

Having troubles using this software to find what you need?
[Click here to send us a message.](#)

Fig. 9.9 Summary metadata page.

Sea Ice Concentration Grids for the Polar Regions on CD-ROM

Metadata also available as - [\[Parseable text\]](#) - [\[SGML\]](#)

Metadata:

- [Identification Information](#)
- [Distribution Information](#)
- [Metadata Reference Information](#)

Identification Information:

Citation:

Citation Information:

Originator:

Department of Commerce (DOC), National Oceanic and Atmospheric Administration (NOAA), National Environmental Satellite, Data, and Information Service (NESDIS), National Climatic Data Center (NCDC)

Publication Date: 2001

Title: Sea Ice Concentration Grids for the Polar Regions on CD-ROM

Geospatial Data Presentation Form: vector digital data, map

Publication Information:

Publication Place: Boulder, CO

Publisher:

National Environmental Satellite Data and Information Service, National Climatic Data Center

Fig. 9.10 Full metadata description.

Portal Geoespacial de America del Sur

Home | Last results | Contact us | Links | About | Help

Find Interactive Maps, GIS datasets, Satellite Imagery and Related Applications

Title:

Abstract:

Free Text:

Keywords:

Location: - Any -

Profile:

Server:

Timeout:

Hits per page:

Simple search

Local search

Search

GeoNetwork's purpose is:

- To improve access to and integrated use of spatial data and information
- To support decision making
- To promote multidisciplinary approaches to sustainable development
- To enhance understanding of the benefits of geographic information

GeoNetwork opensource allows to easily share geographically referenced thematic information between different organizations. For more information please contact: anthony@usgs.gov or send us [feedback](#)

Fig. 9.11 GeoNetwork search interface.

Portal Geoespacial de America del Sur

Home | Last results | Contact us | Links | About | Help

English | Français | Español

Aggregate Results matching search criteria : 57

1 - Surficial Geology of Gates of the Arctic- Formations and Deposits	Alaska Geospatial Data Clearinghouse	No preview available
2 - CIK (Cook Inlet/Kenai Peninsula)- Cook Inlet Ice Conditions	Alaska Geospatial Data Clearinghouse	No preview available
3 - Glaciers, 1:63,360	Alaska Geospatial Data Clearinghouse	No preview available
4 - Glacier Extent and Margins - 19th Century - Kenai Fjords National Park	Alaska Geospatial Data Clearinghouse	No preview available
5 - Exit Glacier- Harding Ice Field Trail	Alaska Geospatial Data Clearinghouse	No preview available
6 - Glacier Extent and Margins - mid 1980's - Kenai Fjords National Park	Alaska Geospatial Data Clearinghouse	No preview available
7 - Geology of the Seldovia Quadrangle - KBEC Database	Alaska Geospatial Data Clearinghouse	No preview available
8 - Exit Glacier- Trail Anchors	Alaska Geospatial Data Clearinghouse	No preview available
9 - Chugach National Forest: Geologic Map from USGS Branch of Alaska Geology (GEOLOGY)	Alaska Geospatial Data Clearinghouse	No preview available

Fig.9.12 GeoNetwork search results.

Portal Geoespacial de America del Sur

Home | Last results | Contact us | Links | About | Help

English | Français | Español

1 - Alaska Geospatial Data Clearinghouse

Default view | Advanced view | XML view

Identification Information

Citation Information

Title	Surficial Geology of Gates of the Arctic- Formations and Deposits
Edition	Unknown
Geospatial Data Presentation Form	Map

Time Period of Content

Beginning Date	19750101.000000
Ending Date	19871001.000000

Status

Maintenance and Update Frequency	As needed
----------------------------------	-----------

Bounding Coordinates

West Bounding Coordinate	-157.369700
East Bounding Coordinate	-149.139400
North Bounding Coordinate	68.802000
South Bounding Coordinate	66.505000

Fig. 9.13 GeoNetwork detailed search results – default view.

The screenshot displays the advanced view of search results for '1 - Alaska Geospatial Data Clearinghouse'. The interface includes a navigation menu at the top with links for Home, Last results, Contact us, Links, About, and Help. Language options for English, Français, and Español are available, along with fields for Username, Password, and a Login button. On the left, there are view options: Default view, Advanced view (selected), and XML view. The main content area shows a structured metadata view with the following sections:

- ::Metadata**
- ::Identification Information**
- ::Citation Information**
 - Title: Surficial Geology of Gates of the Arctic- Formations and Deposits
 - Edition: Unknown
 - Geospatial Data Presentation Form: Map
- ::Time Period of Content**
- ::Time Period Information**
 - ::Range of Dates/Times**
 - Beginning Date: 19750101.000000
 - Ending Date: 19871001.000000
- ::Status**
 - Maintenance and Update Frequency: As needed
- ::Spatial Domain**
 - ::Bounding Coordinates**
 - West Bounding Coordinate: -157.369700
 - East Bounding Coordinate: -149.139400

Fig. 9.14 GeoNetwork detailed search results – advanced view.

The screenshot displays the XML view of the search results for '1 - Alaska Geospatial Data Clearinghouse'. The interface is identical to the advanced view, but the main content area shows the corresponding XML metadata structure:

```

<?xml version="1.0" encoding="UTF-8"?>
<metadata>
  <idinfo>
    <citation>
      <citeinfo>
        <title>Surficial Geology of Gates of the Arctic- Formations and Deposits</title>
        <edition>Unknown</edition>
        <geofom>Map</geofom>
      </citeinfo>
    </citation>
    <timeperd>
      <timeinfo>
        <rngdates>
          <begdate>19750101.000000</begdate>
          <enddate>19871001.000000</enddate>
        </rngdates>
      </timeinfo>
    </timeperd>
    <status>
      <update>As needed</update>
    </status>
    <spdom>
      <bounding>
        <westbc>-157.369700</westbc>
        <eastbc>-149.139400</eastbc>
        <northbc>68.802000</northbc>
        <southbc>66.505000</southbc>
      </bounding>
    </spdom>
  </idinfo>
</metadata>
  
```

Fig. 9.15 GeoNetwork detailed search results – XML view.

10. CHALLENGES

The predominant metadata federation challenge to accomplishing such an ambitious undertaking is funding. Initiatives such as this take considerable resources. The original KODC–NPEM activity was supported directly at the level of \$US 4,000 from PICES, \$US 6,000 from NPEM, and \$US

6,000 from KODC. Both NPEM and KODC contributed approximately \$US 35,000 in matching monies. The MIRC–NPEM project was directly funded at the level of \$US 4,000 from PICES and \$US 20,000 from NSDI-CAP, with \$38,732 being contributed in matching funds.

11. EMERGING STANDARDS

The standards upon which the clearinghouse functionality relies are in a state of change. The present NSDI Clearinghouse legacy search gateway will soon be replaced. The legacy search gateway is built with propriety software that is no longer maintained nor supported by the vendor, Blue Angel Software. The legacy interface will eventually be replaced with GeoNetwork, a user-maintained and open-source solution with similar and enhanced capabilities compared to the legacy interface. The GeoNetwork gateway (beta version of the new search gateway that is in development) will be implemented in six months to a year. Both the legacy interface and the new GeoNetwork interface rely on the proven Z39.50 communication protocol.

The final change concerns the way metadata are described. The old method used FGDC standards. While proven and well known, this standard has difficulty with biological data. A new international metadata standard is emerging, ISO 19115, which was built to be compatible with FGDC and to address the deficiencies in describing biological data. Translators that convert from FGDC to ISO 19115 should be easily available.

These and other probable changes will lead to modifications in PICES Metadata Clearinghouse interface standards and will require changes for existing and future Clearinghouse servers.

12. FUTURE ACTIVITY

Future plans include the participation of PICES member countries not already federated (*i.e.*, Canada and China) into a PICES Metadata Federation. Such activity may be supported through future 2007 NSDI-CAP funding possibilities. At the 2006 TCODE meeting, a request for Phase III funding (\$4K) to bring China or Canada into the Federation was submitted (pending successful funding application elsewhere).

With the move of the U.S. metadata clearinghouse interface from proprietary to open-source software, the potential exists for PICES to adopt the open-source standard, federate its metadata

internally and sever its direct relationship with the U.S. Clearinghouse nodes. We conclude, that at some point, it may become desirable for PICES to host its own clearinghouse separate from FGDC NSDI.

Other future developments include mirroring a PICES Clearinghouse node and looking into options for translating web pages hosted by non-English speaking PICES partners into their native language. Because of the language barrier, this latter idea would certainly open up PICES metadata to a wider audience than is now available.

13. ACKNOWLEDGMENTS

Many organizations and individuals contributed directly or indirectly to the work described in this report. PICES and the PICES Technical Committee on Data Exchange (TCODE) were long-standing and primary sponsors. NOAA's ESDIM program provided the initial funds to begin the Bering Sea Ecosystem Metadatabase and the FGDC and NSDI both have provided excellent technical support. The laboratories and agencies of all participants generously donated the time of their employees (through matching contributions

to grant applications) so that they could devote some of their time to this project (NOAA's FOCI, AFSC and PMEL, Republic of Korea's KODC, and NFRDI, and Japan's MIRC) and we would like to recognize this important contribution. Funding from the NSDI-CAP Program is also gratefully recognized. This report is contribution 3038 from NOAA/Pacific Environmental Laboratory and contribution EcoFOCI-0624 to NOAA's Ecosystems and Fisheries-Oceanography Coordinated Investigations.

14. REFERENCES

- Federal Geographic Data Committee. 1998. Content Standard for Digital Geospatial Metadata (revised June 1998), FGDC-STD-001-1998. Federal Geographic Data Committee, Washington, D.C., 90 pp.
- Hillmann, D. 2000. Using Dublin Core. <http://dublincore.org/documents/usageguide/>.
- Macklin, S.A., and B.A. Megrey, 2004. The North Pacific Ecosystem Metadatabase. *PICES Press 12*, No. 2: 32–35.
- Megrey, B.A. and Macklin, S.A. 1998. Bering Sea ecosystem biophysical metadatabase: a collaborative research tool for fisheries oceanography and ecosystem investigations. *PICES Press 6*(1): 37–40.
- Michener, W.K., Brunt, J.W., Helly, J.J., Kirchner, T.B. and Stafford, S.G. 1997. Nongeospatial metadata for the ecological sciences. *Ecological Applications 7*(1): 330–342.
- National Research Council. 1996. *The Bering Sea Ecosystem*. National Academy Press, Washington, D.C., 317 pp.
- PICES. 2004. *Marine Ecosystems of the North Pacific*. PICES Special Publication 1, ISSN 1813-8527, ISBN 1-897176-00-7, http://www.pices.int/publications/special_publications/NPESR/2005/npesr_2005.aspx, 278 pp.
- Weibel, S. 2000. The Dublin Core Metadata Initiative: the Frankfurt Focus and the Year 2000. *Zeitschrift für Bibliothekswesen und Bibliographie: Organ des Vereins Deutscher Bibliothekare und des Vereins der Diplombibliothekare an wissenschaftlichen Bibliotheken 47*(1): 3–13.
- Weibel, S., Godby, J., Miller, E. and Daniel, R. 1995. OCLC/NCSA Metadata Workshop Report. OCLC/NCSA. <http://dublincore.org/workshops/dc1/report.shtml>.

15. ACRONYMS

AFS	American Fisheries Society
AFSC	Alaska Fisheries Science Center, NOAA
AGU	American Geophysical Union
AML	Agent Markup Language
ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
ASP	Active Server Pages
CAP	Cooperative Agreements Program
CDMP	Climate Data Modernization Program
CIIMMS	Cook Inlet Information Management and Monitoring System
CNS	Chew 'n Spit
CSC	Coastal Services Center, NOAA
CSDGM	Content Standard for Digital Geospatial Metadata
CSR	Cruise Summary Report
DC	Dublin Core
DIF	Directory Interchange Format
DINRAC	Data and Information Network Regional Activity Center
DMZ	Demilitarized Zone
DNS	Domain Name System
DTD	Document Type Definition
EML	Ecological Metadata Language
ESDIM	Environmental Services Data Information Management
ESRI	Environmental Systems Research Institute, Inc.
FGDC	Federal Geographic Data Committee
FOCI	Fisheries-Oceanography Coordinated Investigations
FTP	File Transfer Protocol
FWS	U.S. Fish and Wildlife Service
GCMD	Global Change Master Directory
GILS	Government Information Locator Service
GIS	Geographic Information Systems
GOOS	Global Ocean Observing System
GSDI	Global Spatial Data Infrastructure
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
ICES	International Council for the Exploration of the Sea
ILRT	Institute for Learning and Research Technology, UK
IOC	Intergovernmental Oceanographic Commission, UNESCO
IP	Internet Protocol
ISO	International Organization for Standardization
JCOMM	Joint WMO–IOC Technical Commission for Oceanography and Marine Meteorology
JHA	Japan Hydrographic Association
JHOD	Hydrographic and Oceanographic Department, Japan
JISAO	Joint Institute for the Study of the Atmosphere and Ocean
JODC	Japan Oceanographic Data Center
KODC	Korea Oceanographic Data Center

KORDI	Korea Ocean Research and Development Institute
LME	Large Marine Ecosystem
MERMAid	Metadata Enterprise Resource Management Aid
MIRC	Marine Information Research Center
MOMAF	Ministry of Maritime Affairs and Fisheries, Republic of Korea
MOST	Ministry of Science and Technology, Republic of Korea
MP	Metadata Parser
NASA	National Aeronautics and Space Administration
NBII	National Biological Information Infrastructure
NFRDI	National Fisheries Research and Development Institute, Republic of Korea
NISO	National Information Standards Organization
NOAA	National Oceanic and Atmospheric Administration
NODC	National Oceanographic Data Center, NOAA
NORI	National Ocean Research Institute, Republic of Korea
NOWPAP	Northwest Pacific Action Plan
NOWPAP RCU	NOWPAP Regional Coordinating Unit
NPEM	North Pacific Ecosystem Metadatabase
NSDI	National Spatial Data Infrastructure
PICES	North Pacific Marine Science Organization
PMEL	Pacific Marine Environmental Laboratory, NOAA
RDF	Resource Description Framework
SAPI	Search Application Program Interface
SGML	Standard Generalized Markup Language
SMMS	Spatial Metadata Management System
TCODE	Technical Committee on Data Exchange
TKME	Tk Metadata Editor
TOS	The Oceanographic Society
WDC	World Data Center
WestPac	IOC Sub-Commission for the Western Pacific
UNEP	United Nations Environment Program
USACE	U.S. Army Corps of Engineers
USDA	United States Department of Agriculture
USGS	United States Geological Survey
UW	University of Washington
XML	Extensible Markup Language
XSD	XML Schema Document
XSL	Extensible Stylesheet Language
XSLT	XSL Transformations
XTME	Xt Metadata Editor

16. APPENDICES

16.1. KODC–NPEM meeting agendas and minutes

16.1.1. Seattle meeting agenda, August 22–23, 2005

Start	Event	Participants
Day 1		
09:00	Introductions and Overview: NPEM and Isite introduction by Megrey	KODC-NPEM, Stabeno
10:00	Walking tour of AFSC and PMEL	KODC-NPEM
11:30	LUNCH	
13:00	NPEM and Isite implementation overview by Klawitter	KODC-NPEM
15:00	Comment and discussion	KODC-NPEM
Day 2		
09:00	Identify federation procedures	KODC-NPEM
11:30	LUNCH	
13:00	Develop proposal for KODC metadata mapping, KODC-NPEM data sharing and advertising strategy	KODC-NPEM
15:30	Prepare for next meeting	KODC-NPEM

16.1.2. Seattle meeting minutes, August 22–23, 2005

A. Attendance

Republic of Korea: KODC (Korea Oceanographic Data Center)

Kyu Kui Jung, KODC, NFRDI
Hae-Seok Kang, KORDI

U.S.A.: NPEM (North Pacific Ecosystem Metadatabase)

Allen Macklin, PMEL, NOAA
Bernard Megrey, AFSC, NOAA
Dan Klawitter, Macrostaff
Kimberly Bahl, JISAO, UW
Phyllis Stabeno, PMEL, NOAA

B. A review of metadata standards used by PICES member national data centers

Country	Data Center	Metadata Standard
Canada	Canada	FGDC
China	China	JCOMM or ?
Japan	JODC	Internal?
Russia	RODC	FGDC?
Republic of Korea	KODC	DIF
USA	GCMD	DIF
USA	NPEM	FGDC
USA	NSDI	FGDC

Notes: NPEM will submit a request to the PICES TCODE Chairman to request all TCODE members to identify their nation's data centers and metadata standard used respectively for the next TCODE meeting in Vladivostok, Russian.

Question: If All PICES member nations' data centers use DIF, should NPEM switch to the DIF standard?

C. Metadata clearinghouses

Identify global metadata clearinghouses:

Metadata	Communication Protocol	Clearinghouse	Associated Cost
FGDC	z39.50	NSDI	No
DIF	?	?	?
FGDC	z39.50?	GSDI	?
?	?	“PICES”	?
?	?	WDC	?

D. Sasakawa proposal advertising strategy:

Who are the beneficiaries:

- Students
- Scientists
- Resource managers
- Legal
- Policy makers
- Strategic planners
- Administrators

ICES, AGU, AFS, TOS, WestPac, GOOS.

- Two brochures:
 1. For Republic of Korea – in detail
 2. Multilingual – less detailed and on web page

Articles:

- In science and agency newsletters,
- Press release,
- Public Service Announcement (in Korean and English) to appropriate outlets.

Things we can produce:

Brochures:

- Send to conferences for distribution with registration materials, e.g., IOC, PICES,

Advertise to the following government institutes and organizations:

In Republic of Korea	In U.S.A.
Ministry of Maritime Affairs and Fisheries (MOMAF) <ul style="list-style-type: none"> • National Fisheries Research and Development Institute (NFRDI) • National Oceanographic Research Institute (NORI) Ministry of Science and Technology (MOST) <ul style="list-style-type: none"> • Korea Ocean Research and Development Institute (KORDI) Universities <ul style="list-style-type: none"> • Cheju National University • Chungnam National University • Chungnam National University • Gyeongsang National University • Inha University • Pukyong National University • Seoul National University 	National Oceanic and Atmospheric Administration (NOAA) U.S. Fish and Wildlife Service (FWS) Universities: <ul style="list-style-type: none"> • Alaska Pacific University – Pacific Rim Fisheries • Oregon State University • Scripps Institution of Oceanography • University of Alaska • University of California, Davis • University of Colorado, Boulder • University of Hawaii • University of Miami • University of Washington • Woods Hole Oceanographic Institution

E. Action Items for KODC and NPEM until next meeting in October

Priority	Tasks (August–October)	NPEM To Do	KODC To Do
1	Translate parent records to English		X
1a	One record to FGDC, register with NSDI		X
1b	Convert to xml	X	X
1c	Obtain server	X	X
1d	Install OS, Isite, <i>etc.</i>	X	X
1e	Test	X	X
2	Are there DIF-based Z39.50 Clearinghouses?	X	X
3	Acquire MORPH routines (metadata translation)	X	X
4	Query TCODE members via Igor Shevchenko	X	
4a	Interested in federating?	X	
4b	Metadata Standard?	X	
4c	WDC, <i>etc.</i> participant?	X	
5	Seek continued funding	X	

Notes: KODC Clearinghouse node – Isite implementation steps: KODC will take one parent record, translate into English, and send it to NPEM to convert to FGDC XML file. KODC will then take the file and implement Isite Z39.50 software package and register with the NSDI Clearinghouse as a clearinghouse node.

Lola Olsen, Responsible NASA Official
NASA/GSFC Code 902, U.S.A.

E-mail: olsen@gcmd.gsfc.nasa.gov

GCMD User Support Office:

Tel.: +301-441-4202 Fax: +301-441-9486

E-mail: gcmduso@gcmd.gsfc.nasa.gov

Discussion of DIFmorph software package:

- Product by GCMD/NASA (www: <http://gcmd.gsfc.nasa.gov>),
- Converts DIF to FGDC standard,
- Find out if it is still available,
- Contact Info:

Discussion of possible funding sources for future activity:

- NOAA Arctic Research Institute – John Calder – focus on the northern latitudes,
- UN,
- Republic of Korea–U.S. Coop Program – Syd Levitus,
- IPY 2007 money – International Polar Year.

16.1.3. Busan meeting agenda, October 10–11, 2005

Start	Event	Participants
Monday		
09:30	Introductions and Overview	KODC–NPEM
10:30	Walking tour of NFRDI	KODC–NPEM
12:00	LUNCH	
13:30	KODC metadatabase demonstration	KODC–NPEM
15:00	Comment and discussion	KODC–NPEM
16:00	Review progress on tasks	KODC–NPEM

Start	Event	Participants
Tuesday		
09:30	Discussion of the KODC-NPEM federation, including metadata mapping , keywords, <i>etc.</i>	KODC-NPEM
11:30	LUNCH	
13:00	Draw up future plans	KODC-NPEM

16.1.4. Busan meeting minutes, October 10–11, 2005

(Available from Kyu Kui Jung)

16.2. MIRC-NPEM meeting agendas and minutes

16.2.1. Seattle Meeting agenda, August 14-15, 2006

Time	Event	Facilitator
Day 1		
09:00	Introductions; Overview of NPEM and Isite	B. Megrey, A. Macklin
09:45	MIRC's data overview and metadata needs	T. Suzuki
10:30	Walking tour of AFSC and PMEL	All
11:30	LUNCH	
13:00	FGDC Content Standard for Digital Geospatial Metadata	K. Bahl
14:00	NSDI Clearinghouse Network overview	K. Bahl
15:00	Comment, discussion	All
Day 2		
09:00	Isite implementation overview	K. Bahl
11:30	LUNCH	
13:00	MIRC metadata and node development strategy	All
15:30	Preparation for October meeting	All

16.2.2. Seattle meeting minutes, August 14–15, 2006

Participants

Kimberly Bahl, UW/JISAO
Allen Macklin, NOAA/PMEL
Bernard Megrey, NOAA/AFSC
Toru Suzuki, MIRC

Introductions

Overview of NPEM and Isite. Bernard Megrey and Allen Macklin opened the meeting by introducing all participants and presenting a history and overview of the NPEM and PICES federation project. The overview was based on a presentation given at the annual PICES meeting in Vladivostok, October 2005. Two elements are needed for federation: metadata in a standard format and a common communication protocol. The PICES federation uses FGDC metadata standard and Z39.50 protocol.

Locating ongoing funding for NPEM and federation is a challenge

PICES has been very supportive but has limited resources. At present, NOAA funding is unlikely. We have been unsuccessful in attracting money

from international funding organizations. There is a possibility that NOWPAP can provide support for federation. Also, a new 3-year funding cycle at MIRC will start next year.

MIRC's Data Overview and Metadata Needs

Toru Suzuki informed participants of the hierarchical structure of MIRC, JODC, and Japan Hydrographic Association (JHA). He then discussed the varied types of data holdings available through JODC. JODC's data holdings are extremely valuable to scientists working in the North Pacific. They maintain data from several million stations dating back to the early 1800s. Although there is no formal metadatabase, metadata information are extensive in the form of header records, etc.

Walking Tour of AFSC and PMEL

Toru Suzuki was introduced to leadership at AFSC and PMEL and viewed essential laboratory spaces.

FGDC and NPEM Websites: Overview and Search Demo. Kimberly Bahl introduced FDGC and NPEM websites and provided a live metadata

search demonstration. Presently the NSDI Clearinghouse legacy search gateway fails to connect with the NPEM Clearinghouse node. The legacy search gateway is built with propriety software that is no longer maintained and supported by the vendor, Blue Angel Software. The legacy interface will eventually be replaced with GeoNetwork, a user-maintained and open-source solution with similar and enhanced capabilities compared to the legacy interface. The GeoNetwork gateway (beta version of the new search gateway that is in development) will be implemented in 6 months to a year. A demonstration of GeoNetwork (via a demonstration sever) provided successful search results. NPEM will work with NSDI to determine the cause of the legacy search gateway failure.

FGDC Metadata Content Standard

Kimberly Bahl, who received training from FGDC last spring, introduced the FGDC metadata content standard and its sections and elements. This gave Toru Suzuki the metadata rules to write FGDC-compliant metadata records from MIRC information.

Metadata Creation and Validation Tools

Kimberly Bahl demonstrated several open source metadata creation and validation tools, Metavist 2005 and Metadata Parser (MP). These tools allow easy creation of individual metadata records in XML file format (required for any clearinghouse node) and validation that they are FGDC compliant. Participants used Metavist and

MP to create and validate an XML metadata record from a JODC Cruise Summary Report. This completes the first requirement for federation: FGDC compliant metadata.

Isite Implementation

The second requirement for PICES federation is to supply a common communication protocol: Z39.50. Kimberly Bahl provided specific instructions of how to install and configure the Isite application that allows the use of Z39.50 protocol. The Isite software suite is a free, open-source application available from the FGDC website.

MIRC Metadata and Node Development Strategy

This year MIRC will request proposals for three-year projects to begin April 2007. Participants of this meeting will work with Toru Suzuki to develop a MIRC proposal to their funding agency, the Nippon Foundation. The proposal will provide support for ongoing MIRC participation in the PICES federation. A primary component will be development of a MIRC metadatabase. As soon as possible, MIRC will develop a demonstration site using Isite and the XML record created at this meeting. This will be a registered node of the PICES federation.

In addition we discussed the strategies for long term funding and progress reports on this project and overall PICES federation:

Priority	Task	NPEM	MIRC	Due Date
1	Develop proposal to Nippon Foundation to assist with metadata transcription and translation.	X	X	Sept. 30, 2006
1	Configure Isite server, register PICES-MIRC node with NSDI, test sample XML record.		X	Oct. 1, 2006
2	Prepare project meeting minutes.	X		Sept. 1, 2006
2	Check with NOWPAP office in Toyama to see if UNEP has a program that could help fund federation.		X	Sept. 1, 2006
2	Check NODC and CDMP as possible sources of support.	X		Sept. 1, 2006
3	Prepare project report for PICES TCODE	X	X	Sept. 1, 2006
3	Prepare article for PICES Press after Tokyo meeting.	X	X	April 1, 2007
4	Determine proper Japanese affiliation (MIRC/ JHA, JODC)		X	Oct. 1, 2006

Preparation for October meeting: Participants developed the following draft agenda for the next meeting to bring MIRC into the PICES federation. The meeting will be held on Thursday and Friday,

October 19 and 20, in Yokohama and Tokyo, respectively. The agenda will be finalized at the TCODE meeting on Wednesday, October 18.

16.2.3. Tokyo meeting agenda, October 19–20, 2006

Start	Event	Participants
Thursday, October 19, 2006 at Yokohama		
14:00	Review TCODE meeting	MIRC–NPEM–NOWPAP
14:30	Discussion of the MIRC–NPEM federation, including metadata mapping, node registration, <i>etc.</i>	MIRC–NPEM–NOWPAP
15:30	Review progress on tasks	MIRC–NPEM–NOWPAP
Friday, October 20, 2006 at Tokyo		
10:30	Walking tour of Tsukiji Fish Market	MIRC–NPEM–NOWPAP
11:00	Walking tour of JODC/JHOD	MIRC–NPEM–NOWPAP–JODC
12:00	LUNCH	
13:30	Introduction and Overview	MIRC–NPEM–NOWPAP
14:00	PICES-MIRC Metadatabase demo	MIRC–NPEM–NOWPAP
15:00	Comment, discussion	MIRC–NPEM–NOWPAP
16:00	Review tasks and draw up future plans	MIRC–NPEM–NOWPAP
18:00	Dinner in Tokyo (Tsukiji)	
20:00	Train to Yokohama	

16.2.4. Tokyo, meeting minutes, October 19–20, 2006

Participants

Norio Baba, NOWPAP RCU
 Kimberly Bahl, UW/JISAO
 Allen Macklin, NOAA/PMEL
 Bernard Megrey, NOAA/AFSC
 Toru Suzuki, MIRC/JHA

Participants reviewed TCODE meeting on October 18 and discussed with (1) promotion using metadatabase in NOWPAP DINRAC, (2) capacity building, (3) investigation utility of Asian-side metadatabase mirror server, and (4) review and revise a PICES TCODE technical report.

Toru Suzuki introduced that a PICES-MIRC metadatabase registered to NSDI clearinghouse and he reported that there were some problems for installation and configuration on the site. Participants discussed to resolve them.

Participants reviewed the progress on tasks described in Seattle meeting (see Sec. 15.2.2). Toru Suzuki promised to develop a proposal to Nippon Foundation on October 23. He also reported that Isite had been installed on MIRC's site and registered it as a node to NSDI clearinghouse named 'PICES–MIRC metadatabase' on October 18. Kimberly Bahl finished a meeting minutes of Seattle meeting on August 25. Participants discussed about relationship between PICES TCODE and NOWPAP DINRAC activities and new program for capacity building.

Toru Suzuki introduced a summary of MIRC and JODC activities. Allen Macklin asked about running cost of MIRC after termination support by Nippon Foundation and Toru Suzuki answered that MIRC profit from distribution of value adding data products and commission from Fisheries

Research Agency, National Institute for Environmental Studies and so on.

Toru Suzuki demonstrated to entry two sample metadata to PICES–MIRC metadatabase and pointed out some problems in searching results. Toru Suzuki also reported that Isite compiled from source package on Solaris machine successfully but searching results was not correct, nevertheless Linux binary package show the correct results using the same sample metadata. Participants discussed about them and Kimberly Bahl and Toru Suzuki decided to keep contact and ask to Isite support by email after this meeting. Toru Suzuki also tried to register with Geonetwork.

Participants drew up future plans. Participants deeply discussed about NSDI CAP 2007 proposal relation with NPEM–China federation and decided to contact China delegation of TCODE. Norio Baba said that the different agency in China delegate to DINRAC and therefore it may require any help to establishment of metadatabase in NOWPAP. Norio Baba also said that NOWPAP has worked on capacity building of metadata and might be able to invite a specialist from NPEM. Toru Suzuki said that representatives of the Republic of Korea and Japan may also assist the DINRAC activity.

Priority	Task	NPEM	MIRC	Due Date
2	Contact FGDC about Geonetwork Beta-version	X		Oct. 30, 2006
1	Contact FGDC about Isite issues	X	X	Oct. 30, 2006
1	Revise PICES technical report No. 1 (NOWPAP to review)	X	X	Nov. 22, 2006
1	NSDI CAP 2007 proposal Category I	X		Jan. 19, 2007
1	Contact China delegation of TCODE	X		asap
1	Develop proposal to Nippon Foundation			Oct. 23, 2006
1	Prepare article of PICES press	X		Nov. 23, 2006
1	Minutes of this meeting	X	X	Oct. 30, 2006
2	Prepare conversion to FGDC standard from CSR stored in JODC		X	Mar. 30, 2007

16.3. XML stylesheet conversion crosswalks

16.3.1. FGDCI to DIF stylesheet converter

[A copy of this stylesheet (fgdcxml-difxml_NOAA_NODC.xml) can be found at the PICES web site.]

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE DIF SYSTEM
"http://gcmd.gsfc.nasa.gov/Aboutus/xml/dif/dif_v9.4.dtd">

<!-- since an XSL stylesheet is an XML document itself, it always begins with
the XML declaration -->
<!-- <?xml version="1.0" encoding="ISO-8859-1"?> -->
<!-- DTD declared with a Document Type Declaration.
To create an external reference to a DTD, use the SYSTEM command -->

<!-- start a XSL stylesheet -->

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:java="http://xml.apache.org/xalan/java" exclude-result-
  prefixes="java"
  version="1.0">

<!-- xmlns:java="http://xml.apache.org/xslt/java" exclude-result-
prefixes="java" -->
<!-- declare that this document is an XSLT style sheet and points to the
official W3C XSLT namespace; points to Java to support different browsers and
different user needs (Establishes the java namespace for use with extensions.
Xalan is written in Java). JavaScript that uses an XML parser to do the
transformation, making XML data available to all skind of browsers -->

<xsl:output method="xml" />

<!--The root element that declares the document to be an XSL style sheet is
<xsl:stylesheet> or <xsl:transform>, either can be used. This is following
W3C XSLT recommendation. To get access to the XSLT elements, attributes,
and features, XSLT namespace must be declared at the top of the document
(xmlns:xsl=" ").

DIF xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://gcmd.gsfc.nasa.gov/Aboutus/xml/dif/dif.
xsd">

Additional information:
http://www.w3schools.com/xsl/xsl_transformation.asp
http://www.fgdc.gov/metadata/csdgm/
http://biology.usgs.gov/fgdc.metadata/version2/

Command line:

java org.apache.xalan.xslt.Process -IN FGDC_*.xml -XSL fgdcxml-
difxml_NOAA_NODC.xml -OUT DIF_*.xml
/home/mmeaux/bin/tidy -config /home/mmeaux/bin/config.txt DIF_*.xml >
DIF_*_clean.xml
-->
```



```

<!-- This is an XSLT style sheet that converts metadata in FGDC XML format to
IDN DIF XML format.-->
<!-- Warning: This XSLT is a work in progress -->

<!-- Author: Scott Ritz, ritz@gcmd.nasa.gov -->
<!-- Modified by Melanie Meaux in 2005s -->
<!-- Contact: mmeaux@gcmd.nasa.gov -->

<!-- apply templates; the <xsl:template> element is used to build templates,
one or more set of rules. The match attribute is used to associate a template
with an XML element. The match attribute can also be used to define a
template for the entire XML document. The value of the match attribute is an
XPath expression (i.e. match="/" defines the whole document). The content
inside the element defines some HTML to write to the output.

The <xsl:apply-templates> element applies a template to the current element
or to the current element's child nodes. If we add a select attribute to the
<xsl:apply-templates> element it will process only the child element that
matches the value of the attribute. We can use the "select" attribute to
specify the order in which the child nodes are processed.
-->

<!--Main template for the DIF document; -->
<!-- for repeated fields (i.e personnel) need to include DIF field attributes
in xsl:template match = ""> and not in template below -->

<xsl:template match="metadata">
  <DIF>
  <!--
  <Entry_ID>Place_Holder</Entry_ID>-->
  <!-- Inserts the Resource description short name into the Entry_ID -->
  <Entry_ID><xsl:apply-templates select="/metadata/distinfo/resdesc"
  /></Entry_ID>
  <Entry_Title><xsl:apply-templates
  select="/metadata/idinfo/citation/citeinfo/title" /></Entry_Title>

  <xsl:apply-templates select="idinfo" /> <!-- Identification Information --
  >
  <xsl:apply-templates select="metainfo" /> <!-- Metadata Reference
  Information -->
  <xsl:apply-templates select="distinfo" /> <!-- Distribution Informaiton --
  >
  <xsl:apply-templates select="dataqual" /> <!-- Data Quality Information --
  >
  <IDN_Node>
  <Short_Name>USA/NOAA</Short_Name>
  </IDN_Node>
  <Data_Set_Language>ENGLISH</Data_Set_Language>
  <Originating_Metadata_Node>GCMD</Originating_Metadata_Node>
  <Metadata_Name>CEOS IDN DIF</Metadata_Name> <!-- Metadata Name -->
  <Metadata_Version>VERSION 9.0</Metadata_Version> <!-- Metadata Version -->
  </DIF>
</xsl:template>

<!-- calling above templates -->
<!--
-->
1. IDENTIFICATION INFORMATION
-->

```

```

<xsl:template match="idinfo">
  <xsl:apply-templates select="citation" /> <!-- citation -->
  <Summary>
  <xsl:apply-templates select="descript" /> <!-- Description -->
  </Summary>
  <xsl:apply-templates select="timeperd" /> <!--Time Period of Content -->
  <!-- place-holder for Paleo Temporal field -->
  <xsl:apply-templates select="status" /> <!-- Status-->
  <xsl:apply-templates select="spdom" /> <!-- Spatial Domain -->
  <xsl:apply-templates select="keywords" />
  <xsl:apply-templates select="acconst" /> <!-- access constraints -->
  <Use_Constraints> <!-- use constraints -->
  <xsl:apply-templates select="useconst" />
  <xsl:apply-templates select="/metadata/distinfo/distliab" />
  </Use_Constraints>
  <xsl:apply-templates select="ptcontac" /> <!-- point of contact -->
  <xsl:apply-templates select="browse" />
  <Reference>
    <xsl:apply-templates select="/metadata/idinfo/crossref/citeinfo/origin"
    />
    <xsl:apply-templates select="/metadata/idinfo/crossref/citeinfo/title" />
    <xsl:apply-templates
    select="/metadata/idinfo/crossref/citeinfo/serinfo/sername" />
    <xsl:apply-templates select="/metadata/idinfo/crossref/citeinfo/pubdate"
    />
    <xsl:apply-templates
    select="/metadata/idinfo/crossref/citeinfo/pubinfo/pubplace" />
    <xsl:apply-templates
    select="/metadata/idinfo/crossref/citeinfo/pubinfo/publish" />
    <xsl:apply-templates select="/metadata/idinfo/crossref/citeinfo/edition"
    />
    <xsl:apply-templates
    select="/metadata/idinfo/crossref/citeinfo/serinfo/issue" />
    <xsl:apply-templates select="/metadata/idinfo/crossref/citeinfo/geoform"
    />
    <xsl:apply-templates select="/metadata/idinfo/crossref/citeinfo/othercit"
    />
    <xsl:apply-templates select="/metadata/idinfo/crossref/citeinfo/onlink"
    />
  </Reference>
</xsl:template>

<!-- begin citation -->
<xsl:template match="citation">
  <Data_Set_Citation><xsl:apply-templates select="citeinfo"
  /></Data_Set_Citation>
</xsl:template>

<xsl:template match="citeinfo"> <!-- Citation Information -->
  <Dataset_Creator><xsl:apply-templates select="origin" /></Dataset_Creator>
  <!-- Originator: the name of an organization or individual that developed
  the data set. If the name of editors or compilers are provided, the name
  must be followed by "(ed.)" or "(comp.)" respectively. This list can be
  quite long and exceed the character length specified by the DIF in
  Dataset_Creator. In some cases, the Pulisher can replace the Originator in

```

```

DIF Dataset_Creator. Publisher: the name of the individual or organization
that published the data set.-->
<Dataset_Title><xsl:apply-templates select="title" /></Dataset_Title>
<xsl:apply-templates select="serinfo" />
<Dataset_Release_Date><xsl:apply-templates select="pubdate" />
</Dataset_Release_Date>

<xsl:apply-templates select="pubinfo" />

<Version><xsl:apply-templates select="edition" /></Version>

<Data_Presentation_Form><xsl:apply-templates select="geoform"
/></Data_Presentation_Form>

<Other_Citation_Details><xsl:apply-templates select="othercit"
/></Other_Citation_Details>
  <Online_Resource><xsl:apply-templates select="onlink" /></Online_Resource>
</xsl:template>

<xsl:template match="serinfo"> <!-- Series Information -->
  <Dataset_Series_Name><xsl:apply-templates select="sername"
  /></Dataset_Series_Name>
  <Issue_Identification><xsl:apply-templates select="issue"
  /></Issue_Identification>
</xsl:template>

<xsl:template match="pubinfo"> <!-- Publication Information -->
  <Dataset_Release_Place><xsl:apply-templates
  select="pubplace"/></Dataset_Release_Place>
  <Dataset_Publisher><xsl:apply-templates select="publish"
  /></Dataset_Publisher>
</xsl:template>

<!-- DIF Data Set Citation -->
<xsl:template match="origin"><xsl:value-of select="normalize-space(.)"
/></xsl:template>
<xsl:template match="title"><xsl:value-of select="normalize-space(.)"
/></xsl:template>
<xsl:template match="sername"><xsl:value-of select="." /></xsl:template>
<xsl:template match="pubdate">
<xsl:variable name="datetime" select="normalize-space(.)"/>
<xsl:variable name="year" select="substring($datetime, 1, 4)"/>
<xsl:variable name="month" select="substring($datetime, 5, 2)"/>
<xsl:variable name="day" select="substring($datetime, 7, 2)"/>
<xsl:value-of select="concat($year, '-', $month, '-', $day)" />
</xsl:template>
<xsl:template match="pubplace"><xsl:value-of select="." /></xsl:template>
<xsl:template match="publish"><xsl:value-of select="." /></xsl:template>
<xsl:template match="edition"><xsl:value-of select="." /></xsl:template>
<xsl:template match="issue"><xsl:value-of select="." /></xsl:template>
<xsl:template match="geoform"><xsl:value-of select="." /></xsl:template>
<xsl:template match="othercit"><xsl:value-of select="." /></xsl:template>
<xsl:template match="onlink"><xsl:value-of select="." /></xsl:template>
<!-- end citation -->

<!-- begin summary -->

```

```

        <!-- DIF Summary -->
<xsl:template match="descript">
  <xsl:apply-templates select="abstract" />
  <xsl:apply-templates select="purpose" />
  <xsl:apply-templates select="supplinf" />
  <xsl:apply-templates select="/metadata/idinfo/native" />
<!-- native: a description of the data set in the producer's processing
environment, including items such as the name of the software (including
version), the computer operating system, file name (including host-, path-,
and filenames), and the data set size.-->
  <!-- <xsl:apply-templates select="/metadata/spref" /> spatial information
not in DIF; different from Spatial Domain in Identification Information
section-->
</xsl:template>
<xsl:template match="abstract"><xsl:text> ABSTRACT: </xsl:text><xsl:value-of
select="normalize-space(.)" /></xsl:template>
<xsl:template match="purpose"><xsl:text> PURPOSE: </xsl:text><xsl:value-of
select="normalize-space(.)" /></xsl:template>
<xsl:template match="supplinf"><xsl:text> SUPPLEMENTAL INFORMATION:
</xsl:text><xsl:value-of select="normalize-space(.)" /></xsl:template>
<xsl:template match="native"><xsl:text> NATIVE: </xsl:text><xsl:value-of
select="normalize-space(.)" /></xsl:template>
<!-- end summary -->

<!-- begin Temporal Coverage -->
  <!-- DIF Temporal_Coverage -->
<xsl:template match="timeperd"><xsl:apply-templates select="timeinfo"
/></xsl:template>
  <xsl:template match="timeinfo">
<Temporal_Coverage><xsl:apply-templates select="rngdates"
/></Temporal_Coverage>
</xsl:template>

<xsl:template match="rngdates">
  <Start_Date><xsl:apply-templates select="begdate" /></Start_Date>
  <Stop_Date><xsl:apply-templates select="enddate" /></Stop_Date>
</xsl:template>

<xsl:template match="begdate">
<xsl:variable name="datetime" select="normalize-space(.)"/>
<xsl:variable name="year" select="substring($datetime, 1, 4)"/>
<xsl:variable name="month" select="substring($datetime, 5, 2)"/>
<xsl:variable name="day" select="substring($datetime, 7, 2)"/>
<xsl:value-of select="concat($year, '-', $month, '-', $day)" />
</xsl:template>

<xsl:template match="enddate">
<xsl:variable name="datetime" select="normalize-space(.)"/>
<xsl:variable name="year" select="substring($datetime, 1, 4)"/>
<xsl:variable name="month" select="substring($datetime, 5, 2)"/>
<xsl:variable name="day" select="substring($datetime, 7, 2)"/>
<xsl:value-of select="concat($year, '-', $month, '-', $day)" />
</xsl:template>
<!-- end temporal coverage -->

<!-- begin status -->

```

```

        <!-- DIF Data_Set_Progress -->
<xsl:template match="status">
  <Data_Set_Progress><xsl:apply-templates select="progress"
  /></Data_Set_Progress>
  <!-- <xsl:apply-templates select="update" /> -->
</xsl:template>

<xsl:template match="progress">
<xsl:value-of
select="translate(.,'abcdefghijklmnopqrstuvwxyz','ABCDEFGHIJKLMNOPQRSTUVWXYZ'
)" />
</xsl:template>

<!-- <xsl:template match="update"><xsl:value-of select="." /></xsl:template>
-->
<!-- end status -->

<!-- begin spatial coverage -->
  <!-- DIF Spatial_Coverage -->
<xsl:template match="spdom">
  <Spatial_Coverage><xsl:apply-templates select="bounding"
  /></Spatial_Coverage>
</xsl:template>

<xsl:template match="bounding">
  <Southernmost_Latitude><xsl:apply-templates
select="southbc"/></Southernmost_Latitude>
  <Northernmost_Latitude><xsl:apply-templates select="northbc"
  /></Northernmost_Latitude>
  <Westernmost_Longitude><xsl:apply-templates select="westbc"
  /></Westernmost_Longitude>
  <Easternmost_Longitude><xsl:apply-templates select="eastbc"
  /></Easternmost_Longitude>
</xsl:template>

<xsl:template match="westbc"><xsl:value-of select="." /></xsl:template>
<xsl:template match="eastbc"><xsl:value-of select="." /></xsl:template>
<xsl:template match="northbc"><xsl:value-of select="." /></xsl:template>
<xsl:template match="southbc"><xsl:value-of select="." /></xsl:template>
<!-- end Spatial Coverage -->

<!-- begin keywords/additional mapping needed to Parameter and keyword DIF
fields -->
<xsl:template match="keywords">
  <xsl:apply-templates select="theme" />
  <xsl:apply-templates select="place" />
</xsl:template>
<xsl:template match="theme">
  <xsl:apply-templates select="themekt" /> <!-- Theme_Keyword_Thesaurus -->
  <xsl:apply-templates select="themekey" /> <!-- Theme Keyword -->
</xsl:template>

<xsl:template match="place">
  <xsl:apply-templates select="placekt" />
  <xsl:apply-templates select="placekey" />
</xsl:template>

```

```

<!-- DIF Keyword and Science Parameter Field -->
<xsl:template match="themekt">
  <xsl:choose>
    <xsl:when test="'Global Change Master Directory'"> <!-- specific to NOAA
    NODC -->
      <Keyword/>
    </xsl:when>
    <xsl:otherwise>
      <Keyword><xsl:value-of select="." /></Keyword>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template match="themekey">
  <xsl:choose>
    <xsl:when test="'Global Change Master Directory'">
      <Parameters>
        <Category>
          <xsl:value-of select="." />
        </Category>
      </Parameters>
    </xsl:when>
    <xsl:otherwise>
      <Keyword><xsl:value-of select="." /></Keyword>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template match="placekt">
  <xsl:choose>
    <xsl:when test="'Global Change Master Directory'"> <!-- specific to NOAA
    NODC -->
      <Keyword/>
    </xsl:when>
    <xsl:otherwise>
      <Keyword><xsl:value-of select="." /></Keyword>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template match="placekey">
  <xsl:choose>
    <xsl:when test="'Global Change Master Directory'">
      <Location>
        <Location_Name>
          <xsl:value-of select="." />
        </Location_Name>
      </Location>
    </xsl:when>
    <xsl:otherwise>
      <Keyword><xsl:value-of select="." /></Keyword>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
<!-- end keywords -->

<!-- begin access constraints -->

```

```

        <!-- DIF Access_Constraints -->
<xsl:template match="accconst"><Access_Constraints><xsl:value-of select="."
/></Access_Constraints></xsl:template>
<!-- end access constraints -->

<!-- begin use constraints -->
        <!-- DIF Use_Constraints -->
<xsl:template match="useconst"><xsl:value-of select="." /></xsl:template>
<xsl:template match="distliab"><xsl:value-of select="." /></xsl:template>
<!-- Distribution Liability-statement of the liability assumed by the
distributor -->
<!-- end use constraints -->

<!-- begin point of contact -->
        <!-- DIF Technical Contact -->
<xsl:template match="ptcontac">
  <Personnel>
  <Role>TECHNICAL CONTACT</Role>
  <xsl:apply-templates select="/metadata/idinfo/ptcontac/cntinfo" />
  </Personnel>
</xsl:template>

<xsl:template match="/metadata/idinfo/ptcontac/cntinfo">
  <xsl:apply-templates select="cntperp" /> <!-- contact person primary -->
  <xsl:apply-templates select="cntorgp" /> <!-- contact Organization primary
-->
  <Contact_Address><xsl:apply-templates select="cntaddr"
/></Contact_Address>
  <xsl:apply-templates select="cntemail" />
  <xsl:apply-templates select="cntvoice" />
  <xsl:apply-templates select="cntfax" />
</xsl:template>
<xsl:template match="/metadata/idinfo/ptcontac/cntinfo/cntperp">
<xsl:apply-templates
select="/metadata/idinfo/ptcontac/cntinfo/cntperp/cntper" /> <!-- contact
person -->
<xsl:apply-templates
select="/metadata/idinfo/ptcontac/cntinfo/cntperp/cntorg" /> <!-- contact
organization -->
</xsl:template>

<xsl:template
match="/metadata/idinfo/ptcontac/cntinfo/cntperp/cntper"><First_Name><xsl:valu
e-of select="." /></First_Name></xsl:template>
<xsl:template
match="/metadata/idinfo/ptcontac/cntinfo/cntperp/cntorg"><Last_Name><xsl:valu
e-of select="." /></Last_Name></xsl:template>

<xsl:template match="cntaddr">
  <xsl:apply-templates select="address" />
  <xsl:apply-templates select="city" />
  <xsl:apply-templates select="state" />
  <xsl:apply-templates select="postal" />
  <xsl:apply-templates select="country" />
</xsl:template>

```

```

<xsl:template match="address"><Address><xsl:value-of select="."
/></Address></xsl:template>
<xsl:template match="city"><City><xsl:value-of select="."
/></City></xsl:template>
<xsl:template match="state"><Province_or_State><xsl:value-of select="."
/></Province_or_State></xsl:template>
<xsl:template match="postal"><Postal_Code><xsl:value-of select="."
/></Postal_Code></xsl:template>
<xsl:template match="country"><Country><xsl:value-of select="."
/></Country></xsl:template>

<xsl:template match="cntemail"><Email><xsl:value-of select="."
/></Email></xsl:template>
<xsl:template match="cntvoice"><Phone><xsl:value-of select="."
/></Phone></xsl:template>
<xsl:template match="cntfax"><Fax><xsl:value-of select="."
/></Fax></xsl:template>
<!-- end point of contact -->

<!-- begin browse -->
    <!-- DIF Multimedia_Sample -->
<xsl:template match="browse">
    <Multimedia_Sample>
        <File>
            <xsl:apply-templates select="browsen" /> <!-- Browse Graphic File Name --
            >
        </File>
        <Format>
            <xsl:apply-templates select="browset" /> <!-- Browse Graphic File Type -->
        </Format>
        <Description>
            <xsl:apply-templates select="browsed" /> <!-- Browse Graphic File
            Description -->
        </Description>
    </Multimedia_Sample>
</xsl:template>

<xsl:template match="browsen"><xsl:value-of select="." /></xsl:template>
<xsl:template match="browset"><xsl:value-of select="." /></xsl:template>
<xsl:template match="browsed"><xsl:value-of select="." /></xsl:template>
<!-- end browse -->

<!-- begin cross reference -->
<!-- Cross Reference: information about other, related data sets that are
likely to be of interest.-->
<xsl:template match="/metadata/idinfo/crossref/citeinfo/origin"><xsl:value-of
select="normalize-space(.)" /></xsl:template>
<xsl:template match="/metadata/idinfo/crossref/citeinfo/title"><xsl:value-of
select="normalize-space(.)" /></xsl:template>
<xsl:template
match="/metadata/idinfo/crossref/citeinfo/serinfo/sername"><xsl:value-of
select="." /></xsl:template>
<xsl:template match="/metadata/idinfo/crossref/citeinfo/pubdate">
<xsl:variable name="datetime" select="normalize-space(.)"/>
<xsl:variable name="year" select="substring($datetime, 1, 4)"/>
<xsl:variable name="month" select="substring($datetime, 5, 2)"/>
<xsl:variable name="day" select="substring($datetime, 7, 2)"/>

```



```

<xsl:value-of select="concat($year, '-', $month, '-', $day)" />
</xsl:template>
<xsl:template
match="/metadata/idinfo/crossref/citeinfo/pubinfo/pubplace"><xsl:value-of
select="." /></xsl:template>
<xsl:template
match="/metadata/idinfo/crossref/citeinfo/pubinfo/publish"><xsl:value-of
select="." /></xsl:template>
<xsl:template match="/metadata/idinfo/crossref/citeinfo/edition"><xsl:value-
of select="." /></xsl:template>
<xsl:template match="/metadata/idinfo/crossref/citeinfo/issue"><xsl:value-of
select="." /></xsl:template>
<xsl:template match="/metadata/idinfo/crossref/citeinfo/geoform"><xsl:value-
of select="." /></xsl:template>
<xsl:template match="/metadata/idinfo/crossref/citeinfo/othercit"><xsl:value-
of select="." /></xsl:template>
<xsl:template match="/metadata/idinfo/crossref/citeinfo/onlink"><xsl:value-of
select="." /></xsl:template>
<!-- end cross reference -->

<!--          2. METADATA REFERENCE INFORMATION          -
->
<!-- DIF Author -->
<xsl:template match="metainfo">
  <Personnel>
    <Role>DIF AUTHOR</Role>
    <xsl:apply-templates select="/metadata/metainfo/metc/cntinfo" /> <!--
Metadata Contact -->
  </Personnel>
  <!-- <xsl:apply-templates select="metstdn" /> --> <!-- Metadata Name -->
  <!-- <xsl:apply-templates select="metstdv" /> --> <!-- Metadata Version --
  >
  <xsl:apply-templates select="metd" /> <!-- Metadata date -->
  <xsl:apply-templates select="metrd" /> <!-- Metadata Review date -->
  <xsl:apply-templates select="metfrd" /> <!-- Metadata Future Review date -
  ->
</xsl:template>

<xsl:template match="/metadata/metainfo/metc/cntinfo">
  <xsl:apply-templates select="cntperp" /> <!-- contact person primary -->
  <xsl:apply-templates select="cntorgp" /> <!-- contact Organization primary
  -->
  <Contact_Address><xsl:apply-templates select="cntaddr"
  /></Contact_Address>
  <xsl:apply-templates select="cntemail" />
  <xsl:apply-templates select="cntvoice" />
  <xsl:apply-templates select="cntfax" />
</xsl:template>

<xsl:template match="/metadata/metainfo/metc/cntinfo/cntperp">
<xsl:apply-templates select="/metadata/metainfo/metc/cntinfo/cntperp/cntper"
/> <!-- contact person -->
<xsl:apply-templates select="/metadata/metainfo/metc/cntinfo/cntperp/cntorg"
/> <!-- contact organization -->
</xsl:template>

```

```

<xsl:template
match="/metadata/metainfo/metc/cntinfo/cntperp/cntper"><First_Name><xsl:value
-of select="." /></First_Name></xsl:template>
<xsl:template
match="/metadata/metainfo/metc/cntinfo/cntperp/cntorg"><Last_Name><xsl:value-
of select="." /></Last_Name></xsl:template>

<xsl:template match="cntaddr">
  <xsl:apply-templates select="address" />
  <xsl:apply-templates select="city" />
  <xsl:apply-templates select="state" />
  <xsl:apply-templates select="postal" />
  <xsl:apply-templates select="country" />
</xsl:template>

<xsl:template match="address"><Address><xsl:value-of select="."
/></Address></xsl:template>
<xsl:template match="city"><City><xsl:value-of select="."
/></City></xsl:template>
<xsl:template match="state"><Province_or_State><xsl:value-of select="."
/></Province_or_State></xsl:template>
<xsl:template match="postal"><Postal_Code><xsl:value-of select="."
/></Postal_Code></xsl:template>
<xsl:template match="country"><Country><xsl:value-of select="."
/></Country></xsl:template>

<xsl:template match="cntemail"><Email><xsl:value-of select="."
/></Email></xsl:template>
<xsl:template match="cntvoice"><Phone><xsl:value-of select="."
/></Phone></xsl:template>
<xsl:template match="cntfax"><Fax><xsl:value-of select="."
/></Fax></xsl:template>

<!-- DIF Metadata Creation Date -->
<xsl:template match="metd">
  <DIF_Creation_Date>
    <xsl:variable name="datetime" select="normalize-space(.)"/>
    <xsl:variable name="year" select="substring($datetime, 1, 4)"/>
    <xsl:variable name="month" select="substring($datetime, 5, 2)"/>
    <xsl:variable name="day" select="substring($datetime, 7, 2)"/>
    <xsl:value-of select="concat($year, '-', $month, '-', $day)" />
  </DIF_Creation_Date>
</xsl:template>

<!-- DIF Last DIF Revision Date -->
<xsl:template match="metrd">
  <Last_DIF_Revision_Date>
    <xsl:variable name="datetime" select="normalize-space(.)"/>
    <xsl:variable name="year" select="substring($datetime, 1, 4)"/>
    <xsl:variable name="month" select="substring($datetime, 5, 2)"/>
    <xsl:variable name="day" select="substring($datetime, 7, 2)"/>
    <xsl:value-of select="concat($year, '-', $month, '-', $day)" />
  </Last_DIF_Revision_Date>
</xsl:template>

<!-- DIF Future DIF Revision Date -->
<xsl:template match="metfrd">

```

```

<Future_DIF_Review_Date>
  <xsl:variable name="datetime" select="normalize-space(.)"/>
  <xsl:variable name="year" select="substring($datetime, 1, 4)"/>
  <xsl:variable name="month" select="substring($datetime, 5, 2)"/>
  <xsl:variable name="day" select="substring($datetime, 7, 2)"/>
  <xsl:value-of select="concat($year, '-', $month, '-', $day)" />
</Future_DIF_Review_Date>
</xsl:template>

<!--          3. DISTRIBUTION INFORMATION
-->
<xsl:template match="distinfo">
  <xsl:apply-templates select="distrib" /> <!-- Distributor -->
  <xsl:apply-templates select="stdorder" /> <!-- Standard Order Process -->
  <!-- <xsl:apply-templates select="distliab" /> DIF Use_Constraints not
Distribution field -->
</xsl:template>

<!-- DIF Data Center & Data Center Personnel Information -->
<xsl:template match="distrib">
  <Data_Center>
  <Data_Center_Name>
  <Short_Name>DOC/NOAA/NESDIS/NODC</Short_Name>
  <Long_Name>National Oceanographic Data Center, NESDIS, NOAA, U.S.
Department of Commerce</Long_Name>
  </Data_Center_Name>
  <Data_Center_URL>
  <xsl:apply-templates select="/metadata/idinfo/citation/citeinfo/onlink" />
  </Data_Center_URL>
  <Data_Set_ID><xsl:apply-templates select="/metadata/distinfo/resdesc"
/></Data_Set_ID>
  <Personnel>
  <Role>DATA CENTER CONTACT</Role>
  <xsl:apply-templates select="/metadata/distinfo/distrib/cntinfo" />
  </Personnel>
  </Data_Center>
</xsl:template>

<xsl:template match="/metadata/distinfo/distrib/cntinfo">
  <!-- <xsl:apply-templates select="cntperp" /> --> <!-- contact person
primary -->
  <xsl:apply-templates select="cntorgp" /> <!-- contact Organization primary
-->
  <Contact_Address><xsl:apply-templates select="cntaddr"
/></Contact_Address>
  <xsl:apply-templates select="cntemail" />
  <xsl:apply-templates select="cntvoice" />
  <xsl:apply-templates select="cntfax" />
</xsl:template>

<xsl:template match="/metadata/distinfo/distrib/cntinfo/cntorgp">
  <!-- <xsl:apply-templates
select="/metadata/distinfo/distrib/cntinfo/cntperp/cntper" /> --> <!--
contact person -->
  <xsl:apply-templates
select="/metadata/distinfo/distrib/cntinfo/cntorgp/cntorg" /> <!-- contact
organization -->

```

```

</xsl:template>

<!-- <xsl:template
match="/metadata/distinfo/distrib/cntinfo/cntperp/cntper"><First_Name><xsl:va
lue-of select="." /></First_Name></xsl:template -->
<xsl:template
match="/metadata/distinfo/distrib/cntinfo/cntorgp/cntorg"><Last_Name><xsl:val
ue-of select="." /></Last_Name></xsl:template>

<xsl:template match="cntaddr">
  <xsl:apply-templates select="address" />
  <xsl:apply-templates select="city" />
  <xsl:apply-templates select="state" />
  <xsl:apply-templates select="postal" />
  <xsl:apply-templates select="country" />
</xsl:template>

<xsl:template match="address"><Address><xsl:value-of select="."
/></Address></xsl:template>
<xsl:template match="city"><City><xsl:value-of select="."
/></City></xsl:template>
<xsl:template match="state"><Province_or_State><xsl:value-of select="."
/></Province_or_State></xsl:template>
<xsl:template match="postal"><Postal_Code><xsl:value-of select="."
/></Postal_Code></xsl:template>
<xsl:template match="country"><Country><xsl:value-of select="."
/></Country></xsl:template>

<xsl:template match="cntemail"><Email><xsl:value-of select="."
/></Email></xsl:template>
<xsl:template match="cntvoice"><Phone><xsl:value-of select="."
/></Phone></xsl:template>
<xsl:template match="cntfax"><Fax><xsl:value-of select="."
/></Fax></xsl:template>

<xsl:template match="resdesc">
<!-- Resource description - identifier by the which
the distributor knows the data set. // DIF Entry_ID -->
  <!-- <xsl:value-of select="." /> -->
  <xsl:value-of select="translate(.,' ','_')" />
</xsl:template>

<!-- DIF Distribution Field -->
<xsl:template match="stdorder">
  <xsl:apply-templates select="digform" />
  <xsl:apply-templates select="digform/digtopt" />
  <!--<xsl:apply-templates select="fees" /> -->
</xsl:template>

<xsl:template match="digform">
<xsl:apply-templates select="digtinfo" />
<!-- <xsl:apply-templates select="digtopt" /> -->
</xsl:template>

<xsl:template match="digtinfo">
  <Distribution>
  <Distribution_Media>ONLINE</Distribution_Media>

```

```

<Distribution_Format>
  <xsl:apply-templates select="formname" />
  <xsl:apply-templates select="formvern" />
</Distribution_Format>
<Distribution_Size>
  <xsl:apply-templates select="transize" />
</Distribution_Size>
<Fees>
  <xsl:apply-templates select="/metadata/distinfo/stdorder/fees"/>
</Fees>
</Distribution>
</xsl:template>

<xsl:template match="formname"><xsl:value-of select="." /></xsl:template>
<xsl:template match="formvern"><xsl:value-of select="." /></xsl:template>
<xsl:template match="transize"><xsl:value-of select="." /></xsl:template>
<xsl:template match="fees"><xsl:value-of select="." /></xsl:template>

<!-- RELATED URLs -->
<xsl:template match="/digform/digtopt">
  <xsl:apply-templates select="/onlinopt" />
</xsl:template>

<xsl:template match="onlinopt">
<Related_URL>
  <URL_Content_Type>GET DATA</URL_Content_Type>
  <xsl:apply-templates select="computer" />
  <Description>
    <xsl:apply-templates select="accinstr" />
  </Description>
</Related_URL>
</xsl:template>

<xsl:template match="computer">
  <xsl:apply-templates select="networka" />
</xsl:template>

<xsl:template match="networka">
  <URL><xsl:apply-templates select="networkr" /></URL>
</xsl:template>

<xsl:template match="networkr"><xsl:value-of select="." /></xsl:template>
<xsl:template match="accinstr"><xsl:value-of select="." /></xsl:template>

<!--
<Distribution>
<Distribution_Media>
<xsl:apply-templates select="offmedia" />
</Distribution_Media>
</Distribution>
-->

<!-- DIF Originating Center -->
<xsl:template
match="/metadata/distinfo/distinfo/distrib/cntinfo/cntorgp/cntorg">
  <Originating_Center><xsl:value-of select="." /></Originating_Center>

```

```

</xsl:template>

<!--          4. DATA QUALITY INFORMATION  -->
<!-- Begin DIF Quality Field -->
<xsl:template match="dataqual">
  <Quality>
    <xsl:apply-templates select="attracc" /> <!-- Attribute Accuracy Report --
    >
    <xsl:apply-templates select="logic" /> <!-- Logical Consistency Report -->
    <xsl:apply-templates select="complete" /> <!-- Completeness Report -->
    <xsl:apply-templates select="lineage/procstep/procdesc" /> <!-- Process
    Description -->
  </Quality>
</xsl:template>

<xsl:template match="attracc">
  <xsl:apply-templates select="attraccr" />
</xsl:template>

<xsl:template match="attraccr"><xsl:value-of select="." /></xsl:template>
<xsl:template match="logic"><xsl:value-of select="." /></xsl:template>
<xsl:template match="complete"><xsl:value-of select="." /></xsl:template>
<xsl:template match="procdesc"><xsl:value-of select="." /></xsl:template>
<!-- End DIF Quality Field -->

<!-- end of stylesheet -->
</xsl:stylesheet>

```

16.3.2. *DIF to FGDCI stylesheet converter*

[A copy of this stylesheet (dif_to_fgdc.xls) can be found at the PICES web site.]

```

<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>

<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:str="http://xslt.org/string"
xmlns:xalan="http://xml.apache.org/xalan">

<!-- Stylesheet created by Fedele Stella (GeoConnections Discovery Portal) --
>

<!-- Stylesheet modified by Scott Ritz (SSAI,Inc.) in April 2005
Validates against the FGDC DTD.
Contact: ritz@gcmd.nasa.gov -->

<!-- Stylesheet modified by Melanie F. Meaux (SSAI,Inc.) in July 2005
Validates against the FGDC DTD.
Contact: mmeaux@gcmd.nasa.gov -->

<!-- Additional information:
http://www.w3schools.com/xsl/xsl_transformation.asp
http://www.fgdc.gov/metadata/csdgm/
http://biology.usgs.gov/fgdc.metadata/version2/

```

Command line:

```
java org.apache.xalan.xslt.Process -IN DIF_*.xml -XSL dif_to_fgdc3.xsl -OUT  
FGDC_*.xml
```

```
/home/mmeaux/bin/tidy -config /home/mmeaux/bin/config.txt FGDC_*.xml >  
FGDC_*_clean.xml
```

```
mp.lnx FGDC_*_clean.xml -e FGDC_*_clean.err
```

```
for multiple files, run  
cd in directory where files are (US_GLOBEC_NEP_0001 ...etc)  
(source .bashrc)  
csh  
dif_fgdc_mload  
dif_fgdc_clean
```

```
DIF xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:noNamespaceSchemaLocation="http://gcmd.gsfc.nasa.gov/Aboutus/xml/dif/dif.  
xsd">  
-->
```

```
<xsl:import href="string-modified.xsl"/>
```

```
<xsl:output method="xml" encoding="ISO-8859-1" omit-xml-declaration="no"  
standalone="yes" indent="yes" xalan:indent-amount="3"/>
```

```
<xsl:preserve-space elements="*" />
```

```
<xsl:template match="/DIF" xml:space="preserve">  
<metadata>  
  <idinfo>
```

```
<!-- begin Citation -->  
  <xsl:choose>  
    <xsl:when test="Data_Set_Citation">  
      <xsl:apply-templates select="Data_Set_Citation"/>  
    </xsl:when>  
    <xsl:otherwise>  
      <xsl:apply-templates select="Originating_Metadata_Node" />  
    </xsl:otherwise>  
  </xsl:choose>  
<!-- end Citation -->
```

```
<!-- begin Description -->  
  <descript>  
    <abstract>  
      <xsl:apply-templates select="Summary"/>  
    </abstract>  
    <purpose>Not Available</purpose>  
    <xsl:choose>  
      <xsl:when test="Reference">  
        <xsl:apply-templates select="Reference"/>  
      </xsl:when>  
      <xsl:otherwise>  
        <supplinf>Not Available</supplinf>  
      </xsl:otherwise>
```

```

        </xsl:choose>
        </descript>
<!-- end Description -->

<!-- begin Time Period of Content -->
    <xsl:choose>
        <xsl:when test="Temporal_Coverage">
            <xsl:apply-templates select="Temporal_Coverage"/>
        </xsl:when>
        <xsl:otherwise>
            <begdate>Unknown</begdate>
            <enddate>Unknown</enddate>
        </xsl:otherwise>
    </xsl:choose>
<!-- end Time Period of Content -->

<!-- begin Status -->
    <xsl:choose>
        <xsl:when test="Data_Set_Progress">
            <xsl:apply-templates select="Data_Set_Progress"/>
        </xsl:when>
        <xsl:otherwise>
            <status>
                <progress>Complete</progress> <!-- status and Update are mandatory
fields in FGDC -->
                <update>As needed</update>
            </status>
        </xsl:otherwise>
    </xsl:choose>
<!-- end Status -->

<!-- begin Spatial Domain -->
    <xsl:choose>
        <xsl:when test="Spatial_Coverage">
            <xsl:apply-templates select="Spatial_Coverage"/>
        </xsl:when>
        <xsl:otherwise>
            <spdom>
                <bounding>
                    <westbc>Not Available</westbc>
                    <eastbc>Not Available</eastbc>
                    <northbc>Not Available</northbc>
                    <southbc>Not Available</southbc>
                </bounding>
            </spdom>
        </xsl:otherwise>
    </xsl:choose>
<!-- end Spatial Domain -->

<!-- begin Keywords -->
    <keywords>
        <theme>
            <themekt>GCMD ENTRY ID</themekt>
            <xsl:apply-templates select="Entry_ID"/>
            <themekt>GCMD SCIENCE PARAMETERS</themekt>
            <xsl:apply-templates select="Parameters"/>
            <themekt>GCMD PLATFORM</themekt>

```



```

    <xsl:apply-templates select="Source_Name"/>
    <themekt>GCMD INSTRUMENT</themekt>
      <xsl:apply-templates select="Sensor_Name"/>
    <themekt>PROJECT</themekt>
      <xsl:apply-templates select="Project"/>
    <themekt>AUXILIARY KEYWORDS</themekt>
      <xsl:apply-templates select="Keyword"/>
    <themekt>ISO TOPIC CATEGORY</themekt>
      <xsl:apply-templates select="ISO_Topic_Category"/>
    <themekt>DATA SET LANGUAGE</themekt>
      <xsl:apply-templates select="Data_Set_Language"/>
  </theme>
<xsl:choose>
<xsl:when test="Location">
  <place>
    <placekt>GCMD</placekt>
    <xsl:apply-templates select="Location"/>
  </place>
</xsl:when>
</xsl:choose>
<xsl:choose>
<xsl:when test="/DIF/Data_Resolution/Temporal_Resolution">
  <temporal>
    <tempkt>GCMD</tempkt>
    <xsl:apply-templates
select="/DIF/Data_Resolution/Temporal_Resolution"/>
  </temporal>
</xsl:when>
</xsl:choose>
</keywords>
<!-- end Keywords -->

<!-- begin Access & Use Constraints -->
<xsl:choose>
<xsl:when test="Access_Constraints">
  <xsl:apply-templates select="Access_Constraints"/>
</xsl:when>
<xsl:otherwise>
  <acconst>Not Available</acconst>
</xsl:otherwise>
</xsl:choose>

<xsl:choose>
<xsl:when test="Use_Constraints">
  <xsl:apply-templates select="Use_Constraints"/>
</xsl:when>
<xsl:otherwise>
  <useconst>Not Available</useconst>
</xsl:otherwise>
</xsl:choose>
<!-- end Access & Use Constraints -->

<!-- begin Point of Contact -->
<xsl:if test="Personnel/Role/text() = 'TECHNICAL CONTACT'">
<ptcontac>
  <cntinfo>
    <xsl:apply-templates select="Personnel[Role='TECHNICAL CONTACT']"/>

```

```

    </cntinfo>
  </ptcontac>
</xsl:if>
<!--
<xsl:if test="Personnel/Role/text() = 'INVESTIGATOR'">
  <ptcontac>
  <cntinfo>
    <xsl:apply-templates select="Personnel[Role='INVESTIGATOR']"/>
  </cntinfo>
  </ptcontac>
</xsl:if>
-->
<!-- end Point of Contact -->

<!-- begin Browse Graphic-->
  <xsl:apply-templates select="Multimedia_Sample"/>
<!-- end Browse Graphic -->

<!-- begin Cross Reference -->
<!--   <xsl:apply-templates select="Reference"/> -->
<!-- end Cross Reference -->

</idinfo>

<!-- begin Data Quality information -->
<dataqual>
  <xsl:choose>
    <xsl:when test="Quality">
      <xsl:apply-templates select="Quality"/>
    </xsl:when>
    <xsl:otherwise>
      <attracc>
        <attraccr>Not Available</attraccr>
      </attracc>
      <logic>Not Available</logic> <!-- Mandatory in data quality -->
      <complete>Not Available</complete> <!-- Mandatory in data quality -
->
      <lineage>
        <procstep>
          <procdesc>Not Available</procdesc>
          <procddate>Unknown</procddate>
        </procstep>
      </lineage> <!-- Mandatory in data quality -->
    </xsl:otherwise>
  </xsl:choose>
</dataqual>
<!-- end Data Quality information -->

<!-- begin Spatial Reference information -->
<spref>
  <xsl:choose>
    <xsl:when test="Data_Resolution">
      <xsl:apply-templates select="Data_Resolution"/>
    </xsl:when>
  </xsl:choose>
<!--

```

```

    <xsl:otherwise>
      <horizsys>
<geograph>
<latres>Not Available</latres>
<longres>Not Available</longres>
<geogunit>Decimal degrees</geogunit>
</geograph>
</horizsys>
<vertdef>
  <altsys>
    <altdatum>Not Available</altdatum>
    <altres>Not Available</altres>
    <altunits>Not Available</altunits>
    <altenc>Implicit coordinate</altenc>
  </altsys>
  <depthsys>
    <depthdn>Not Available</depthdn>
    <depthres>Not Available</depthres>
    <depthdu>Not Available</depthdu>
    <depthem>Implicit coordinate</depthem>
  </depthsys>
</vertdef>
  </xsl:otherwise>
  -->
</xsl:choose>
</spref>
<!-- end Spatial Reference information -->

<!-- begin Distribution Info -->
<distinfo>
  <xsl:apply-templates select="Data_Center"/> <!-- distributor -->

  <xsl:choose>
    <xsl:when test="Data_Center/Data_Set_ID">
      <xsl:apply-templates select="Data_Center/Data_Set_ID"/> <!-- resource
description -->
    </xsl:when>
    <xsl:otherwise>
      <resdesc>Not Available</resdesc>
    </xsl:otherwise>
  </xsl:choose>

  <distliab>Not Available</distliab>

  <stdorder>
  <digform>
  <xsl:choose>
    <xsl:when test="Distribution">
      <digtinfo>
        <xsl:apply-templates select="Distribution"/>
      </digtinfo>
    </xsl:when>
    <xsl:otherwise>
      <digtinfo>
        <formname>Not Available</formname> <!-- Mandatory field -->
      </digtinfo>
    </xsl:otherwise>

```

```

</xsl:choose>

<xsl:choose>
<xsl:when test="/DIF/Data_Center/Data_Center_URL">
  <digtopt>
    <xsl:apply-templates select="/DIF/Data_Center/Data_Center_URL"/>
  </digtopt>
</xsl:when>
<xsl:otherwise>
  <digtopt/>
</xsl:otherwise>
</xsl:choose>

<xsl:choose>
<xsl:when test="Related_URL">
  <xsl:apply-templates select="Related_URL"/>
</xsl:when>
<xsl:otherwise>
  <digtopt/>
</xsl:otherwise>
</xsl:choose>

</digform>

<xsl:choose>
<xsl:when test="/DIF/Distribution/Fees">
  <xsl:apply-templates select="/DIF/Distribution/Fees"/>
</xsl:when>
<xsl:otherwise>
  <fees>Not Available</fees>
</xsl:otherwise>
</xsl:choose>

</stdorder>
</distinfo>
<!-- end Distribution Info -->

<!-- begin Metadata Info -->
<metainfo>
  <xsl:choose>
  <xsl:when test="DIF_Creation_Date">
    <xsl:apply-templates select="DIF_Creation_Date"/>
  </xsl:when>
  <xsl:otherwise>
    <metd/>
  </xsl:otherwise>
</xsl:choose>

  <xsl:apply-templates select="Last_DIF_Revision_Date"/>
  <xsl:apply-templates select="Future_DIF_Review_Date"/>

<metc>
  <xsl:choose>
    <xsl:when test="Personnel[Role='DIF AUTHOR']">
      <cntinfo>
        <xsl:apply-templates select="Personnel[Role='DIF AUTHOR']"/>
      </cntinfo>
    </xsl:when>
  </xsl:choose>

```

```

        </xsl:when>
        <xsl:otherwise>
        <cntinfo>
        <cntperp>
        <cntper>GCMD User Support Office</cntper>
        <cntorg>NASA Global Change Master Directory</cntorg>
        </cntperp>
        <cntaddr>
        <addrtype>Mailing and Physical Address</addrtype>
        <address>Not Available</address>
        <city>Lanham</city>
        <state>MD</state>
        <postal>20706</postal>
        <country>USA</country>
        </cntaddr>
        <cntvoice>Not Available</cntvoice>
        <cntemail>gcmduso@gcmd.gsfc.nasa.gov</cntemail>
        </cntinfo>
        </xsl:otherwise>
    </xsl:choose>
</metc>
    <metstdn>FGDC Content Standards for Digital Geospatial
Metadata</metstdn>
    <metstdv>FGDC-STD-001-1998</metstdv>
    <mettc>local time</mettc>
</metainfo>
<!-- end Metadata Info -->

</metadata>
</xsl:template>

<!-- <xsl:apply-templates select="Data_Set_Citation"/> -->
    <xsl:template match="Data_Set_Citation">
        <citation>
            <citeinfo>
                <xsl:choose>
                    <xsl:when test="Dataset_Creator">
                        <xsl:apply-templates select="Dataset_Creator"/>
                    </xsl:when>
                    <xsl:otherwise>
                        <origin>Unknown</origin>
                    </xsl:otherwise>
                </xsl:choose>
                <xsl:choose>
                    <xsl:when test="Dataset_Release_Date">
                        <xsl:apply-templates select="Dataset_Release_Date"/>
                    </xsl:when>
                    <xsl:otherwise>
                        <pubdate>Unknown</pubdate>
                    </xsl:otherwise>
                </xsl:choose>
                <xsl:choose>
                    <xsl:when test="Dataset_Title">
                        <xsl:apply-templates select="Dataset_Title"/>
                    </xsl:when>
                    <xsl:otherwise>
                        <xsl:apply-templates select="Entry_Title"/>
                    </xsl:otherwise>
                </xsl:choose>
            </citeinfo>
        </citation>
    </xsl:template>

```

```

        </xsl:otherwise>
    </xsl:choose>
    <edition><xsl:apply-templates select="Version"/></edition>
    <geoform><xsl:apply-templates
select="Data_Presentation_Form"/></geoform>
    <serinfo>
    <sername><xsl:apply-templates select="Dataset_Series_Name"/></sername>
    <issue><xsl:apply-templates select="Issue_Identification"/></issue>
    </serinfo>
    <pubinfo>
    <pubplace><xsl:apply-templates
select="Dataset_Release_Place"/></pubplace>
    <publish><xsl:apply-templates select="Dataset_Publisher"/></publish>
    </pubinfo>
    <othercit><xsl:apply-templates
select="Other_Citation_Details"/></othercit>
    <onlink><xsl:apply-templates select="Online_Resource"/></onlink>
    </citeinfo>
    </citation>
</xsl:template>

<!-- <xsl:apply-templates select="Originating_Metadata_Node" /> -->
<xsl:template match="Originating_Metadata_Node">
    <citation>
    <citeinfo>
    <origin>Unknown</origin>
    <pubdate>Unknown</pubdate>
    <xsl:apply-templates select="/DIF/Entry_Title"/>
    </citeinfo>
    </citation>
</xsl:template>

<!-- <xsl:apply-templates select="Dataset_Creator"/> etc-->
<xsl:template match="Dataset_Creator">
    <origin><xsl:value-of select="."/></origin>
</xsl:template>
<xsl:template match="Dataset_Release_Date">
    <pubdate><xsl:value-of select="translate(.,'-','')"/></pubdate>
</xsl:template>
<xsl:template match="Dataset_Title">
    <title><xsl:value-of select="."/></title>
</xsl:template>
<xsl:template match="Entry_Title">
    <title><xsl:value-of select="."/></title>
</xsl:template>
<xsl:template match="Version">
    <xsl:value-of select="."/>
</xsl:template>
<xsl:template match="Data_Presentation_Form">
    <xsl:value-of select="."/>
</xsl:template>
<xsl:template match="Dataset_Series_Name">
    <xsl:value-of select="."/>
</xsl:template>
<xsl:template match="Issue_Identification">
    <xsl:value-of select="."/>
</xsl:template>

```

```

<xsl:template match="Dataset_Release_Place">
  <xsl:value-of select="."/>
</xsl:template>
<xsl:template match="Dataset_Publisher">
  <xsl:value-of select="."/>
</xsl:template>
<xsl:template match="Dataset_Series_Name">
  <xsl:value-of select="."/>
</xsl:template>
<xsl:template match="Other_Citation_Details">
  <xsl:value-of select="."/>
</xsl:template>
<xsl:template match="Online_Resource">
  <xsl:value-of select="."/>
</xsl:template>

<!-- <xsl:apply-templates select="Summary"/> -->
<xsl:template match="Summary">
  <xsl:value-of select="."/>
</xsl:template>

<!-- <xsl:apply-templates select="Reference"/> -->
<xsl:template match="Reference">
  <supplinf><xsl:text> REFERENCE: </xsl:text><xsl:value-of
select="."/></supplinf>
</xsl:template>

<!-- <xsl:apply-templates select="Temporal_Coverage"/> -->
<xsl:template match="Temporal_Coverage">
<timeperd>
<timeinfo>
<rngdates>
  <xsl:choose>
    <xsl:when test="Start_Date">
      <xsl:apply-templates select="Start_Date"/>
    </xsl:when>
    <xsl:otherwise>
      <begdate>Unknown</begdate>
    </xsl:otherwise>
  </xsl:choose>
  <xsl:choose>
    <xsl:when test="Stop_Date">
      <xsl:apply-templates select="Stop_Date"/>
    </xsl:when>
    <xsl:otherwise>
      <enddate>Unknown</enddate>
    </xsl:otherwise>
  </xsl:choose>
</rngdates>
</timeinfo>
<current>Unknown</current>
</timeperd>
</xsl:template>

<xsl:template match="Start_Date">
  <begdate><xsl:value-of select="translate(.,'-','')"/></begdate>
</xsl:template>

```

```

<xsl:template match="Stop_Date">
  <enddate><xsl:value-of select="translate(.,'-','')"/></enddate>
</xsl:template>

<!-- <xsl:apply-templates select="Data_Set_Progress"/> -->
<xsl:template match="Data_Set_Progress">
  <status>
  <progress>
  <xsl:call-template name="str:capitalise">
    <xsl:with-param name="text" select="."/>
    <xsl:with-param name="all" select="true()"/>
  </xsl:call-template></progress>
  <update>As needed</update>
  </status>
</xsl:template>

<!-- <xsl:apply-templates select="Spatial_Coverage"/> -->
<xsl:template match="Spatial_Coverage">
  <spdom>
  <bounding>
    <xsl:apply-templates select="Westernmost_Longitude"/>
    <xsl:apply-templates select="Easternmost_Longitude"/>
    <xsl:apply-templates select="Northernmost_Latitude"/>
    <xsl:apply-templates select="Southernmost_Latitude"/>
  </bounding>
  </spdom>
</xsl:template>
<xsl:template match="Westernmost_Longitude">
  <westbc><xsl:value-of select="."/></westbc>
</xsl:template>
<xsl:template match="Easternmost_Longitude">
  <eastbc><xsl:value-of select="."/></eastbc>
</xsl:template>
<xsl:template match="Southernmost_Latitude">
  <southbc><xsl:value-of select="."/></southbc>
</xsl:template>
<xsl:template match="Northernmost_Latitude">
  <northbc><xsl:value-of select="."/></northbc>
</xsl:template>

<!-- <xsl:apply-templates select="Parameters"/> -->
<xsl:template match="Entry_ID">
  <themekey><xsl:value-of select="."/></themekey>
</xsl:template>

<xsl:template match="Parameters">
  <themekey>
  <xsl:apply-templates select="Category"/>
  <xsl:apply-templates select="Topic"/>
  <xsl:apply-templates select="Term"/>
  <xsl:apply-templates select="Variable"/>
  <xsl:apply-templates select="Detailed_Variable"/>
  </themekey>
</xsl:template>
<xsl:template match="Category">
  <xsl:call-template name="str:to-upper">
  <xsl:with-param name="text" select="."/>

```



```

        <xsl:with-param name="all" select="true()"/>
    </xsl:call-template>
</xsl:template>
<xsl:template match="Topic"> &gt; <xsl:call-template name="str:to-upper">
    <xsl:with-param name="text" select="."/>
    <xsl:with-param name="all" select="true()"/>
</xsl:call-template>
</xsl:template>
<xsl:template match="Term"> &gt; <xsl:call-template name="str:to-upper">
    <xsl:with-param name="text" select="."/>
    <xsl:with-param name="all" select="true()"/>
</xsl:call-template>
</xsl:template>
<xsl:template match="Variable"> &gt; <xsl:call-template name="str:to-upper">
    <xsl:with-param name="text" select="."/>
    <xsl:with-param name="all" select="true()"/>
</xsl:call-template>
</xsl:template>
<xsl:template match="Detailed_Variable"> &gt; <xsl:call-template
name="str:to-upper">
    <xsl:with-param name="text" select="."/>
    <xsl:with-param name="all" select="true()"/>
</xsl:call-template>
</xsl:template>

<xsl:template match="Source_Name">
    <themekey>
        <xsl:apply-templates select="Short_Name"/>
        <xsl:apply-templates select="Long_Name"/>
    </themekey>
</xsl:template>
<xsl:template match="Sensor_Name">
    <themekey>
        <xsl:apply-templates select="Short_Name"/>
        <xsl:apply-templates select="Long_Name"/>
    </themekey>
</xsl:template>
<xsl:template match="Project">
    <themekey>
        <xsl:apply-templates select="Short_Name"/>
        <xsl:apply-templates select="Long_Name"/>
    </themekey>
</xsl:template>
<xsl:template match="Short_Name">
    <xsl:call-template name="str:to-upper">
    <xsl:with-param name="text" select="."/>
    <xsl:with-param name="all" select="true()"/>
</xsl:call-template>
</xsl:template>
<xsl:template match="Long_Name"> &gt; <xsl:call-template name="str:to-upper">
    <xsl:with-param name="text" select="."/>
    <xsl:with-param name="all" select="true()"/>
</xsl:call-template>
</xsl:template>

<xsl:template match="Keyword">
    <themekey>

```

```

        <xsl:call-template name="str:capitalise">
        <xsl:with-param name="text" select="."/>
        <xsl:with-param name="all" select="true()"/>
        </xsl:call-template>
    </themekey>
</xsl:template>

<xsl:template match="Location">
    <placekey>
        <xsl:apply-templates select="Location_Name"/>
        <xsl:apply-templates select="Detailed_Location"/>
    </placekey>
</xsl:template>
<xsl:template match="Location_Name">
    <xsl:call-template name="str:to-upper">
    <xsl:with-param name="text" select="normalize-space(.)"/>
    <xsl:with-param name="all" select="true()"/>
    </xsl:call-template>
</xsl:template>
<xsl:template match="Detailed_Location"> &gt; <xsl:call-template
name="str:to-upper">
    <xsl:with-param name="text" select="normalize-space(.)"/>
    <xsl:with-param name="all" select="true()"/>
    </xsl:call-template>
</xsl:template>

<xsl:template match="/DIF/Data_Resolution/Temporal_Resolution">
    <tempkey><xsl:call-template name="str:capitalise">
    <xsl:with-param name="text" select="."/>
    <xsl:with-param name="all" select="true()"/>
    </xsl:call-template>
    </tempkey>
</xsl:template>

<xsl:template match="ISO_Topic_Category">
    <themekey><xsl:call-template name="str:to-upper">
    <xsl:with-param name="text" select="."/>
    <xsl:with-param name="all" select="true()"/>
    </xsl:call-template>
    </themekey>
</xsl:template>

<xsl:template match="Data_Set_Language">
    <themekey><xsl:call-template name="str:capitalise">
    <xsl:with-param name="text" select="."/>
    <xsl:with-param name="all" select="true()"/>
    </xsl:call-template>
    </themekey>
</xsl:template>

<!-- <xsl:apply-templates select="Access and Use Constraints"/> -->
<xsl:template match="Access_Constraints">
    <acconst><xsl:value-of select="."/></acconst>
</xsl:template>
<xsl:template match="Use_Constraints">
    <useconst><xsl:value-of select="."/></useconst>
</xsl:template>

```

```

<!-- <xsl:apply-templates select="Multimedia_Sample"/> -->
<xsl:template match="Multimedia_Sample">
  <browse>
    <xsl:choose>
      <xsl:when test="URL">
        <browsten><xsl:apply-templates select="URL"/></browsten>
      </xsl:when>
      <xsl:otherwise>
        <browsten><xsl:apply-templates select="File"/></browsten>
      </xsl:otherwise>
    </xsl:choose>
    <xsl:choose>
      <xsl:when test="Description">
        <browsted><xsl:apply-templates select="Description"/></browsted>
      </xsl:when>
      <xsl:otherwise>
        <browsted><xsl:apply-templates select="Caption"/></browsted>
      </xsl:otherwise>
    </xsl:choose>
    <browset><xsl:apply-templates select="Format"/></browset>
  </browse>
</xsl:template>

<xsl:template match="URL">
  <xsl:value-of select="."/>
</xsl:template>
<xsl:template match="File">
  <xsl:value-of select="."/>
</xsl:template>
<xsl:template match="Description">
  <xsl:value-of select="."/>
</xsl:template>
<xsl:template match="Caption">
  <xsl:value-of select="."/>
</xsl:template>
<xsl:template match="Format">
  <xsl:value-of select="."/>
</xsl:template>

<!-- <xsl:apply-templates select="Reference"/> -->
<!-- NO MATCH BECAUSE FGDC uses citation fields, GCMD one "Reference" fields -->

<!-- <xsl:apply-templates select="Data_Resolution"/> -->
<xsl:template match="Data_Resolution">
  <xsl:if test="Latitude_Resolution/text() != '' or
              Longitude_Resolution/text() != ''">
    <horizsys>
      <geograph>
        <xsl:apply-templates select="Latitude_Resolution"/>
        <xsl:apply-templates select="Longitude_Resolution"/>
      </geogunit>Decimal degrees</geogunit>
    </geograph>
  </horizsys>
</xsl:if>

```

```

        <xsl:if test="Vertical_Resolution/text() != '' ">
            <vertdef>
                <xsl:apply-templates select="Vertical_Resolution"/>
            </vertdef>
        </xsl:if>
    </xsl:template>

<xsl:template match="Latitude_Resolution">
    <latres><xsl:value-of select="."/></latres>
</xsl:template>
<xsl:template match="Longitude_Resolution">
    <longres><xsl:value-of select="."/></longres>
</xsl:template>
<!-- Vertical Resolution is match to both altitude and depth resolution in
FGDC -->
<xsl:template match="Vertical_Resolution">
    <altsys>
        <altdatum>Not Available</altdatum> <!-- Mandatory field -->
        <altres><xsl:value-of select="."/></altres> <!-- Mandatory field -->
        <altunits>Not Available</altunits> <!-- Mandatory field -->
        <altenc>Implicit coordinate</altenc> <!-- Mandatory field -->
    </altsys>
    <depthsys>
        <depthdn>Not Available</depthdn> <!-- Mandatory field -->
        <depthres><xsl:value-of select="."/></depthres> <!-- Mandatory field --
    >
        <depthdu>Not Available</depthdu> <!-- Mandatory field -->
        <depthem>Implicit coordinate</depthem> <!-- Mandatory field -->
    </depthsys>
</xsl:template>

<!-- <xsl:apply-templates select="Quality"/> -->
<xsl:template match="Quality">
    <attracc>
        <attraccr><xsl:value-of select="."/></attraccr>
    </attracc>
    <logic>Not Available</logic> <!-- Mandatory in data quality -->
    <complete>Not Available</complete> <!-- Mandatory in data quality -->
    <lineage>
        <procstep>
            <procdesc>Not Available</procdesc>
            <procdate>Unknown</procdate>
        </procstep>
    </lineage> <!-- Mandatory in data quality -->
</xsl:template>

<!-- <xsl:apply-templates select="Data_Center and Distribution information"/>
-->
<xsl:template match="Data_Center">
    <distrib>
        <cntinfo>
            <cntorgp>
                <cntorg>
                    <xsl:apply-templates select="Data_Center_Name/Short_Name"/>
                    <xsl:apply-templates select="Data_Center_Name/Long_Name"/>
                </cntorg>
            </cntorgp>
        </cntinfo>
    </distrib>

```

```

        <xsl:apply-templates select="/DIF/Data_Center/Personnel/First_Name"/>
        <xsl:apply-templates select="/DIF/Data_Center/Personnel/Middle_Name"/>
        <xsl:apply-templates select="/DIF/Data_Center/Personnel/Last_Name"/>
    </cntper>
</cntorgp>
<!-- <xsl:apply-templates select="Personnel"/>
Contact_Information permits only one of Contact_Person_Primary or
Contact_Organization_Primary -->
<!-- <xsl:apply-templates select="/DIF/Data_Center/Personnel/Role"/> -->
<cntpos>DATA CENTER CONTACT</cntpos>
<xsl:apply-templates select="/DIF/Data_Center/Personnel/Contact_Address"
/>
    <xsl:apply-templates select="/DIF/Data_Center/Personnel/Phone" />
    <xsl:apply-templates select="/DIF/Data_Center/Personnel/Fax" />
    <xsl:apply-templates select="/DIF/Data_Center/Personnel/Email" />
</cntinfo>
</distrib>
</xsl:template>

<xsl:template match="Data_Center_Name/Short_Name"><xsl:value-of
select="."/></xsl:template>
<xsl:template match="Data_Center_Name/Long_Name"> &gt; <xsl:value-of
select="."/></xsl:template>

<xsl:template match="Data_Center/Data_Set_ID"><resdesc><xsl:value-of
select="."/></resdesc></xsl:template>

<!-- Distribution information -->
<xsl:template match="Distribution">
    <xsl:choose>
        <xsl:when test="Distribution_Format">
            <xsl:apply-templates select="Distribution_Format"/>
        </xsl:when>
        <xsl:otherwise>
            <formname>Not Available</formname>
        </xsl:otherwise>
    </xsl:choose>
    <xsl:choose>
        <xsl:when test="Distribution_Size">
            <xsl:apply-templates select="Distribution_Size"/>
        </xsl:when>
        <xsl:otherwise>
            </xsl:otherwise>
        </xsl:choose>
</xsl:template>

<xsl:template match="Distribution_Format"><formname><xsl:value-of
select="."/></formname></xsl:template>
<xsl:template match="Distribution_Size"><transize><xsl:value-of
select="."/></transize></xsl:template>

<xsl:template match="/DIF/Data_Center/Data_Center_URL">
    <onlinopt>
    <computer>
    <networka>
    <networkr><xsl:value-of select="."/></networkr>

```

```

        </networka>
        </computer>
        <accinstr>DATA CENTER URL</accinstr>
        </onlinopt>
    </xsl:template>

    <xsl:template match="Related_URL">
        <xsl:choose>
            <xsl:when test="URL">
                <digtopt>
                    <onlinopt>
                        <computer>
                            <networka>
                                <networkkr><xsl:apply-templates select="URL"/></networkkr>
                            </networka>
                        </computer>
                    <accinstr><xsl:apply-templates select="Description"/></accinstr>
                </onlinopt>
            </digtopt>
            </xsl:when>
            <xsl:otherwise>
                <digtopt><onlinopt><computer><networka><networkkr>Not
Available</networkkr></networka></computer></onlinopt></digtopt>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:template>
    <xsl:template match="URL"><xsl:value-of select="."/></xsl:template>
    <xsl:template match="Description"><xsl:value-of select="."/></xsl:template>

    <xsl:template match="/DIF/Distribution/Fees">
        <xsl:choose>
            <xsl:when test="/DIF/Distribution/Fees">
                <xsl:apply-templates select="/DIF/Distribution/Fees"/>
            </xsl:when>
            <xsl:otherwise>
                <fees>Not Available</fees>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:template>

    <xsl:template match="/DIF/Distribution/Fees">
        <fees><xsl:call-template name="str:capitalise">
            <xsl:with-param name="text" select="."/>
            <xsl:with-param name="all" select="true()"/>
        </xsl:call-template></fees>
    </xsl:template>

    <!--
    <xsl:template match="/DIF/Distribution/Fees"><fees><xsl:value-of
select="."/></fees></xsl:template>
-->

    <!-- Personnel/Contact Information -->
    <xsl:template match="Personnel">
        <cntperp>
            <cntper>
                <xsl:apply-templates select="First_Name"/>

```

```

        <xsl:apply-templates select="Middle_Name"/>
        <xsl:apply-templates select="Last_Name"/>
    </cntper>
</cntperp>
<xsl:apply-templates select="Role"/>
<xsl:choose>
    <xsl:when test="Contact_Address">
        <xsl:apply-templates select="Contact_Address"/>
    </xsl:when>
    <xsl:otherwise>
        <cntaddr>
            <addrtype>Mailing and Physical Address</addrtype>
            <address>Not Available</address>
            <city>Not Available</city>
            <state>Not Available</state>
            <postal>Not Available</postal>
        </cntaddr>
    </xsl:otherwise>
</xsl:choose>
<xsl:apply-templates select="Phone" />
<xsl:apply-templates select="Fax" />
<xsl:apply-templates select="Email" />
</xsl:template>

<xsl:template match="Contact_Address">
    <cntaddr>
        <addrtype>Mailing and Physical Address</addrtype>
        <xsl:choose>
            <xsl:when test="Address">
                <xsl:apply-templates select="Address"/>
            </xsl:when>
            <xsl:otherwise>
                <address>Not Available</address>
            </xsl:otherwise>
        </xsl:choose>
        <xsl:choose>
            <xsl:when test="City">
                <xsl:apply-templates select="City"/>
            </xsl:when>
            <xsl:otherwise>
                <city>Not Available</city>
            </xsl:otherwise>
        </xsl:choose>
        <xsl:choose>
            <xsl:when test="Province_or_State">
                <xsl:apply-templates select="Province_or_State"/>
            </xsl:when>
            <xsl:otherwise>
                <state>Not Available</state>
            </xsl:otherwise>
        </xsl:choose>
        <xsl:choose>
            <xsl:when test="Postal_Code">
                <xsl:apply-templates select="Postal_Code"/>
            </xsl:when>
            <xsl:otherwise>
                <postal>Not Available</postal>
            </xsl:otherwise>
        </xsl:choose>
    </cntaddr>

```

```

        </xsl:otherwise>
    </xsl:choose>
<xsl:choose>
    <xsl:when test="Country">
        <xsl:apply-templates select="Country"/>
    </xsl:when>
    <xsl:otherwise>
        <postal>Not Available</postal>
    </xsl:otherwise>
</xsl:choose>
</cntaddr>
</xsl:template>

<xsl:template match="First_Name"><xsl:value-of select="."/><xsl:text>
</xsl:text></xsl:template>
<xsl:template match="Middle_Name"><xsl:value-of select="."/><xsl:text>
</xsl:text></xsl:template>
<xsl:template match="Last_Name"><xsl:value-of select="."/><xsl:text>
</xsl:text></xsl:template>
<xsl:template match="Role"><cntpos><xsl:value-of
select="."/></cntpos></xsl:template>
<xsl:template match="Phone"><cntvoice><xsl:value-of
select="."/></cntvoice></xsl:template>
<xsl:template match="Fax"><cntfax><xsl:value-of
select="."/></cntfax></xsl:template>
<xsl:template match="Email"><cntemail><xsl:value-of
select="."/></cntemail></xsl:template>
<xsl:template match="Address"><address><xsl:value-of select="."/><xsl:text>
</xsl:text></address></xsl:template>
<xsl:template match="City"><city><xsl:value-of
select="."/></city></xsl:template>
<xsl:template match="Province_or_State"><state><xsl:value-of
select="."/></state></xsl:template>
<xsl:template match="Postal_Code"><postal><xsl:value-of
select="."/></postal></xsl:template>
<xsl:template match="Country"><country><xsl:value-of
select="."/></country></xsl:template>

<!-- <xsl:apply-templates select="DIF_Creation_Date"/> -->
<xsl:template match="DIF_Creation_Date"><metd><xsl:value-of
select="translate(.,'-','')"/></metd></xsl:template>
<xsl:template match="Last_DIF_Revision_Date"><metrd><xsl:value-of
select="translate(.,'-','')"/></metrd></xsl:template>
<xsl:template match="Future_DIF_Review_Date"><metfrd><xsl:value-of
select="translate(.,'-','')"/></metfrd></xsl:template>

</xsl:stylesheet>

```

16.3.3. *String-modified stylesheet*

[A copy of this stylesheet (string-modified.xml) can be found at the PICES web site.]

```

<?xml version="1.0"?>

<xsl:stylesheet version="1.0"

```



```
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:doc="http://xslt.org/xsl/documentation/1.0"
xmlns:str="http://xslt.org/string"
extension-element-prefixes="doc str">
```

```
<doc:reference xmlns="">
  <referenceinfo>
    <releaseinfo role="meta">
      $Id: string-modified.xsl,v 1.1 2004/06/15 19:28:11 fstella Exp $
    </releaseinfo>
    <author>
      <surname>Ball</surname>
      <firstname>Steve</firstname>
    </author>
    <copyright>
      <year>2001</year>
      <holder>Steve Ball</holder>
    </copyright>
  </referenceinfo>

  <title>String Processing</title>

  <partintro>
    <section>
      <title>Introduction</title>

      <para>This module provides templates for manipulating strings.</para>

    </section>
  </partintro>
</doc:reference>
```

```
<!-- Common string constants and datasets as XSL variables -->
```

```
<!-- str:lower and str:upper contain pairs of lower and upper case
characters. Below insanely long strings should contain the
official lower/uppercase pairs, making this stylesheet working
for every language on earth. Hopefully. -->
```

```
<!-- These values are not enough, however. There are some
exceptions, dealt with below. -->
```

```
<xsl:variable name="xslt-str-lower"
```

```
select=" '&#x0061;&#x0062;&#x0063;&#x0064;&#x0065;&#x0066;&#x0067;&#x0068;&#x0069;&#x006A;&#x006B;&#x006C;&#x006D;&#x006E;&#x006F;&#x0070;&#x0071;&#x0072;&#x0073;&#x0074;&#x0075;&#x0076;&#x0077;&#x0078;&#x0079;&#x007A;&#x00B5;&#x00E0;&#x00E1;&#x00E2;&#x00E3;&#x00E4;&#x00E5;&#x00E6;&#x00E7;&#x00E8;&#x00E9;&#x00EA;&#x00EB;&#x00EC;&#x00ED;&#x00EE;&#x00EF;&#x00F0;&#x00F1;&#x00F2;&#x00F3;&#x00F4;&#x00F5;&#x00F6;&#x00F8;&#x00F9;&#x00FA;&#x00FB;&#x00FC;&#x00FD;&#x00FE;&#x00FF;&#x0101;&#x0103;&#x0105;&#x0107;&#x0109;&#x010B;&#x010D;&#x010F;&#x0111;&#x0113;&#x0115;&#x0117;&#x0119;&#x011B;&#x011D;&#x011F;&#x0121;&#x0123;&#x0125;&#x0127;&#x0129;&#x012B;&#x012D;&#x012F;&#x0131;&#x0133;&#x0135;&#x0137;&#x013A;&#x013C;&#x013E;&#x0140;&#x0142;&#x0144;&#x0146;&#x0148;&#x014B;&#x014D;&#x014F;&#x0151;&#x0153;&#x0155;&#x0157;&#x0159;&#x015B;&#x015D;&#x015F;&#x0161;&#x0163;&#x0165;&#x0167;&#x0169;&#x016B;&#x016D;&#x016F;&#x0171;&#x0173;&#x0175;&#x0177;&#x017A;&#x017C;&#x017E;&#x017F;&#x0183;&#x0185;&#x0188;&#x018C;&#x0192;&#x0195;&#x0199;&#x01A1;&#x01A3;&#x01A5;&#x01A8;&#x01AD;&#x01B0;&#x01B4;&#x01B6;&#x01B9;&#x01BD;&#x01BF;&#x01C5;&#x01C6;&#x01C8;&#x01C9;&#
```

x01CB;ǌǎǐǒǔǖǘǚǜ
;ǝǟǡǣǥǧǩǫǭ�
1EF;ǲǳǵǹǻǽǿȁȃ&
#x0205;ȇȉȋȍȏȑȓȕ!
7;șțȝȟȣȥȧȩȫ&#x
022D;ȯȱȳɓɔɖɗəɛ
ɠɣɨɩɯɲɵʀʃ
88;ʊʋʒͅάέήία&#
x03B2;γδεζηθικλ
;μνξοπρςστ�
3C5;φχψωϊϋόύώ&
#x03D0;ϑϕϖϛϝϟϡϣ>
5;ϧϩϫϭϯϰϱϲϵ&#x
0430;бвгдежзий
клмнопрст
43;фхцчшщъыь&#
x044D;юяѐёђѓєѕі
;їјљњћќѝўџ�
461;ѣѥѧѩѫѭѯѱѳ&
#x0475;ѷѹѻѽѿҁҍҏI
1;ғҕҗҙқҝҟҡң&#x
04A5;ҧҩҫҭүұҳҵҷ
ҹһҽҿӂӄӈӌӑ
D3;ӕӗәӛӝӟӡӣӥ&#
x04E7;өӫӭӯӱӳӵӹա
;բգդեզէըթժ�
56B;լխծկհձղճմ&
#x0575;նշոչպջռսW
E;տրցւփքօֆḁ&#x
1E03;ḅḇḉḋḍḏḑḓḕ
ḗḙḛḝḟḡḣḥḧ
29;ḫḭḯḱḳḵḷḹḻ&#
x1E3D;ḿṁṃṅṇṉṋṍṏ
;ṑṓṕṗṙṛṝṟṡ
E63;ṥṧṩṫṭṯṱṳṵ&
#x1E77;ṹṻṽṿẁẃẅẇǨ
9;ẋẍẏẑẓẕẛạả&#x
1EA5;ầẩẫậắằẳẵặ
ẹẻẽếềểễệỉ
CB;ọỏốồổỗộớờ&#
x1EDF;ỡợụủứừửữự
;ỳỵỷỹἀἁἂἃἄ
F05;ἆἇἐἑἒἓἔἕἠ&
#x1F21;ἢἣἤἥἦἧἰἱǳ
2;ἳἴἵἶἷὀὁὂὃ&#x
1F44;ὅὑὓὕὗὠὡὢὣ
ὤὥὦὧὰάὲέὴ
75;ὶίὸόὺύὼώᾀ&#
x1F81;ᾂᾃᾄᾅᾆᾇᾐᾑᾒ
;ᾓᾔᾕᾖᾗᾠᾡᾢᾣ
FA4;ᾥᾦᾧᾰᾱᾳιῃῐ&
#x1FD1;ῠῡῥῳⅰⅱⅲⅳȗ
4;ⅵⅶⅷⅸⅹⅺⅻⅼⅽ&#x
217E;ⅿⓐⓑⓒⓓⓔⓕⓖⓗ
ⓘⓙⓚⓛⓜⓝⓞⓟⓠ$
E1;ⓢⓣⓤⓥⓦⓧⓨⓩａ&#
xFF42;ｃｄｅｆｇｈｉｊｋ


```

<xsl:template name="str:to-upper">
  <xsl:param name="text"/>

  <!-- Below exception is extracted from unicode's SpecialCasing.txt
       file. It's the german lowercase "eszett" (the thing looking
       like a greek beta) that's to become "SS" in uppercase (note:
       that are *two* characters, that's why it doesn't fit in the
       list of upper/lowercase characters). There are more
       characters in that file (103, excluding the locale-specific
       ones), but they seemed to be much less used to me and they
       add up to a hellish long stylesheet.... - Reinout -->
  <xsl:param name="modified-text">
    <xsl:call-template name="str:subst">
      <xsl:with-param name="text">
        <xsl:value-of select="$text"/>
      </xsl:with-param>
      <xsl:with-param name="replace">
        <xsl:text>&#x00DF;</xsl:text>
      </xsl:with-param>
      <xsl:with-param name="with">
        <xsl:text>&#x0053;</xsl:text>
        <xsl:text>&#x0053;</xsl:text>
      </xsl:with-param>
    </xsl:call-template>
  </xsl:param>

  <xsl:value-of select="translate($modified-text, $xsltsl-str-lower,
    $xsltsl-str-upper)"/>
</xsl:template>

<doc:template name="str:to-lower" xmlns="">
  <refpurpose>Make string lowercase</refpurpose>

  <refdescription>
    <para>Converts all uppercase letters to lowercase.</para>
  </refdescription>

  <refparameter>
    <variablelist>
      <varlistentry>
        <term>text</term>
        <listitem>
          <para>The string to be converted</para>
        </listitem>
      </varlistentry>
    </variablelist>
  </refparameter>

  <refreturn>
    <para>Returns string with all lowercase letters.</para>
  </refreturn>
</doc:template>

<xsl:template name="str:to-lower">
  <xsl:param name="text"/>

```

```

    <xsl:value-of select="translate($text, $xsltsl-str-upper, $xsltsl-str-
lower)"/>
  </xsl:template>

  <doc:template name="str:capitalise" xmlns="">
    <refpurpose>Capitalise string</refpurpose>

    <refdescription>
      <para>Converts first character of string to an uppercase letter. All
remaining characters are converted to lowercase.</para>
    </refdescription>

    <refparameter>
      <variablelist>
        <varlistentry>
          <term>text</term>
          <listitem>
            <para>The string to be capitalised</para>
          </listitem>
        </varlistentry>
        <varlistentry>
          <term>all</term>
          <listitem>
            <para>Boolean controlling whether all words in the string are
capitalised.</para>
            <para>Default is true.</para>
          </listitem>
        </varlistentry>
      </variablelist>
    </refparameter>

    <refreturn>
      <para>Returns string with first character uppcase and all remaining
characters lowercase.</para>
    </refreturn>
  </doc:template>

  <xsl:template name="str:capitalise">
    <xsl:param name="text"/>
    <xsl:param name="all" select="true()"/>
    <xsl:choose>
      <xsl:when test="$all and (contains($text, ' ') or contains($text, '
' )
or contains($text, '&#10;') or contains($text, '/') or contains($text, '-
'))">
        <xsl:variable name="firstword">
          <xsl:call-template name="str:substring-before-first">
            <xsl:with-param name="text" select="$text"/>
            <xsl:with-param name="chars" select="$xsltsl-str-ws"/>
          </xsl:call-template>
        </xsl:variable>
        <xsl:call-template name="str:capitalise">
          <xsl:with-param name="text">
            <xsl:value-of select="$firstword"/>
          </xsl:with-param>
          <xsl:with-param name="all" select="false()"/>
        </xsl:call-template>
      </xsl:when>
    </xsl:choose>
  </xsl:template>

```

```

    <xsl:value-of select="substring($text, string-length($firstword) + 1,
1)"/>
    <xsl:call-template name="str:capitalise">
      <xsl:with-param name="text">
        <xsl:value-of select="substring($text, string-length($firstword) +
2)"/>
      </xsl:with-param>
    </xsl:call-template>
  </xsl:when>

  <xsl:otherwise>
    <xsl:call-template name="str:to-upper">
      <xsl:with-param name="text" select="substring($text, 1, 1)"/>
    </xsl:call-template>
    <xsl:call-template name="str:to-lower">
      <xsl:with-param name="text" select="substring($text, 2)"/>
    </xsl:call-template>
  </xsl:otherwise>
</xsl:choose>
</xsl:template>

<doc:template name="str:to-camelcase" xmlns="">
  <refpurpose>Convert a string to one camelcase word</refpurpose>

  <refdescription>
    <para>Converts a string to one lowerCamelCase or UpperCamelCase
word, depending on the setting of the "upper"
parameter. UpperCamelCase is also called MixedCase while
lowerCamelCase is also called just camelCase. The template
removes any spaces, tabs and slashes, but doesn't deal with
other punctuation. It's purpose is to convert strings like
"hollow timber flush door" to a term suitable as identifier or
XML tag like "HollowTimberFlushDoor".
    </para>
  </refdescription>

  <refparameter>
    <variablelist>
    <varlistentry>
      <term>text</term>
      <listitem>
        <para>The string to be capitalised</para>
      </listitem>
    </varlistentry>
    <varlistentry>
      <term>upper</term>
      <listitem>
        <para>Boolean controlling whether the string becomes an
UpperCamelCase word or a lowerCamelCase word.</para>
        <para>Default is true.</para>
      </listitem>
    </varlistentry>
    </variablelist>
  </refparameter>

  <refreturn>

```

```

    <para>Returns string with first character uppercase and all remaining
characters lowercase.</para>
  </refreturn>
</doc:template>

```

```

<xsl:template name="str:to-camelcase">
  <xsl:param name="text"/>
  <xsl:param name="upper" select="true()"/>
  <!-- First change all 'strange' characters to spaces -->
  <xsl:param name="string-with-only-spaces">
    <xsl:value-of select="translate($text,concat($xsltsl-str-ws,'/'),'
')"/>
  </xsl:param>
  <!-- Then process them -->
  <xsl:param name="before-space-removal">
    <xsl:variable name="firstword">
      <xsl:call-template name="str:substring-before-first">
        <xsl:with-param name="text" select="$string-with-only-spaces"/>
        <xsl:with-param name="chars" select="$xsltsl-str-ws"/>
      </xsl:call-template>
    </xsl:variable>
    <xsl:choose>
      <xsl:when test="$upper">
        <xsl:call-template name="str:to-upper">
          <xsl:with-param name="text" select="substring($firstword, 1,
1)"/>
        </xsl:call-template>
        <xsl:call-template name="str:to-lower">
          <xsl:with-param name="text" select="substring($firstword, 2)"/>
        </xsl:call-template>
      </xsl:when>
      <xsl:otherwise>
        <xsl:call-template name="str:to-upper">
          <xsl:with-param name="text" select="$firstword"/>
        </xsl:call-template>
      </xsl:otherwise>
    </xsl:choose>

    <xsl:call-template name="str:capitalise">
      <xsl:with-param name="text">
        <xsl:value-of select="substring($string-with-only-spaces, string-
length($firstword) + 2)"/>
      </xsl:with-param>
      <xsl:with-param name="all" select="true()"/>
    </xsl:call-template>
  </xsl:param>
  <xsl:value-of select="translate($before-space-removal,' ','')"/>
</xsl:template>

<doc:template name="str:substring-before-first" xmlns="">
  <refpurpose>String extraction</refpurpose>

  <refdescription>
    <para>Extracts the portion of string 'text' which occurs before any of
the characters in string 'chars'.</para>
  </refdescription>

```



```

<refparameter>
  <variablelist>
<varlistentry>
  <term>text</term>
  <listitem>
    <para>The string from which to extract a substring.</para>
  </listitem>
</varlistentry>
<varlistentry>
  <term>chars</term>
  <listitem>
    <para>The string containing characters to find.</para>
  </listitem>
</varlistentry>
  </variablelist>
</refparameter>

<refreturn>
  <para>Returns string.</para>
</refreturn>
</doc:template>

<xsl:template name="str:substring-before-first">
  <xsl:param name="text"/>
  <xsl:param name="chars"/>

  <xsl:choose>

    <xsl:when test="string-length($text) = 0"/>

    <xsl:when test="string-length($chars) = 0">
<xsl:value-of select="$text"/>
    </xsl:when>

    <xsl:when test="contains($text, substring($chars, 1, 1))">
<xsl:variable name="this" select="substring-before($text,
substring($chars, 1, 1))"/>
<xsl:variable name="rest">
  <xsl:call-template name="str:substring-before-first">
    <xsl:with-param name="text" select="$text"/>
    <xsl:with-param name="chars" select="substring($chars, 2)"/>
  </xsl:call-template>
</xsl:variable>

<xsl:choose>
  <xsl:when test="string-length($this) &lt; string-length($rest)">
    <xsl:value-of select="$this"/>
  </xsl:when>
  <xsl:otherwise>
    <xsl:value-of select="$rest"/>
  </xsl:otherwise>
</xsl:choose>
  </xsl:when>

  <xsl:otherwise>
<xsl:call-template name="str:substring-before-first">
  <xsl:with-param name="text" select="$text"/>

```

```

    <xsl:with-param name="chars" select="substring($chars, 2)"/>
  </xsl:call-template>
  </xsl:otherwise>

</xsl:choose>
</xsl:template>

<doc:template name="str:substring-after-last" xmlns="">
  <refpurpose>String extraction</refpurpose>

  <refdescription>
    <para>Extracts the portion of string 'text' which occurs after the last
of the character in string 'chars'.</para>
  </refdescription>

  <refparameter>
    <variablelist>
  <varlistentry>
    <term>text</term>
    <listitem>
      <para>The string from which to extract a substring.</para>
    </listitem>
  </varlistentry>
  <varlistentry>
    <term>chars</term>
    <listitem>
      <para>The string containing characters to find.</para>
    </listitem>
  </varlistentry>
  </variablelist>
</refparameter>

  <refreturn>
    <para>Returns string.</para>
  </refreturn>
</doc:template>

<xsl:template name="str:substring-after-last">
  <xsl:param name="text"/>
  <xsl:param name="char"/>

  <xsl:choose>

    <xsl:when test="contains($text, $char)">
      <xsl:variable name="last" select="substring-after($text, $char)"/>

  <xsl:choose>
    <xsl:when test="contains($last, $char)">
      <xsl:call-template name="str:substring-after-last">
        <xsl:with-param name="text" select="$last"/>
        <xsl:with-param name="char" select="$char"/>
      </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="$last"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:choose>

```

```

    </xsl:when>

    <xsl:otherwise>
      <xsl:value-of select="$text" />
    </xsl:otherwise>

  </xsl:choose>
</xsl:template>

<doc:template name="str:substring-before-last" xmlns="">
  <refpurpose>String extraction</refpurpose>

  <refdescription>
    <para>Extracts the portion of string 'text' which occurs before the
first character of the last occurrence of string 'chars'.</para>
  </refdescription>

  <refparameter>
    <variablelist>
<varlistentry>
  <term>text</term>
  <listitem>
    <para>The string from which to extract a substring.</para>
  </listitem>
</varlistentry>
<varlistentry>
  <term>chars</term>
  <listitem>
    <para>The string containing characters to find.</para>
  </listitem>
</varlistentry>
    </variablelist>
  </refparameter>

  <refreturn>
    <para>Returns string.</para>
  </refreturn>
</doc:template>

<xsl:template name="str:substring-before-last">
  <xsl:param name="text" />
  <xsl:param name="chars" />

  <xsl:choose>

    <xsl:when test="string-length($text) = 0" />

    <xsl:when test="string-length($chars) = 0">
<xsl:value-of select="$text" />
    </xsl:when>

    <xsl:when test="contains($text, $chars)">
<xsl:call-template name="str:substring-before-last-aux">
  <xsl:with-param name="text" select="$text" />
  <xsl:with-param name="chars" select="$chars" />
</xsl:call-template>
    </xsl:when>

```

```

    <xsl:otherwise>
      <xsl:value-of select="$text"/>
    </xsl:otherwise>

  </xsl:choose>
</xsl:template>

<xsl:template name="str:substring-before-last-aux">
  <xsl:param name="text"/>
  <xsl:param name="chars"/>

  <xsl:choose>
    <xsl:when test="string-length($text) = 0"/>

    <xsl:when test="contains($text, $chars)">
<xsl:variable name="after">
  <xsl:call-template name="str:substring-before-last-aux">
    <xsl:with-param name="text" select="substring-after($text, $chars)"/>
    <xsl:with-param name="chars" select="$chars"/>
  </xsl:call-template>
</xsl:variable>

  <xsl:value-of select="substring-before($text, $chars)"/>
  <xsl:if test="string-length($after) > 0">
    <xsl:value-of select="$chars"/>
    <xsl:copy-of select="$after"/>
  </xsl:if>
</xsl:when>

    <xsl:otherwise/>
  </xsl:choose>
</xsl:template>

<doc:template name="str:subst" xmlns="">
  <refpurpose>String substitution</refpurpose>

  <refdescription>
    <para>Substitute 'replace' for 'with' in string 'text'.</para>
  </refdescription>

  <refparameter>
    <variablelist>
<varlistentry>
  <term>text</term>
  <listitem>
    <para>The string upon which to perform substitution</para>
  </listitem>
</varlistentry>
<varlistentry>
  <term>replace</term>
  <listitem>
    <para>The string to substitute</para>
  </listitem>
</varlistentry>
<varlistentry>
  <term>with</term>

```

```

    <listitem>
      <para>The string to be substituted</para>
    </listitem>
  </varlistentry>
</variablelist>
</refparameter>

<refreturn>
  <para>Returns string.</para>
</refreturn>
</doc:template>

<xsl:template name="str:subst">
  <xsl:param name="text" />
  <xsl:param name="replace" />
  <xsl:param name="with" />

  <xsl:choose>
    <xsl:when test="string-length($replace) = 0">
      <xsl:value-of select="$text" />
    </xsl:when>
    <xsl:when test="contains($text, $replace)">

<xsl:variable name="before" select="substring-before($text, $replace)" />
<xsl:variable name="after" select="substring-after($text, $replace)" />

<xsl:value-of select="$before" />
<xsl:value-of select="$with" />
      <xsl:call-template name="str:subst">
        <xsl:with-param name="text" select="$after" />
        <xsl:with-param name="replace" select="$replace" />
        <xsl:with-param name="with" select="$with" />
      </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="$text" />
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<doc:template name="str:count-substring" xmlns="">
  <refpurpose>Count Substrings</refpurpose>

  <refdescription>
    <para>Counts the number of times a substring occurs in a string. This
can also counts the number of times a character occurs in a string, since a
character is simply a string of length 1.</para>
  </refdescription>

  <example>
    <title>Counting Lines</title>
    <programlisting><![CDATA[
<xsl:call-template name="str:count-substring">
  <xsl:with-param name="text" select="$mytext" />
  <xsl:with-param name="chars" select="'&#x0a;' />
</xsl:call-template>
]]></programlisting>

```

```

</example>

<refparameter>
  <variablelist>
<varlistentry>
  <term>text</term>
  <listitem>
    <para>The source string.</para>
  </listitem>
</varlistentry>
<varlistentry>
  <term>chars</term>
  <listitem>
    <para>The substring to count.</para>
  </listitem>
</varlistentry>
  </variablelist>
</refparameter>

<refreturn>
  <para>Returns a non-negative integer value.</para>
</refreturn>
</doc:template>

<xsl:template name="str:count-substring">
  <xsl:param name="text" />
  <xsl:param name="chars" />

  <xsl:choose>
    <xsl:when test="string-length($text) = 0 or string-length($chars) = 0">
<xsl:text>0</xsl:text>
    </xsl:when>
    <xsl:when test="contains($text, $chars)">
<xsl:variable name="remaining">
    <xsl:call-template name="str:count-substring">
      <xsl:with-param name="text" select="substring-after($text, $chars)"/>
      <xsl:with-param name="chars" select="$chars"/>
    </xsl:call-template>
</xsl:variable>
<xsl:value-of select="$remaining + 1"/>
    </xsl:when>
    <xsl:otherwise>
<xsl:text>0</xsl:text>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<doc:template name="str:substring-after-at" xmlns="">
  <refpurpose>String extraction</refpurpose>
  <refdescription>
    <para>Extracts the portion of a 'char' delimited 'text' string "array"
at a given 'position' </para>
  </refdescription>
  <refparameter>
    <variablelist>
      <varlistentry>
        <term>text</term>

```

```

        <listitem>
            <para>The string from which to extract a substring.</para>
        </listitem>
    </varlistentry>
</varlistentry>
    <term>chars</term>
    <listitem>
        <para>delimiters</para>
    </listitem>
</varlistentry>
</varlistentry>
    <term>position</term>
    <listitem>
        <para>position of the elements</para>
    </listitem>
</varlistentry>
</variablelist>
</refparameter>
<refreturn>
    <para>Returns string.</para>
</refreturn>
</doc:template>

<xsl:template name="str:substring-after-at">
    <xsl:param name="text" />
    <xsl:param name="char" />
    <xsl:param name="position" />
    <xsl:choose>
        <xsl:when test="$position = 0 or not(contains($text , $char))">
            <xsl:choose>
                <xsl:when test="contains($text , $char)">
                    <xsl:value-of select="substring-before($text, $char)" />
                </xsl:when>
                <xsl:otherwise>
                    <xsl:value-of select="$text" />
                </xsl:otherwise>
            </xsl:choose>
        </xsl:when>
        <xsl:when test="contains($text , $char) and $position > 0">
            <xsl:variable name="last" select="substring-after($text, $char)" />

            <xsl:choose>
                <xsl:when test="$position > 0">
                    <xsl:call-template name="str:substring-after-at">
                        <xsl:with-param name="text" select="$last" />
                        <xsl:with-param name="char" select="$char" />
                        <xsl:with-param name="position" select="$position - 1" />
                    </xsl:call-template>
                </xsl:when>
            </xsl:choose>
        </xsl:when>
    </xsl:choose>
</xsl:template>

<doc:template name="str:insert-at" xmlns="">
    <refpurpose>String insertion</refpurpose>

```

```

<refdescription>
  <para>Insert 'chars' into 'text' at any given 'position'</para>
</refdescription>
<refparameter>
  <variablelist>
    <varlistentry>
      <term>text</term>
      <listitem>
        <para>The string upon which to perform insertion</para>
      </listitem>
    </varlistentry>
    <varlistentry>
      <term>position</term>
      <listitem>
        <para>the position where insertion will be performed</para>
      </listitem>
    </varlistentry>
    <varlistentry>
      <term>with</term>
      <listitem>
        <para>The string to be inserted</para>
      </listitem>
    </varlistentry>
  </variablelist>
</refparameter>
<refreturn>
  <para>Returns string.</para>
</refreturn>
</doc:template>

<xsl:template name="str:insert-at">
  <xsl:param name="text"/>
  <xsl:param name="position"/>
  <xsl:param name="chars"/>

  <xsl:variable name="firstpart" select="substring($text, 0, $position)"/>
  <xsl:variable name="secondpart" select="substring($text, $position,
string-length($text))"/>

  <xsl:value-of select="concat($firstpart, $chars, $secondpart)"/>
</xsl:template>

<doc:template name="str:backward" xmlns="">
  <refpurpose>String reversal</refpurpose>

  <refdescription>
    <para>Reverse the content of a given string</para>
  </refdescription>

  <refparameter>
    <variablelist>
      <varlistentry>
        <term>text</term>
        <listitem>
          <para>The string to be reversed</para>
        </listitem>
      </varlistentry>
    </variablelist>
  </refparameter>

```



```

        </varlistentry>
    </variablelist>
</refparameter>

<refreturn>
    <para>Returns string.</para>
</refreturn>
</doc:template>

<xsl:template name="str:backward">
    <xsl:param name="text"/>
    <xsl:variable name="mirror">
        <xsl:call-template name="str:build-mirror">
            <xsl:with-param name="text" select="$text"/>
            <xsl:with-param name="position" select="string-length($text)"/>
        </xsl:call-template>
    </xsl:variable>
    <xsl:value-of select="substring($mirror, string-length($text) + 1,
string-length($text))"/>
</xsl:template>

<xsl:template name="str:build-mirror">
    <xsl:param name="text"/>
    <xsl:param name="position"/>

    <xsl:choose>
        <xsl:when test="$position > 0">
            <xsl:call-template name="str:build-mirror">
                <xsl:with-param name="text" select="concat($text, substring($text,
$position, 1))"/>
                <xsl:with-param name="position" select="$position - 1"/>
            </xsl:call-template>
        </xsl:when>
        <xsl:otherwise>
            <xsl:value-of select="$text"/>
        </xsl:otherwise>
    </xsl:choose>
</xsl:template>

<doc:template name="str:character-first" xmlns="">
    <refpurpose>Find first occurring character in a string</refpurpose>

    <refdescription>
        <para>Finds which of the given characters occurs first in a
string.</para>
    </refdescription>

    <refparameter>
        <variablelist>
<varlistentry>
    <term>text</term>
    <listitem>
        <para>The source string.</para>
    </listitem>
</varlistentry>
<varlistentry>
    <term>chars</term>

```

```

    <listitem>
      <para>The characters to search for.</para>
    </listitem>
  </varlistentry>
</variablelist>
</refparameter>
</doc:template>

<xsl:template name="str:character-first">
  <xsl:param name="text"/>
  <xsl:param name="chars"/>

  <xsl:choose>
    <xsl:when test="string-length($text) = 0 or string-length($chars) =
0"/>

    <xsl:when test="contains($text, substring($chars, 1, 1))">
<xsl:variable name="next-character">
  <xsl:call-template name="str:character-first">
    <xsl:with-param name="text" select="$text"/>
    <xsl:with-param name="chars" select="substring($chars, 2)"/>
  </xsl:call-template>
</xsl:variable>

  <xsl:choose>
    <xsl:when test="string-length($next-character)">
      <xsl:variable name="first-character-position" select="string-
length(substring-before($text, substring($chars, 1, 1)))/>
      <xsl:variable name="next-character-position" select="string-
length(substring-before($text, $next-character))"/>

      <xsl:choose>
        <xsl:when test="$first-character-position < $next-character-
position">
          <xsl:value-of select="substring($chars, 1, 1)"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:value-of select="$next-character"/>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="substring($chars, 1, 1)"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:when>
  <xsl:otherwise>
    <xsl:call-template name="str:character-first">
      <xsl:with-param name="text" select="$text"/>
      <xsl:with-param name="chars" select="substring($chars, 2)"/>
    </xsl:call-template>
  </xsl:otherwise>
</xsl:choose>
</xsl:template>

<doc:template name="str:string-match" xmlns="">
  <refpurpose>Match A String To A Pattern</refpurpose>

```

```

<refdescription>
  <para>Performs globbing-style pattern matching on a string.</para>
</refdescription>

<example>
  <title>Match Pattern</title>
  <programlisting><![CDATA[
<xsl:call-template name="str:string-match">
  <xsl:with-param name="text" select="$mytext"/>
  <xsl:with-param name="pattern" select="'abc*def?g'"/>
</xsl:call-template>
]]></programlisting>
</example>

<refparameter>
  <variablelist>
<varlistentry>
  <term>text</term>
  <listitem>
    <para>The source string.</para>
  </listitem>
</varlistentry>
<varlistentry>
  <term>pattern</term>
  <listitem>
    <para>The pattern to match against. Certain characters have special
meaning:</para>
    <variablelist>
      <varlistentry>
        <term>*</term>
        <listitem>
          <para>Matches zero or more characters.</para>
        </listitem>
      </varlistentry>
      <varlistentry>
        <term>?</term>
        <listitem>
          <para>Matches a single character.</para>
        </listitem>
      </varlistentry>
      <varlistentry>
        <term>\</term>
        <listitem>
          <para>Character escape. The next character is taken as a literal
character.</para>
        </listitem>
      </varlistentry>
    </variablelist>
  </listitem>
</varlistentry>
  </variablelist>
</refparameter>

<refreturn>
  <para>Returns "1" if the string matches the pattern, "0"
otherwise.</para>

```

```

    </refreturn>
</doc:template>

<xsl:template name="str:string-match">
  <xsl:param name="text"/>
  <xsl:param name="pattern"/>

  <xsl:choose>
    <xsl:when test="$pattern = '*'">
      <!-- Special case: always matches -->
      <xsl:text>1</xsl:text>
    </xsl:when>
    <xsl:when test="string-length($text) = 0 and string-length($pattern) =
0">
      <xsl:text>1</xsl:text>
    </xsl:when>
    <xsl:when test="string-length($text) = 0 or string-length($pattern) =
0">
      <xsl:text>0</xsl:text>
    </xsl:when>
    <xsl:otherwise>
      <xsl:variable name='before-special'>
        <xsl:call-template name='str:substring-before-first'>
          <xsl:with-param name='text' select='$pattern' />
          <xsl:with-param name='chars' select='"*?\"' />
        </xsl:call-template>
      </xsl:variable>
      <xsl:variable name='special'>
        <xsl:call-template name='str:character-first'>
          <xsl:with-param name='text' select='$pattern' />
          <xsl:with-param name='chars' select='"*?\"' />
        </xsl:call-template>
      </xsl:variable>

      <xsl:variable name='new-text' select='substring($text, string-
length($before-special) + 1)' />
      <xsl:variable name='new-pattern' select='substring($pattern, string-
length($before-special) + 1)' />

      <xsl:choose>
        <xsl:when test="not(starts-with($text, $before-special))">
          <!-- Verbatim characters don't match -->
          <xsl:text>0</xsl:text>
        </xsl:when>

        <xsl:when test="$special = '*' and string-length($new-pattern) = 1">
          <xsl:text>1</xsl:text>
        </xsl:when>
        <xsl:when test="$special = '*'">
          <xsl:call-template name='str:match-postfix'>
            <xsl:with-param name='text' select='$new-text' />
            <xsl:with-param name='pattern' select='substring($new-pattern, 2)' />
          </xsl:call-template>
        </xsl:when>

        <xsl:when test="$special = '?'">
          <xsl:call-template name="str:string-match">

```

```

        <xsl:with-param name='text' select='substring($new-text, 2)'/>
        <xsl:with-param name='pattern' select='substring($new-pattern, 2)'/>
    </xsl:call-template>
</xsl:when>

    <xsl:when test="$special = '\' and substring($new-text, 1, 1) =
substring($new-pattern, 2, 1)">
        <xsl:call-template name="str:string-match">
            <xsl:with-param name='text' select='substring($new-text, 2)'/>
            <xsl:with-param name='pattern' select='substring($new-pattern, 3)'/>
        </xsl:call-template>
    </xsl:when>
    <xsl:when test="$special = '\' and substring($new-text, 1, 1) !=
substring($new-pattern, 2, 1)">
        <xsl:text>0</xsl:text>
    </xsl:when>

    <xsl:otherwise>
        <!-- There were no special characters in the pattern -->
        <xsl:choose>
            <xsl:when test='$text = $pattern'>
                <xsl:text>1</xsl:text>
            </xsl:when>
            <xsl:otherwise>
                <xsl:text>0</xsl:text>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:otherwise>
</xsl:choose>
</xsl:otherwise>
</xsl:choose>
</xsl:choose>
</xsl:template>

<xsl:template name="str:match-postfix">
    <xsl:param name="text"/>
    <xsl:param name="pattern"/>

    <xsl:variable name='result'>
        <xsl:call-template name='str:string-match'>
            <xsl:with-param name='text' select='$text' />
            <xsl:with-param name='pattern' select='$pattern' />
        </xsl:call-template>
    </xsl:variable>

    <xsl:choose>
        <xsl:when test='$result = "1"'>
            <xsl:value-of select='$result' />
        </xsl:when>
        <xsl:when test='string-length($text) = 0'>
            <xsl:text>0</xsl:text>
        </xsl:when>
        <xsl:otherwise>
            <xsl:call-template name='str:match-postfix'>
                <xsl:with-param name='text' select='substring($text, 2)'/>
                <xsl:with-param name='pattern' select='$pattern' />
            </xsl:call-template>
        </xsl:otherwise>
    </xsl:choose>

```

```

    </xsl:choose>
</xsl:template>
<doc:template name="str:generate-string" xmlns="">
  <refpurpose>Create A Repeating Sequence of Characters</refpurpose>
  <refdescription>
    <para>Repeats a string a given number of times.</para>
  </refdescription>
  <refparameter>
    <variablelist>
<varlistentry>
  <term>text</term>
  <listitem>
    <para>The string to repeat.</para>
  </listitem>
</varlistentry>
<varlistentry>
  <term>count</term>
  <listitem>
    <para>The number of times to repeat the string.</para>
  </listitem>
</varlistentry>
    </variablelist>
  </refparameter>
</doc:template>
<xsl:template name="str:generate-string">
  <xsl:param name="text" />
  <xsl:param name="count" />
  <xsl:choose>
    <xsl:when test="string-length($text) = 0 or $count &lt;= 0"/>
    <xsl:otherwise>
      <xsl:value-of select="$text" />
      <xsl:call-template name="str:generate-string">
        <xsl:with-param name="text" select="$text" />
        <xsl:with-param name="count" select="$count - 1" />
      </xsl:call-template>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
</xsl:stylesheet>

```

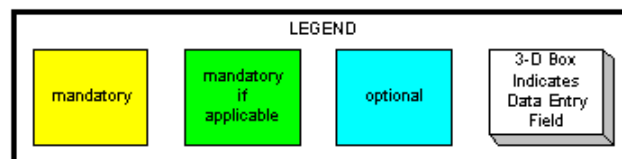
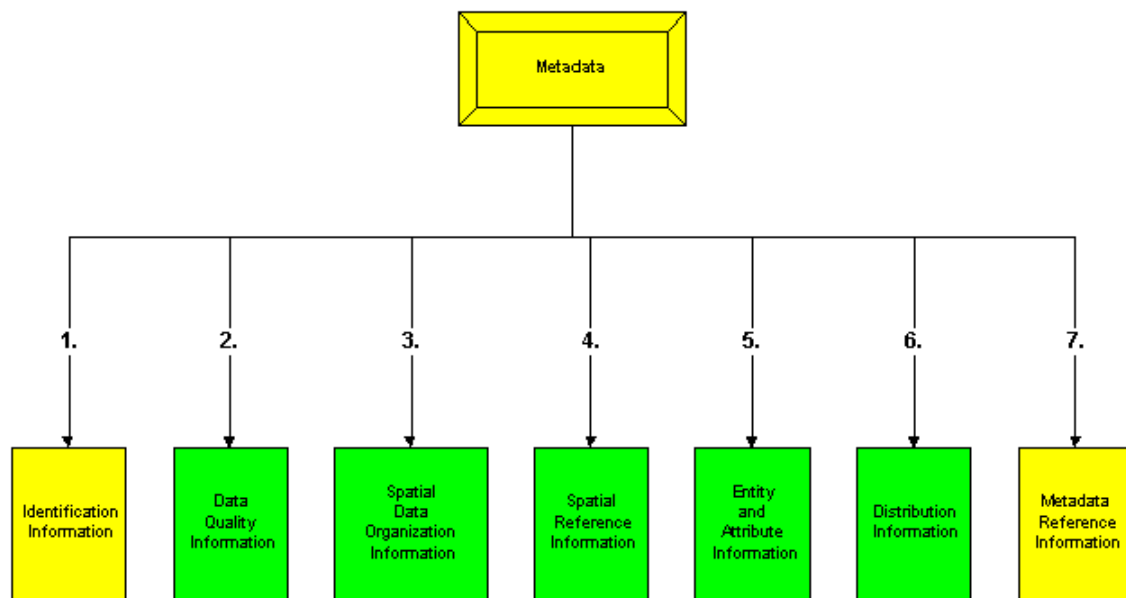
16.4. FGDC Metadata Standard

16.4.1. Overall structure

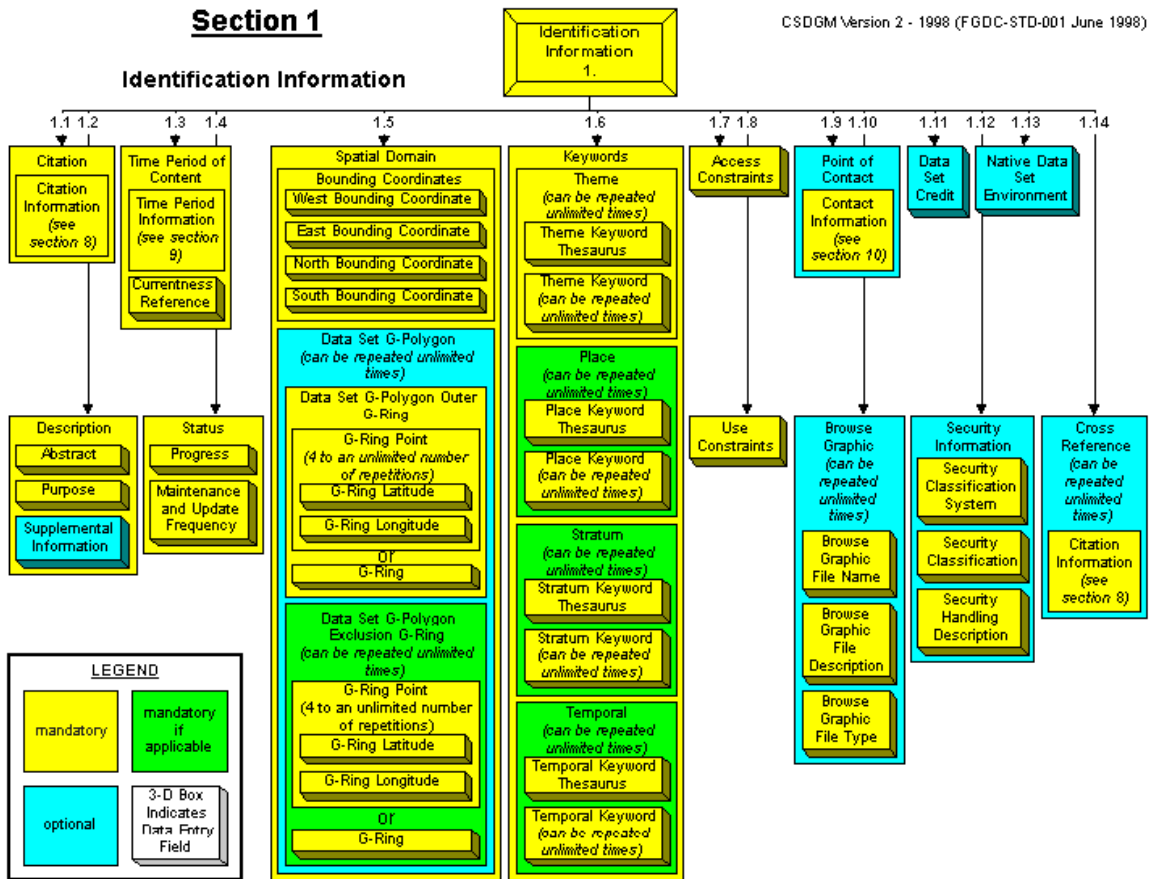
Graphical Representation of: The Federal Geographic Data Committee's Content Standard for Digital Geospatial Metadata

FGDC-STD-001, June 1998

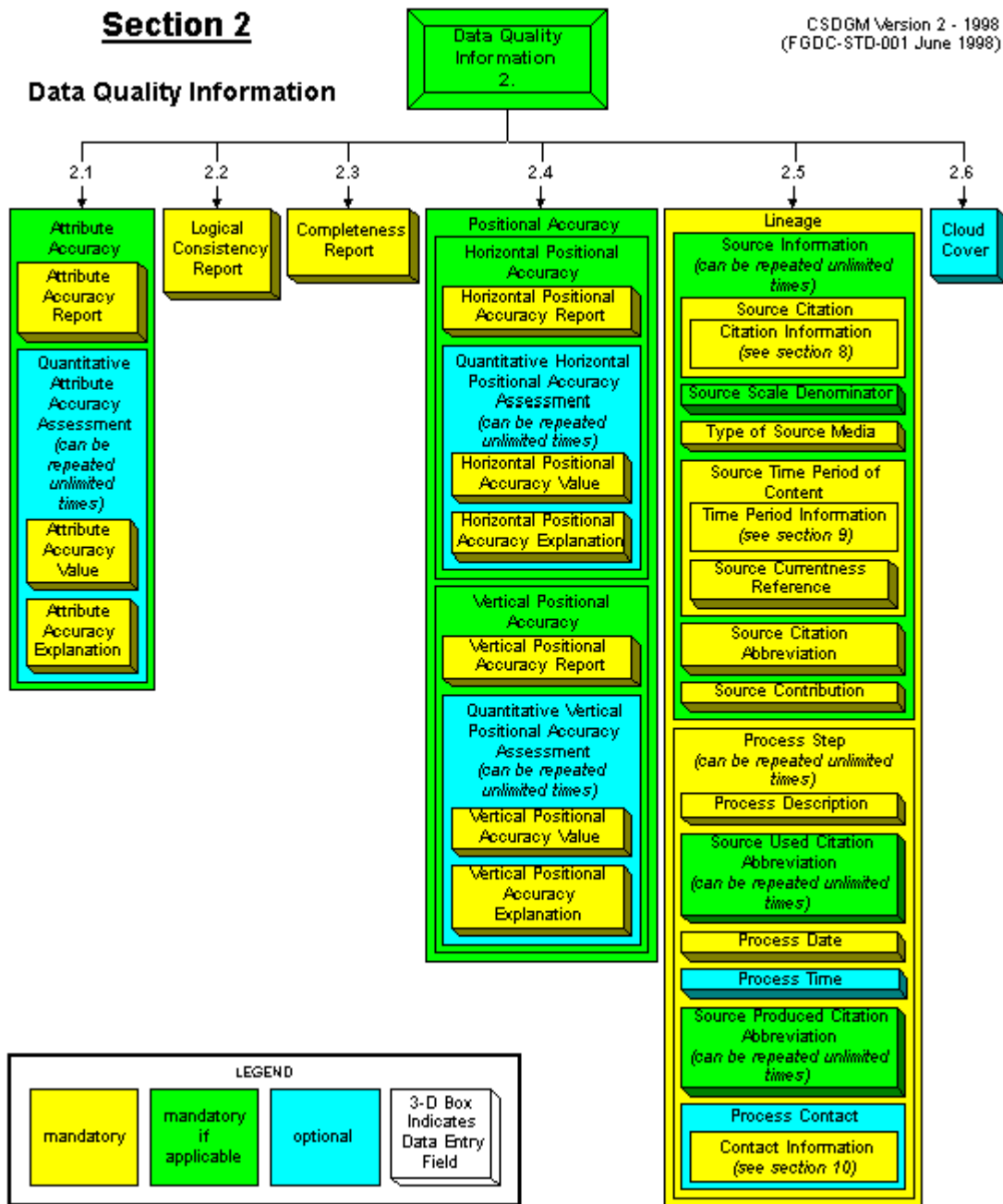
Prepared by Susan Stitt, Center for Biological Informatics, USGS
In conjunction with the Federal Geographic Data Committee



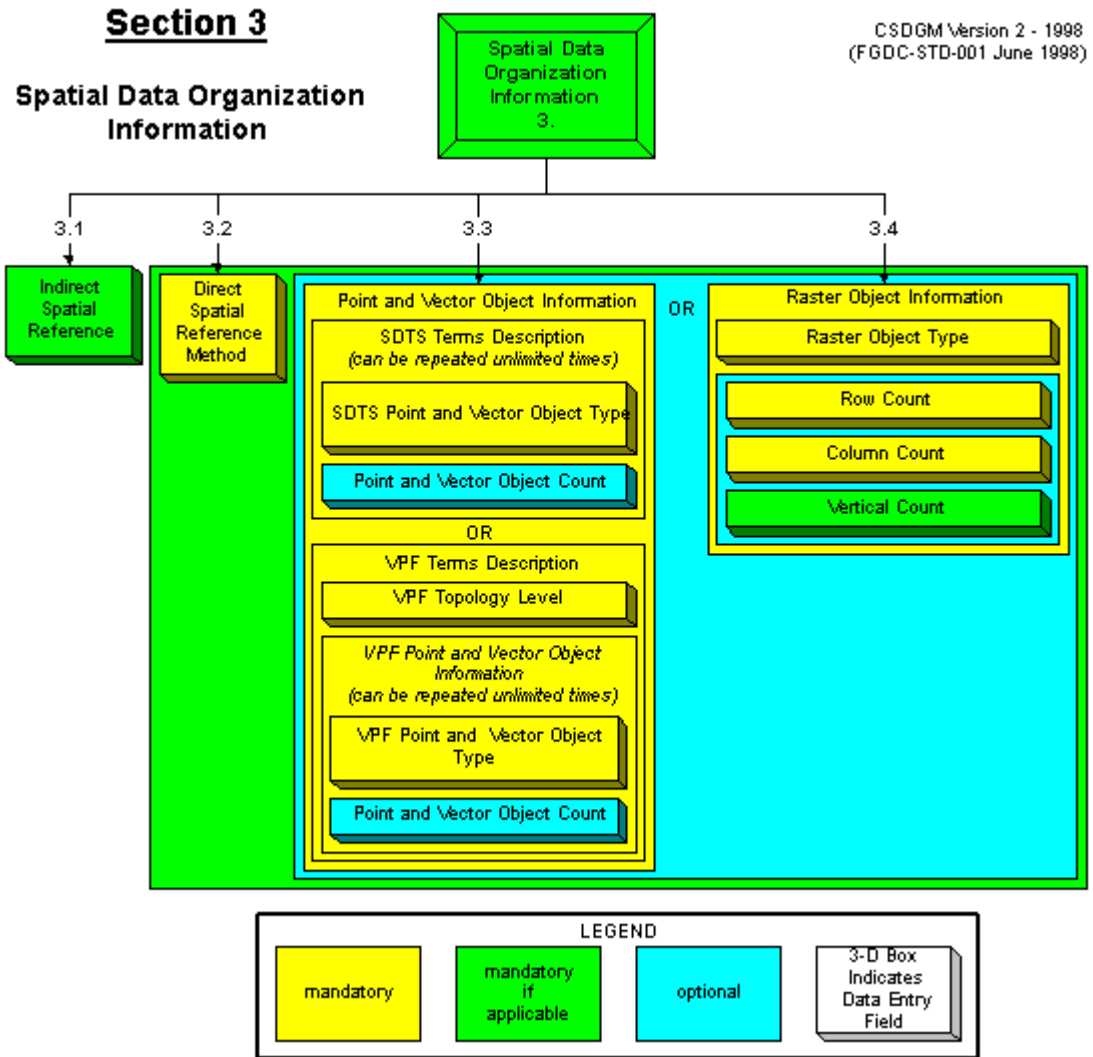
16.4.2. Section 1: Identification information



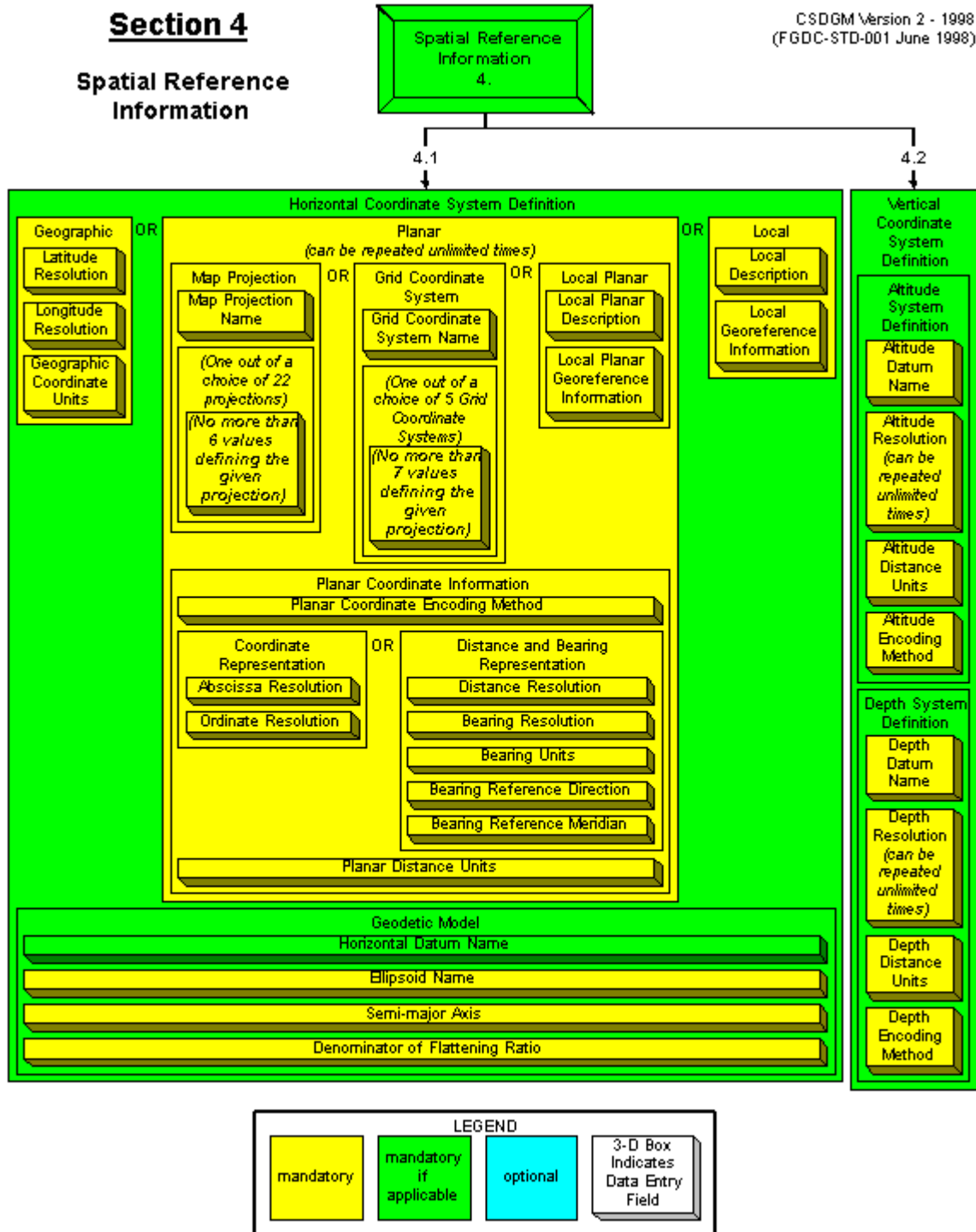
16.4.3. Section 2: Data quality information



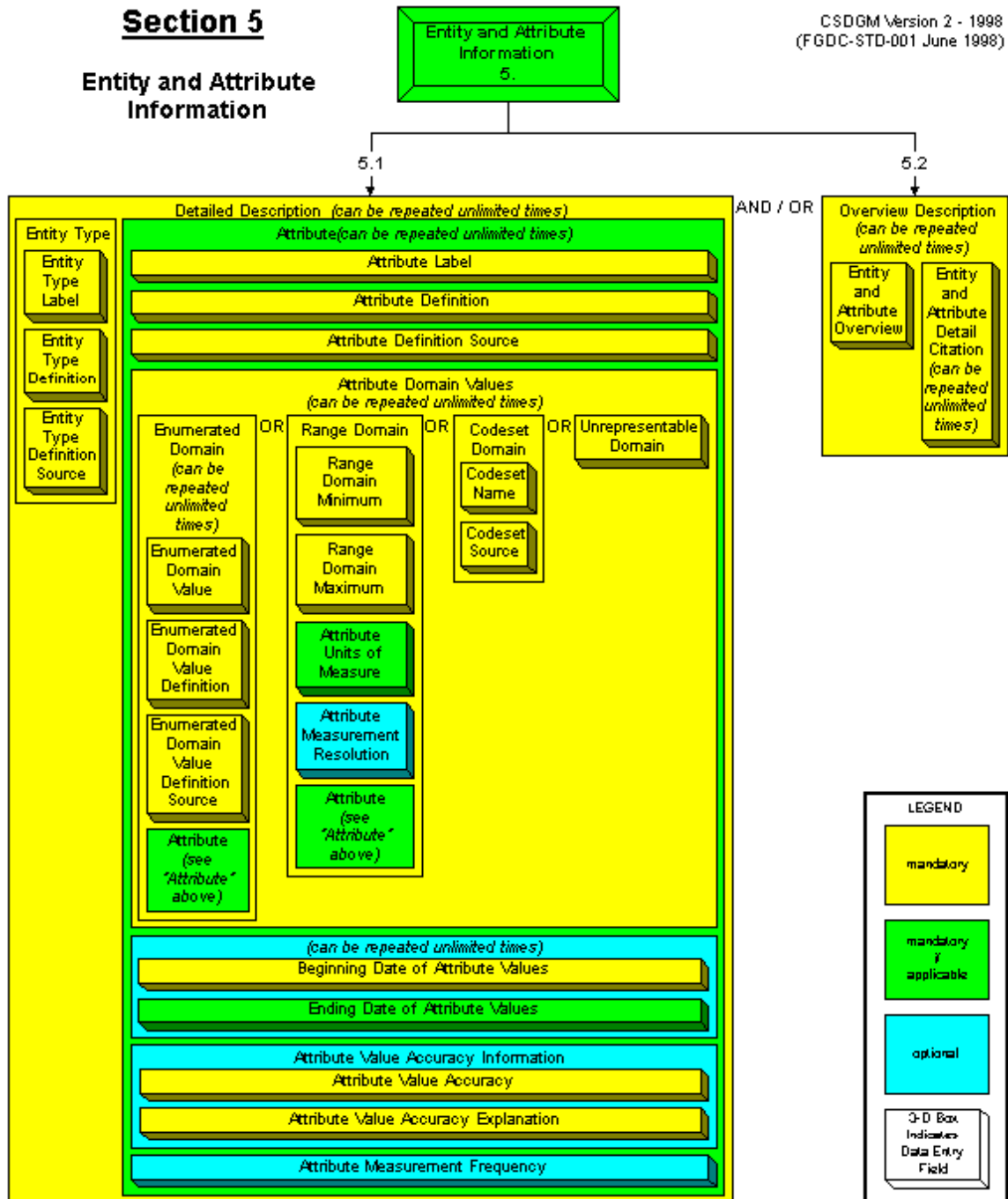
16.4.4. Section 3: Spatial data organization information



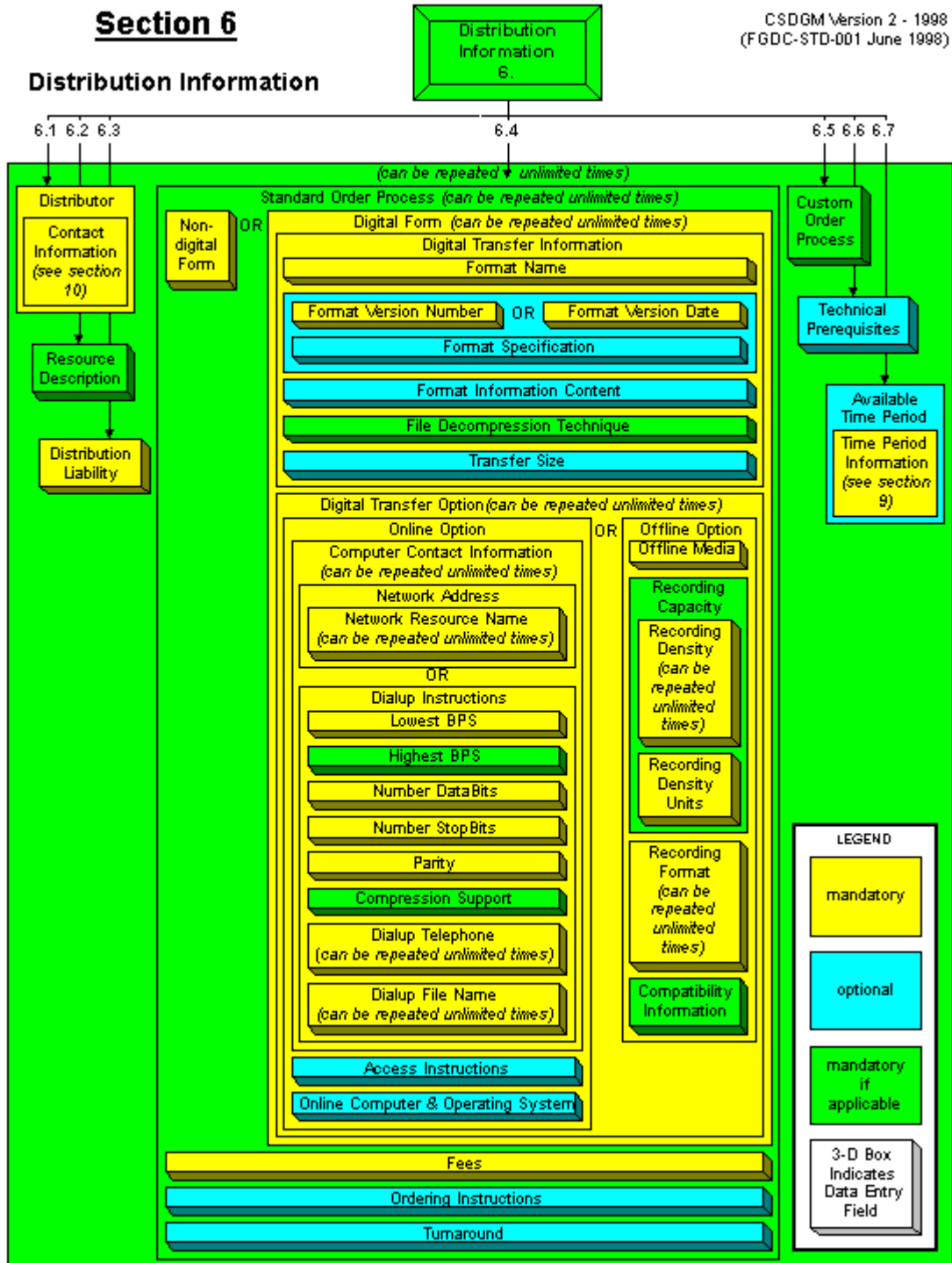
16.4.5. Section 4: Spatial reference information



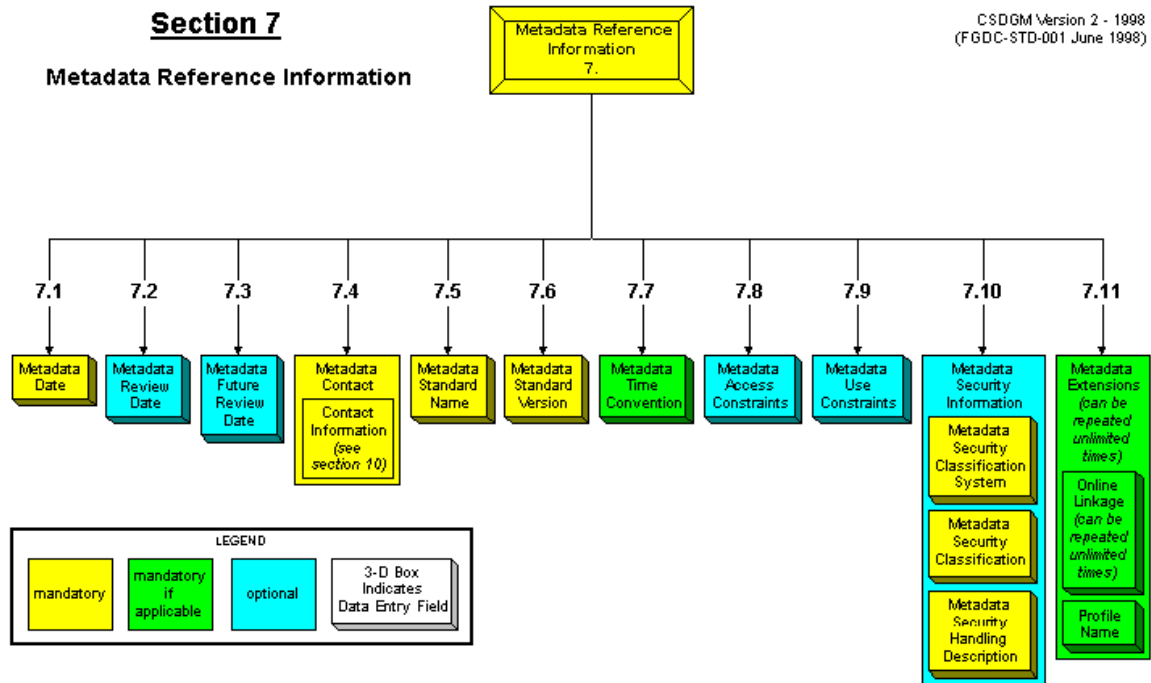
16.4.6. Section 5: Entity and attribute information



16.4.7. Section 6: Distribution information



16.4.8. Section 7: Metadata reference information



16.4.9. Sections 8, 9 and 10: Citation information, time period information, and contact information

CSDGM Version 2 - 1998 (FGDC-STD-001 June 1998)

Section 8

Citation Information

Citation Information
Originator <i>(can be repeated unlimited times)</i>
Publication Date
Publication Time
Title
Edition
Geospatial Data Presentation Form
Series Information
Series Name
Issue Identification
Publication Information
Publication Place
Publisher
Other Citation Details
Online Linkage <i>(can be repeated unlimited times)</i>
Larger Work Citation
Citation Information <i>(see Section 8)</i>

Section 9

Time Period Information

Time Period Information
Single Date / Time
Calendar Date
Time of Day
OR
Multiple Dates / Times
Single Date / Time <i>(2 or more repetitions)</i>
Calendar Date
Time of Day
OR
Range of Dates / Times
Beginning Date
Beginning Time
Ending Date
Ending Time

Section 10

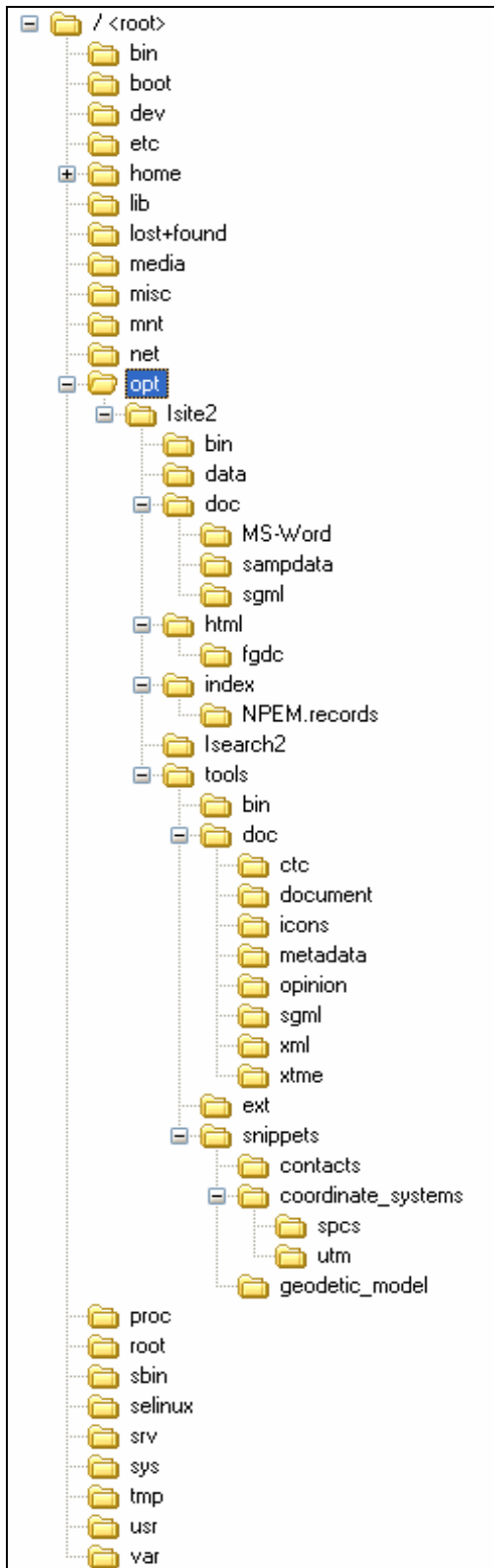
Contact Information

Contact Information
Contact Person Primary
Contact Person
Contact Organization
OR
Contact Organization Primary
Contact Organization
Contact Person
Contact Position
Contact Address <i>(can be repeated unlimited times)</i>
Address Type
Address <i>(can be repeated unlimited times)</i>
City
State or Province
Postal Code
Country
Contact Voice Telephone <i>(can be repeated unlimited times)</i>
Contact TDD/TTY Telephone <i>(can be repeated unlimited times)</i>
Contact Facsimile Telephone <i>(can be repeated unlimited times)</i>
Contact Electronic Mail Address <i>(can be repeated unlimited times)</i>
Hours of Service
Contact Instructions

LEGEND

mandatory	mandatory if applicable	optional	3-D Box Indicates Data Entry Field
-----------	-------------------------	----------	------------------------------------

16.5. Images of the Isite server directory structure and the files contained in each subdirectory after Isite installation

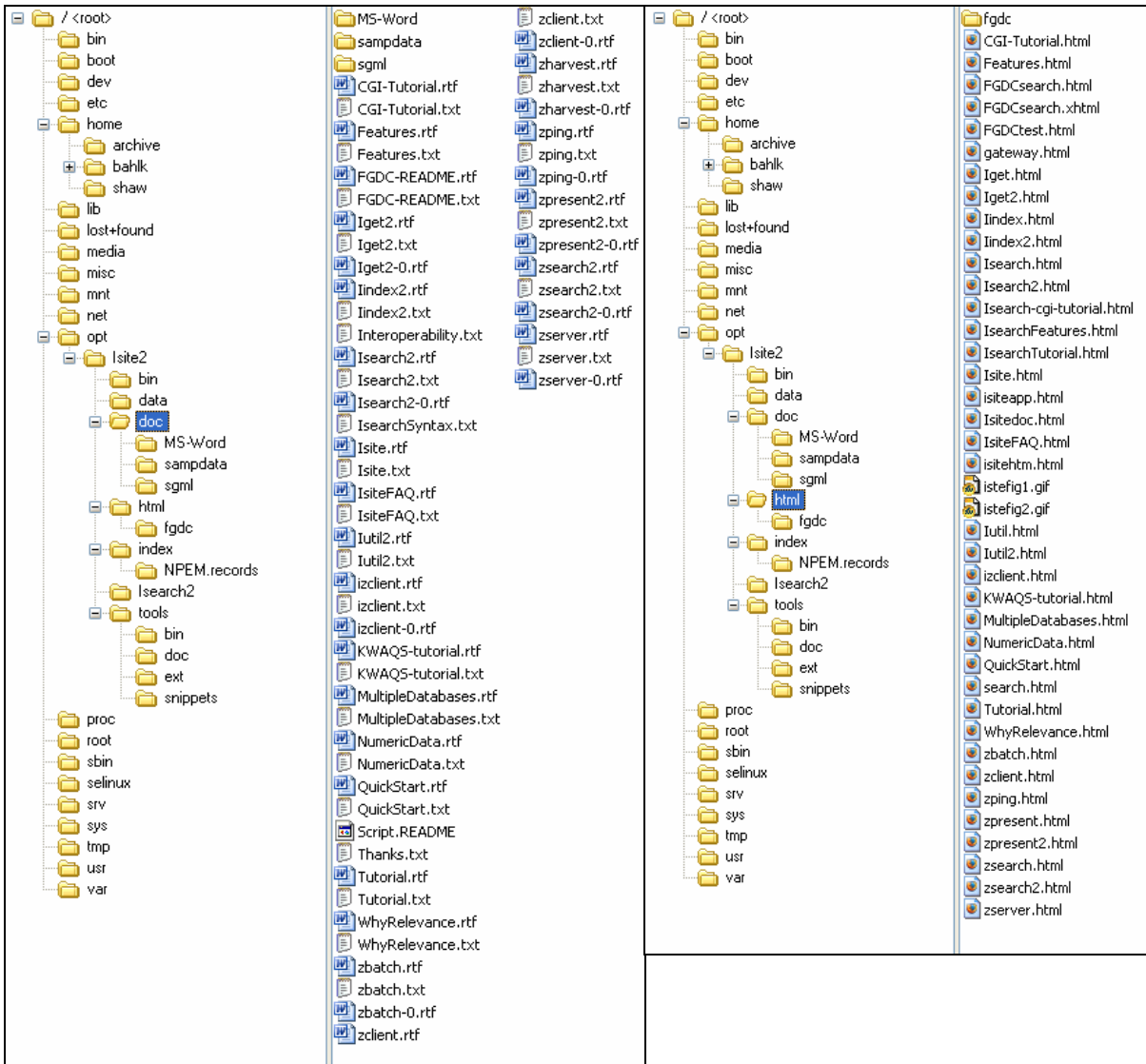


File explorer view of a Linux system root directory. The left pane shows the directory tree with 'bin' selected under 'Isearch2'. The right pane lists files and folders including:

- bin
- bib1.map
- bib1_fgdc.map
- bib1_iso19115.map
- build_test_db.sh
- build-testindex
- fgdc.fields
- geo.map
- geo_fgdc.map
- geo_iso19115.map
- GetScript
- gils.map
- gils_fgdc.map
- gils_iso19115.map
- Iget2
- Iindex2
- Isearch2
- IsearchTest
- IsearchTest-cygwin
- Isearch-uuid
- Iutil2
- izclient
- Key1
- sapi.ini
- sapi.ini-dist
- SearchScript
- zbatch
- zbatch.pl
- zclient
- zharvest
- zharvest.in
- zping
- zpresent2
- zsearch2
- zserver
- zserver.ini
- zserver.ini-dist
- zserver.pid
- zserver_access.log
- zserver-test.ini
- ztest
- ztest-testdb.in

File explorer view of a Linux system root directory. The left pane shows the directory tree with 'data' selected under 'Isearch2'. The right pane lists files and folders including:

- bin
- cgia-wswtemp.html
- cgia-wswtemp.shtml
- cgia-wswtemp.txt
- cgia-wswtemp.xml
- dds10.html
- dds10.shtml
- dds10.txt
- dds10.xml
- demomarc
- dust.html
- dust.shtml
- dust.txt
- dust.xml
- glaciers.html
- glaciers.shtml
- glaciers.txt
- glaciers.xml
- goes_9_conus.html
- goes_9_conus.shtml
- goes_9_conus.txt
- goes_9_conus.xml
- npem0.xml
- npem1.xml
- npem10.xml
- npem100.xml
- npem1000.xml
- npem1001.xml
- npem1002.xml
- npem1003.xml
- npem1004.xml
- npem1005.xml
- npem1006.xml
- npem1007.xml
- npem1008.xml
- npem1009.xml
- npem101.xml
- npem1010.xml
- npem1011.xml
- npem1012.xml
- npem1013.xml
- npem1014.xml
- npem1015.xml
- npem1016.xml
- npem1017.xml
- npem1018.xml
- npem1019.xml
- npem102.xml
- npem1020.xml
- npem1021.xml
- npem1022.xml
- npem1023.xml
- npem1024.xml
- npem1025.xml
- npem1026.xml
- npem1027.xml
- npem1028.xml
- npem1029.xml
- npem103.xml
- npem1030.xml
- npem1031.xml
- npem1032.xml
- npem1033.xml
- npem1034.xml
- npem1035.xml
- npem1036.xml
- npem1037.xml
- npem1038.xml
- npem1039.xml
- npem104.xml
- npem1040.xml
- npem1041.xml
- npem1042.xml
- npem1043.xml
- npem1044.xml
- npem1045.xml
- npem1046.xml
- npem1047.xml
- npem1048.xml
- npem1049.xml
- npem105.xml
- npem1051.xml
- npem1052.xml
- npem1053.xml
- npem1054.xml
- npem1055.xml
- npem1056.xml
- npem1057.xml
- npem1058.xml
- npem1059.xml
- npem106.xml
- npem1060.xml



The image displays three panels related to a file system structure:

- Left Panel:** A directory tree view of the root directory. The path `opt/Isite2/html/index` is highlighted in blue.
- Middle Panel:** A list of files, starting with `NPEM.records` and ending with `NPEM.060`. The files are arranged in three columns. Some files have a yellow icon, while others have a blue icon.
- Right Panel:** A directory tree view of the root directory. The path `opt/Isite2/html/index/lsearch2` is highlighted in blue. The right edge of this panel shows the start of a window titled "changes".

16.6. Listing of NPEM's Isite configuration files

[A copy of these files (zserver.ini and sapi.ini) can be found at the PICES web site.]

16.6.1. *zserver.ini*

```
# Example zserver Configuration File for the NPEM database
#
# Please read /opt/Isite2/doc/isite.ps or go to
# http://vinca.cnidr.org/software/Isite/Isite.html for more
# information on this file.
#
# lines beginning with a "#" are comments

# This is the Server Information Group

[Default]

#
# What detail of debugging information would you like printed to the
# screen?
#
# Valid values are 0-9 ranging from less to more debugging information.
# Do not set this to 9 unless you really want to see the gory details
# of the actual protocol encoding of requests and responses.
#

DebugLevel=9

#
# Do you want to run the server as a forking daemon or a single
# connection server? Note that INETD doesn't mean you have to run
# under inetd, just that the server will accept a single client
# connection and then exit. This is probably undesirable for production
# purposes, because of the startup time involved.
#
# STANDALONE or INETD
#

ServerType=STANDALONE

#
# What TCP port number for the server? 210 is the IETF-approved
# standard port, but requires system-level privileges to use. We used
# port 6668
#

Port=6668

#
# What is the maximum number of simultaneous client connections you'll
# accept?
#

MaxSessions=50
```

```

#
# Where should zserver record its process ID number when it starts?
#

PidFile=zserver.pid

#
# What is the FULL FILENAME of your Search API configuration file?
#
# For example, "SAPI=/usr/local/bin/sapi.ini"
#

SAPI=sapi.ini

#
# Access (transfer) log filename
#

AccessLog=zserver_access.log

#
# Server activity timeout
#

TimeOut=900

#
# List of databases available in your sapi.ini file, which ones
# do you want clients to have access to (comma-delimited, no spaces)?
#

DBList=NPEM

#
# STOP!!
#
# That should be all you have to edit in this file.
# Make sure you have your
# sapi.ini file configured properly and your own you way!
#

# This is a "special" diagnostic group.  If you want the error messages
# changed, do so here.
[1.2.840.10003.3.1]
1=Permanent system error
2=Temporary system error
3=Unsupported search
4=Terms only exclusion (stop) words
5=Too many argument words
6=Too many boolean operators
7=Too many truncated words
8=Too many incomplete subfields
9=Truncated words too short
10=Invalid format for record number (search term)
11=Too many characters in search statement
12=Too many records retrieved
13=Present request out of range

```

14=System error in presenting records
15=Record no authorized to be sent intersystem
16=Record exceeds Preferred-message-size
17=Record exceeds Maximum-record-size
18=Result set not supported as a search term
19=Only single result set as search term supported
20=Only ANDing of a single result set as search term supported
21=Result set exists and replace indicator off
22=Result set naming not supported
23=Combination of specified databases not supported
24=Element set names not supported
25=Specified element set name not valid for specified database
26=Only a single element set name supported
27=Result set no longer exists - unilaterally deleted by target
28=Result set is in use
29=One of the specified databases is locked
30=Specified result set does not exist
31=Resources exhausted - no results available
32=Resources exhausted - unpredictable partial results available
33=Resources exhausted - valid subset of results available
100=Unspecified error
101=Access-control failure
102=Security challenge required but could not be issued - request terminated
103=Security challenge required but could not be issued - record not included
104=Security challenge failed - record not included
105=Terminated by negative continue response
106=No abstract syntaxes agreed to for this record
107=Query type not supported
108=Malformed query
109=Database unavailable
110=Operator unsupported
111=Too many databases specified
112=Too many result sets created
113=Unsupported attribute type
114=Unsupported Use attribute
115=Unsupported value for Use attribute
116=Use attribute required but not supplied
117=Unsupported Relation attribute
118=Unsupported Structure attribute
119=Unsupported Position attribute
120=Unsupported Truncation attribute
121=Unsupported Attribute Set
122=Unsupported Completeness attribute
123=Unsupported attribute combination
124=Unsupported coded value for term
125=Malformed search term
126=Illegal term value for attribute
127=Unparsable format for un-normalized value
128=Illegal result set name
129=Proximity search of sets not supported
130=Illegal result set in proximity search
131=Unsupported proximity relation
132=Unsupported proximity unit code

16.6.2. *sapi.ini*

```
[DEFAULT]
```

```
DBList=NPEM
```

```
[NPEM]
```

```
type=ISEARCH
```

```
location=/opt/Isite2/index
```

```
FieldMaps=/opt/Isite2/bin/bib1_fgdc.map,\  
  /opt/Isite2/bin/gils_fgdc.map,\  
  /opt/Isite2/bin/geo_fgdc.map
```

16.7. Java program to extract records from the NPEM metadata database and write one XML file for each record.

[A copy of this file (FGDCEXtract.java) can be found at the PICES web site.]

```
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.net.InetAddress;
import java.sql.*;
import java.util.*;

import gov.noaa.foci.mdb.MDBSearchCommandHelper;

/**
 * Extracts FGDC data elements from North Pacific Metadata database
 * Creation date: (08/21/2004)
 * Modified date: (11/07/2004)
 * @author: dank
 */
public class FGDCEXtract {

    // local member variables
    private Vector title = new Vector();
    private Vector abstract = new Vector();
    private Vector purpose = new Vector();
    private Vector originator = new Vector();
    private Vector pubdate = new Vector();
    private Vector tpcdate = new Vector();
    private Vector tpcmdt = new Vector();
    private Vector tpcrdtbegdate = new Vector();
    private Vector tpcrdtenddate = new Vector();
    private Vector currref = new Vector();
    private Vector statprog = new Vector();
    private Vector statmuf = new Vector();
    private Vector sdnbc = new Vector();
    private Vector sdwbc = new Vector();
    private Vector sdebc = new Vector();
    private Vector sdsbc = new Vector();
    private Vector accessconstraint = new Vector();
    private Vector useconstraint = new Vector();
    private Vector sname = new Vector();
    private Vector sat = new Vector();
    private Vector sadd = new Vector();
    private Vector scity = new Vector();
    private Vector ssop = new Vector();
    private Vector szip = new Vector();
    private Vector scountry = new Vector();
    private Vector sphone = new Vector();
    private Vector sfax = new Vector();
    private Vector sem = new Vector();
    private Vector fgdcdate = new Vector();
    private Vector entrypersonname = new Vector();
    private Vector mdbid = new Vector();

    private Vector keyword = new Vector();
    private Vector thesaurus = new Vector();
```



```

private String dataDir = "";

// total records in the BSEMetaDatabase table
private int tablesize = 0;

// SQL members
private ResultSet result;
private Connection conn = null;

// Helper class
private MDBSearchCommandHelper helper = new MDBSearchCommandHelper();

// Query strings
private String query1 = "SELECT Title, Abstract, Purpose, " +
"Originator, PubDate, TPCDate, TPCMDT, TPCRDTBegDate, " +
"TPCRDTEndDate, CurRef, StatProg, StatMUF, SDNBC, SDWBC, SDEBC, SDSBC,
" +
"AccConstraint, UseConstraint, SName, SAT, SAdd, SCity, SSOP, " +
"SZIP, SCountry, SPhone, SFax, SEM, FGDCDate, EntryPersonName, MdbId FROM
bsemetadata";

private String query2 = "SELECT Thesaurus, Keyword FROM keywords WHERE MdbID
= ";
private String query3 = "";

private String distLiab = " While every effort has been made to ensure that
these data " +
"are accurate and reliable within the limits of the current state of the art,
NOAA cannot " +
"assume liability for any damages caused by any errors or omissions in the
data, nor as a " +
"result of the failure of the data to function on a particular system. NOAA
makes no warranty, " +
"expressed or implied, nor does the fact of distribution constitute such a
warranty.";

/**
 * execute
 * This is the work horse method for the command.
 * It will execute the queries and get the result set.
 */
public void execute()
throws
java.lang.Exception,
java.io.IOException,
java.sql.SQLException {

    try {
        Counter counter = new Counter();
        tablesize = counter.countTotalRecords(conn);
        Searcher searcher = new Searcher();
        searcher.searchBSEMetaDataTable(conn, query1);
        for (int i=0; i<tablesize; i++) {
            thesaurus.clear();
            keyword.clear();
query3 = query2 + (String) mdbid.elementAt(i) + " ORDER BY Thesaurus ASC";

```

```

        searcher.searchKeywordTable(conn, query3);
        writeFGDCXML(i);
    }
} catch (Throwable theException) {
    theException.printStackTrace();
} finally {
    conn.close();
}
}

public class Counter {
private static final String COUNT_SQL = "SELECT Count(*) AS trecs FROM
bsemetadata";
    public int countTotalRecords(Connection con) {
        int size = 0;
        try {
            Statement statement = conn.createStatement();
            ResultSet rs = statement.executeQuery(COUNT_SQL);
            while (rs.next()) {
                size = rs.getInt("trecs");
            }
        } catch (Throwable theException) {
            theException.printStackTrace();
        }
        return size;
    }
}

public class Searcher {
    public void searchBSEMetaDataTable(Connection con, String query) {
        try {
            Statement statement = conn.createStatement();
            ResultSet rs = statement.executeQuery(query);
            while (rs.next()) {
                title.addElement(rs.getString(1));
                abstrct.addElement(rs.getString(2));
                purpose.addElement(rs.getString(3));
                originator.addElement(rs.getString(4));
                pubdate.addElement(rs.getString(5));
                tpcdate.addElement(rs.getString(6));
                tpcmdt.addElement(rs.getString(7));
                tpcrdtbegdate.addElement(rs.getString(8));
                tpcrdtenddate.addElement(rs.getString(9));
                currref.addElement(rs.getString(10));
                statprog.addElement(rs.getString(11));
                statmuf.addElement(rs.getString(12));
                sdnbc.addElement(rs.getString(13));
                sdwbc.addElement(rs.getString(14));
                sdebc.addElement(rs.getString(15));
                sdsbc.addElement(rs.getString(16));

accessconstraint.addElement(rs.getString(17));
                useconstraint.addElement(rs.getString(18));
                sname.addElement(rs.getString(19));
                sat.addElement(rs.getString(20));
                sadd.addElement(rs.getString(21));
                scity.addElement(rs.getString(22));
            }
        }
    }
}

```

```

        ssop.addElement(rs.getString(23));
        szip.addElement(rs.getString(24));
        scountry.addElement(rs.getString(25));
        sphone.addElement(rs.getString(26));
        sfax.addElement(rs.getString(27));
        sem.addElement(rs.getString(28));
        fgdcdate.addElement(rs.getString(29));
        entriypersonname.addElement(rs.getString(30));
        mdbid.addElement(rs.getString(31));
    }
} catch (Throwable theException) {
    theException.printStackTrace();
}
}

public void searchKeywordTable(Connection con, String query) {
try {
    Statement statement = conn.createStatement();
    ResultSet rs = statement.executeQuery(query);
    while (rs.next()) {
        thesaurus.addElement(rs.getString(1));
        keyword.addElement(rs.getString(2));
    }
} catch (Throwable theException) {
    theException.printStackTrace();
}
}

private String dataDir() {
String dir = "";
try {
    InetAddress localaddr = InetAddress.getLocalHost();
if (localaddr.getHostName().equals("afscmaps.akctr.noaa.gov") ||
localaddr.getHostName().equals("afscmaps.afsc.noaa.gov")) { // production
    dir="/usr/local/Isite2/data/";
    } else {
if (localaddr.getHostName().equals("gandalf")) { // Dan's RH9.0 box
    dir = "/usr/local/Isite2/data/";
    } else {
        if (localaddr.getHostName().equals("pippin")) { //
Dan's Win2K box
            dir = "c:\\fgdc\\";
        } else {
System.out.println("Could not set the directory from the hostname");
        }
    }
} catch (Throwable t) {
    t.printStackTrace();
}
return dir;
}

private void writeFGDCXML(int index) {
    dataDir = dataDir();
    try {

```

```

FileOutputStream fout = new FileOutputStream(dataDir + "npem" + index +
".xml");
    OutputStreamWriter out = new OutputStreamWriter(fout, "UTF-8");
    out.write("<?xml version=\"1.0\"?>\r\n");
    out.write("<metadata>\r\n");
    out.write("    <idinfo>\r\n");
    out.write("        <citation>\r\n");
    out.write("            <citeinfo>\r\n");
        out.write("                <origin>" + originator.elementAt(index) +
"</origin>\r\n");
        out.write("                    <pubdate>" + pubdate.elementAt(index) +
"</pubdate>\r\n");
        out.write("                        <title>" + title.elementAt(index) +
"</title>\r\n");
        out.write("                            </citeinfo>\r\n");
        out.write("                                </citation>\r\n");
        out.write("                                    <descript>\r\n");
        out.write("                                        <abstract>" + abstrct.elementAt(index) +
"</abstract>\r\n");
        out.write("                                            <purpose>" + purpose.elementAt(index) +
"</purpose>\r\n");
        out.write("                                                </descript>\r\n");
        out.write("                                                    <timeperd>\r\n");
        out.write("                                                        <timeinfo>\r\n");
if (!(tpcdate.elementAt(index) == null ||
tpcdate.elementAt(index).equals("")))) {
            out.write("                    <sngdate>\r\n");
            out.write("                        <caldate>" + tpcdate.elementAt(index) +
"</caldate>\r\n");
            out.write("                            </sngdate>\r\n");
} else if (!(tpcmdt.elementAt(index) == null ||
tpcmdt.elementAt(index).equals("")))) {
            out.write("                <mdattim>\r\n");
            out.write("                    <sngdate>\r\n");
            out.write("                        <caldate>" + tpcmdt.elementAt(index) +
"</caldate>\r\n");
            out.write("                            </sngdate>\r\n");
            out.write("                                </mdattim>\r\n");
        } else if (!(tpcrdtbegdate.elementAt(index) == null ||
tpcrdtbegdate.elementAt(index).equals("")))) {
            out.write("                    <rngdates>\r\n");
            out.write("                        <begdate>" + tpcrdtbegdate.elementAt(index)
+
"</begdate>\r\n");
            out.write("                            <enddate>" + tpcrdtenddate.elementAt(index)
+
"</enddate>\r\n");
            out.write("                                </rngdates>\r\n");
        }
        out.write("                </timeinfo>\r\n");
        out.write("                    <current>" + currref.elementAt(index) +
"</current>\r\n");
        out.write("                        </timeperd>\r\n");
        out.write("                            <status>\r\n");
        out.write("                                <progress>" + statprog.elementAt(index) +
"</progress>\r\n");

```

```

        out.write("        <update>" + statmuf.elementAt(index) +
"</update>\r\n");
        out.write("        </status>\r\n");
        out.write("        <spdom>\r\n");
        out.write("        <bounding>\r\n");
        out.write("        <westbc>" + sdwbc.elementAt(index) +
"</westbc>\r\n");
        out.write("        <eastbc>" + sdebc.elementAt(index) +
"</eastbc>\r\n");
        out.write("        <northbc>" + sdnbc.elementAt(index) +
"</northbc>\r\n");
        out.write("        <southbc>" + sdsbc.elementAt(index) +
"</southbc>\r\n");
        out.write("        </bounding>\r\n");
        out.write("        </spdom>\r\n");
        if (!thesaurus.isEmpty()) {
            out.write("        <keywords>\r\n");
            String lastThesaurus = "";
            for (int i = 0; i < thesaurus.size(); i++) {
                if (!((String)thesaurus.elementAt(i)).equals(lastThesaurus)) {
                    if (!lastThesaurus.equals("")) out.write("</theme>\r\n");
                    out.write("        <theme>\r\n");
                    out.write("        <themekt>" +
thesaurus.elementAt(i) + "</themekt>\r\n");
                }
                out.write("        <themekey>" + keyword.elementAt(i) +
"</themekey>\r\n");
                lastThesaurus = (String) thesaurus.elementAt(i);
            }
            out.write("        </theme>\r\n");
            out.write("        </keywords>\r\n");
        }
        out.write("        <accconst>" + accessconstraint.elementAt(index) +
"</accconst>\r\n");
        out.write("        <useconst>" + useconstraint.elementAt(index) +
"</useconst>\r\n");
        out.write("        <ptcontac>\r\n");
        out.write("        <cntinfo>\r\n");
        out.write("        <cntperp>\r\n");
        out.write("        <cntper>" + sname.elementAt(index) +
"</cntper>\r\n");
        out.write("        </cntperp>\r\n");
        out.write("        <cntaddr>\r\n");
        out.write("        <addrtype>" + sat.elementAt(index) +
"</addrtype>\r\n");
        out.write("        <address>" + sadd.elementAt(index) +
"</address>\r\n");
        out.write("        <city>" + scity.elementAt(index) +
"</city>\r\n");
        out.write("        <state>" + ssop.elementAt(index) +
"</state>\r\n");
        out.write("        <postal>" + szip.elementAt(index) +
"</postal>\r\n");
        out.write("        <country>" + scountry.elementAt(index) +
"</country>\r\n");
        out.write("        </cntaddr>\r\n");
        out.write("        <cntvoice>" + sphone.elementAt(index) +

```

```

    "</cntvoice>\r\n");
        out.write("                <cntfax>" + sfax.elementAt(index) +
"</cntfax>\r\n");
        out.write("                <cntemail>" + sem.elementAt(index) +
"</cntemail>\r\n");
        out.write("                </cntinfo>\r\n");
        out.write("                </ptcontac>\r\n");
        out.write("        </idinfo>\r\n");
        out.write("        <distinfo>\r\n");
        out.write("        <distrib>\r\n");
        out.write("        <cntinfo>\r\n");
        out.write("        <cntperp>\r\n");
        out.write("        <cntper>" + sname.elementAt(index) +
"</cntper>\r\n");
        out.write("        </cntperp>\r\n");
        out.write("        <cntaddr>\r\n");
        out.write("        <addrtype>" + sat.elementAt(index) +
"</addrtype>\r\n");
        out.write("        <address>" + sadd.elementAt(index) +
"</address>\r\n");
        out.write("        <city>" + scity.elementAt(index) +
"</city>\r\n");
        out.write("        <state>" + ssop.elementAt(index) +
"</state>\r\n");
        out.write("        <postal>" + szip.elementAt(index) +
"</postal>\r\n");
        out.write("        <country>" + scountry.elementAt(index) +
"</country>\r\n");
        out.write("        </cntaddr>\r\n");
        out.write("        <cntvoice>" + sphone.elementAt(index) +
"</cntvoice>\r\n");
        out.write("        <cntfax>" + sfax.elementAt(index) +
"</cntfax>\r\n");
        out.write("        <cntemail>" + sem.elementAt(index) +
"</cntemail>\r\n");
        out.write("        </cntinfo>\r\n");
        out.write("        </distrib>\r\n");
        out.write("        <distliab>" + distLiab + "</distliab>\r\n");
        out.write("        </distinfo>\r\n");
        out.write("        <metainfo>\r\n");
        out.write("        <metd>" + fgdcdate.elementAt(index) + "</metd>\r\n");
        out.write("        <metc>\r\n");
        out.write("        <cntinfo>\r\n");
        out.write("        <cntperp>\r\n");
        out.write("        <cntper>Kimberly Bahl</cntper>\r\n");
        out.write("        </cntperp>\r\n");
        out.write("        <cntaddr>\r\n");
        out.write("        <addrtype>Mailing and Physical
Address</addrtype>\r\n");
        out.write("        <address> NOAA / AFSC / PMEL, 7600 Sand Point
Way
N.E.</address>\r\n");
        out.write("        <city>Seattle</city>\r\n");
        out.write("        <state>WA</state>\r\n");
        out.write("        <postal>98115</postal>\r\n");
        out.write("        <country>USA</country>\r\n");
        out.write("        </cntaddr>\r\n");

```

```

        out.write("          <cntvoice>206-526-4314</cntvoice>\r\n");
        out.write("          <cntfax>206-526-6485</cntfax>\r\n");
        out.write("
<cntemail>kimberly.bahl@noaa.gov</cntemail>\r\n");
        out.write("          </cntinfo>\r\n");
        out.write("          </metc>\r\n");
        out.write("          <metstdn>Content Standards for Digital Geospatial
Metadata</metstdn>\r\n");
        out.write("          <metstdv>FGDC-STD-001-1998</metstdv>\r\n");
        out.write("          <metac>" + accessconstraint.elementAt(index) +
"</metac>\r\n");
        out.write("          <metuc>" + useconstraint.elementAt(index) +
"</metuc>\r\n");
        out.write("    </metainfo>\r\n");
        out.write("</metadata>\r\n");
        out.close();
    } catch (IOException e) {
        System.err.println(e);
    }
}

/**
 * initialize
 * This method will connect to the database.
 */
public void initialize() throws java.io.IOException {
    conn = helper.connectToDB();
}

public int getTableSize() {
    return tablesize;
}

public String getDataDir() {
    return dataDir;
}
}

```

16.8. Java program to execute the metadata extraction program

[A copy of this file (FGDCEExtractClient.java) can be found at the PICES web site.]

```
package gov.noaa.foci.mdb;

/**
 * @author Dan Klawitter
 * 11/07/2004
 *
 * Client for running FGDCEExtract from command line or cron job
 */
public class FGDCEExtractClient {

    public static void main(String[] args) {
        try {
            FGDCEExtract ext = new FGDCEExtract();
            System.out.println("Running FGDC Extract ...");
            ext.initialize();
            ext.execute();
            System.out.println(ext.getTableSize() + " Records Extracted to
" + ext.getDataDir());
        } catch (Throwable theException) {
            theException.printStackTrace();
        }
    }
}
```