

WEB REGISTRY SERVER DISCUSSION PAPER

INTRODUCTION

This is the first draft of a specification of a Web Registry Server. This document is based on the experience gained in the WMT Stateless Catalog prototype efforts and the discussions in the WMT2 telecons and the Catalog Revision Working Group. This is the first attempt to cast the stateless catalog as a first class service OWS and collect and formalize the experience gained in the populating the catalog. This specification has not been reviewed by the other team members and represents the editor's view of those activities. This is a draft and may change significantly based on reviews and the evolution of the Basic/General Services Model.

There many technical issues that need to be discussed and resolved including:

- Do others agree with this view of the evolution of the the Stateless catalogs into a Web Registry Service with service description registration and update as a formal interface and the use of GetCapabilities to substitute for the Explain capabilities in the OGC Catalog Specification
- Will the same service registry catalog WMS, WFS and WCS. If so how different are the metadata. For example is a Feature Type or a Feature Collection the correct granularity for a registry descriptor and how does it's metadata vary from that of a mapping layer
- If we decide that the Registry descriptors need an incremental update service for scalability, can service providers use the WFS transaction interface to specify the required updates
- The specification of the Registry descriptor contents and schema are equivalent/dependant on the same research as the Basic/General Service Model. How should the two efforts interact
- Is the full GetDescriptors service interface acceptable for lightweight clients or should we maintain a GetDescriptors profile with defaults similar to those of the Stateless Catalog prototypes

TITLE:	Web Registry Server Specification		
VERSION:	0.0.2		
EDITOR:	Name:	Lou Reich	
	Address:	Computer Sciences Corp	
		7700 Hubble Dr	
		Lanham-Seabrook MD 20706 USA	
	Phone:	+1 301 794-2195	
	FAX:	+1 301 794-8355	
	Email:	louis.i.reich@gsfc.nasa.gov	
DATE:	2001-1-26		
CATEGORY:	OGC-IP Draft Candidate Specification		

PREFACE

This document describes the OGC Web Registry Server (WRS) interfaces. The WRS interfaces support “one stop shopping” for the registration, metadata harvesting and descriptor ingest, push and pull update of descriptors, and discovery of OGC Web Service types and instances using HTTP as the distributed computing platform.

STATUS OF THIS DOCUMENT

This specification is a draft document submitted to OGC for review and may be updated, replaced or obsoleted by other documents at any time. Considering the early stages of this specification it is inappropriate to use it as reference material or cite this document as other than “work in progress”. The purpose of this document is to be the basis of an OGC Discussion paper as described in the OGC Policies and Procedures. This document is dependant on the OGC Basic Service Model for the definition of the general semantics, exception handling and the definition of the GetCapabilities operation.

Submitting Companies

The following companies submitted this implementation specification to the OGC as a Request For Comment:

Compusalt

SGT

Submission Contact Points

All questions regarding this submission should be directed to the editor or the submitters:

Editor

Lou Reich
Computer Sciences Corp
7700 Hubble Dr
Lanham-Seabrook MD 20706

Contributors

Larry Bourzran
Doug Nebert
Barry O'Rourke
George Percivall
Ananth Rao
Archie Warnock

Document Conventions

This document contains sections that are “informative”, meaning they serve as explanation and background. Other sections are “normative”, meaning they contain the formal specification against which conformance testing can be done. The normative sections are labeled as such.

In the sections labeled as normative, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [15].

Revision History

0.0.1 WMT2 development version, 2000 October 4

0.0.2 WMT2 development version, 2001 January 31

Changes To The OpenGIS® Abstract Specification

No changes to the Abstract Specification other than the additions noted in this document are required for this specification.

This specification should form the basis of a new Web Services Volume of the OGC Bookshelf.

OVERVIEW

Motivation

Phase 2 of the Web Mapping Testbed (WMT2) resulted in several candidate OGC interface specifications for operations on a Web Map Server (WMS), Web Coverage Server (WCS) or Web Feature Server (WFS). The specs all differ in their purposes and details, but a number of elements are common to most or all of them. There was a desire by service users to have a single entry point (or at most a few) and protocol to search for service types of interest. In order to meet this goal, there needs to be an interface that populates a database of service descriptions and an interface to query the description database to discover service locations. This document specifies a Registry Service framework that operates in a stateless HTTP environment. This allows service type discovery using a single catalog site rather than visiting each server and using GetCapabilities operations.

This specification is mapped to the OGC Catalog Service version 1 and is intended to be consistent with the OGC Catalog Service version 1.1. This is intended to allow Web Registries to interoperate with stateful catalogs implementing the OGC Catalog Service in service discovery.

Terminology (Editor's note must update)

A service is one of the types of functionality described in this document and its companion volumes.

A service instance is a particular host computer that offers a service.

An operation is a combination of a client's request that a service instance perform an action and a server's response to that request.

An interface is a collection of one or more operations by which a service interacts with the external environment. A service instance establishes a particular network connection point as its interface.

A protocol specifies the mechanism and format for submitting an operation request and receiving a response. Various different computing platforms may use different protocols for the same operations.

As an analogy, if we apply this terminology to the World Wide Web then access to documents and multimedia objects is a service, a web site is a service instance, HTTP is a protocol, and HTTP GET is an operation combining a request for an object and a response containing that object. A geodata object is a unit of geospatial informational content that can be requested from an OGC Web Service. For example, a feature collection, a map layer, and a coverage are all geodata objects.

Document Organization (Editor's note TBS)

BACKGROUND

Web Mapping Testbed phase 1 (WMT1) resulted in two specifications: "OpenGIS Web Map Server Interface Implementation Specification" version 1, and "OpenGIS Recommendation - Geography Markup Language" (GML) v.1. These documents were the result of a consensus development process by WMT1 sponsors and participants and were approved by the full OGC membership. GML is an XML-based encoding of geographic features, and is described elsewhere; being a file format, GML does not itself specify any operations. The WMS 1.0 specification was an important first step towards interoperable access to georeferenced information, but it does have a number of acknowledged limitations. The goal of Web Mapping Testbed phase 2, and a simultaneous Geospatial Fusion Services (GFS) testbed, was to advance the state of affairs by enhancing the WMS spec and creating new standards for Web Coverage Server, Web Feature Server, GeoParser, GeoCoder, GeoLinker, and others.

Web Map, Feature and Coverage Servers--collectively called OGC Web Services--are each the subject of detailed specifications. They are described conceptually in the next section as background material.

Architecture Overview

WMT 2 Overview

Three principal types of georeferenced information access services have been defined by WMT2: Web Map Server (WMS), Web Coverage Server (WCS), and Web Feature Server (WFS). In addition, the Geospatial Fusion Services (GFS) Testbed defined services that return spatially referenced results: GeoParser, GeoCoder, and GeoLinker. Collectively, such services are referred to as OGC Web Services (OWS). Figure 5-1 is an architecture diagram showing conceptually how these components are related, and naming some (not all) of the interfaces between them. The subject of this document and its companion volumes is the interfaces. The internal details of the components are irrelevant; vendors may build them in any manner that correctly implements the required interfaces. This version of the figure emphasises the interfaces/operations proposed for the Web Registry Server (WRS) and how they would fit into the OGC Web Services

architecture.

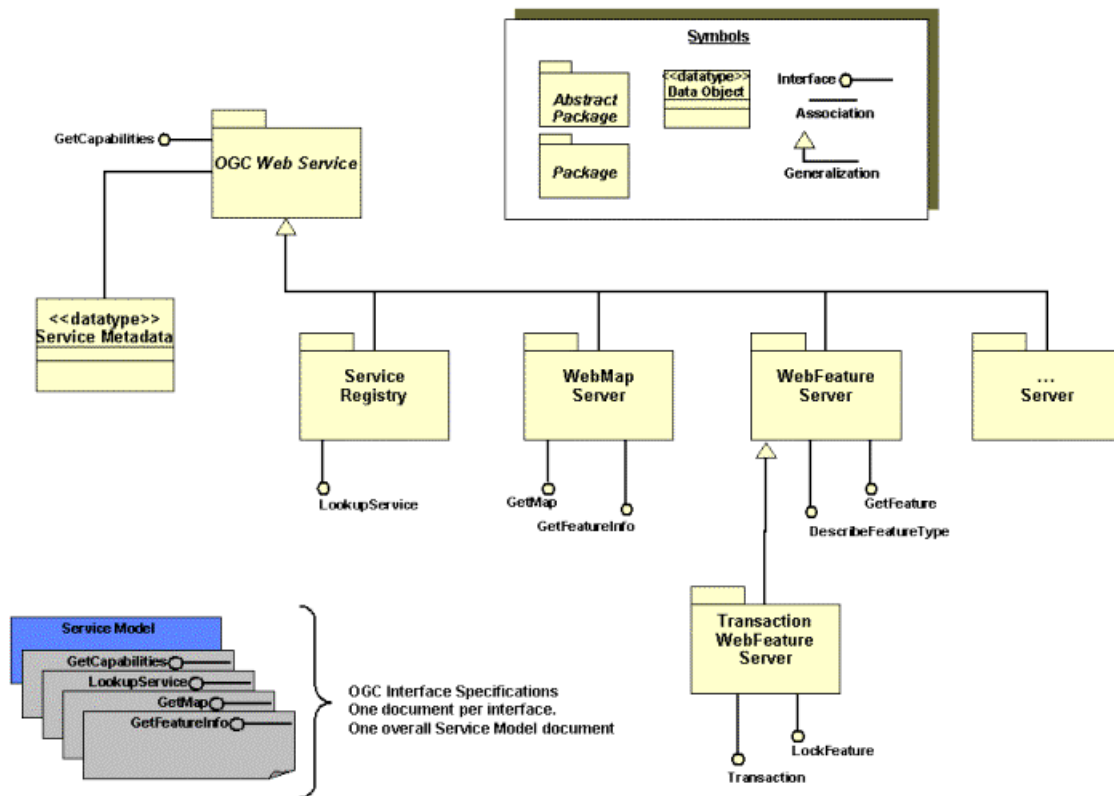


Figure 1:WMT2 Architectue emphasizing Web Registry Server

(EDITOR's note: Figure needs to be smoothed).

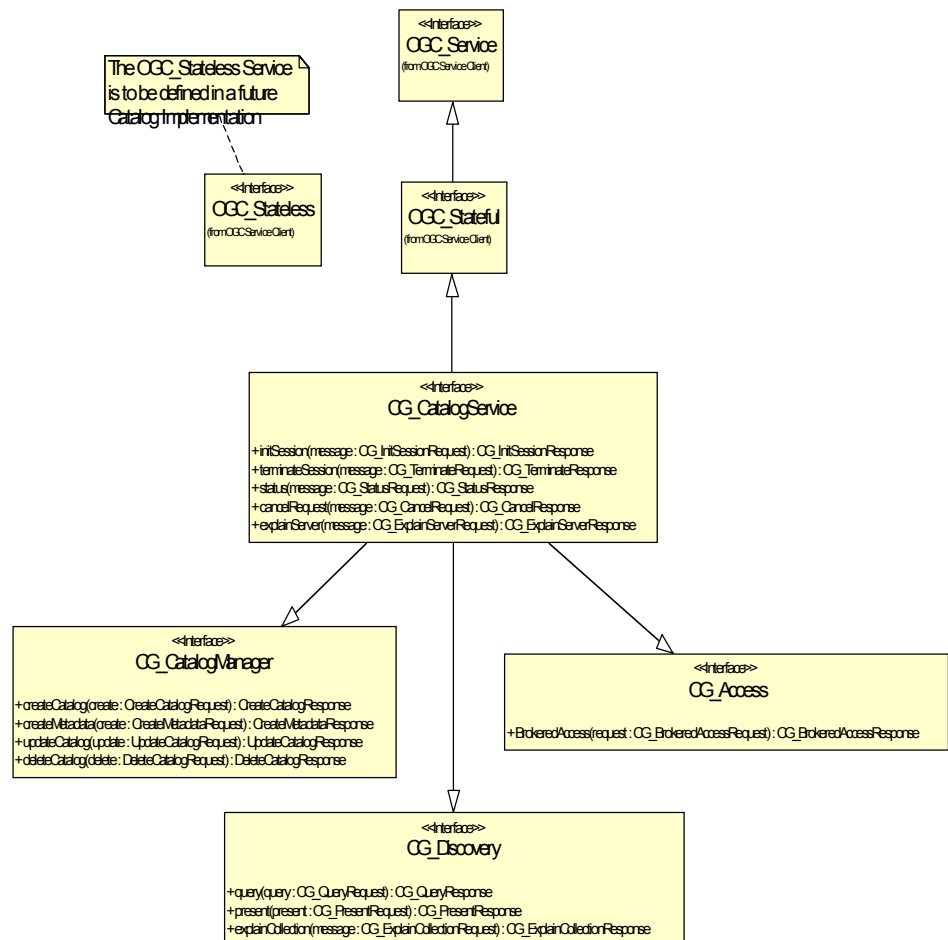
Each service has several required operations, but it is not required that the same software or internet host computer implement all of them. For example, a WMS may offer GetMap directly but allow its GetCapabilities response to be handled by another machine on its behalf. The service metadata in Capabilities XML includes distinct online resource addresses (e.g., HTTP URLs) for each service offered. There is no requirement in this specification dictating the format or content of those URLs; rather, this document describes only the packaging of the query parameters that must be appended to the service URL.

Overview of OGC Catalog Specification

Figure 2 shows the general service interfaces. These interfaces allow the discovery, access and management of geospatial data and services. This model is based on the concept of interface operations passing Request – Response Message Pairs between a client and a server. Stated another way, this architecture uses a messaging based structure to describe the access and invocation of Catalog services.

As seen in Figure 2 there are four major interfaces, CG_CatalogService, CG_Discovery, CG_Access and CG_CatalogManager. These are described in more detail in the following sections

of this document. The taxonomy of interfaces that have been placed above the CG_CatalogService interface (i.e., OGC_Service and OGC_Stateful) have been created to put forth the idea of having an overall architectural framework for the different services that will be developed over time to populate the OGC Service Architecture.



This specification defines stateless versions of the CG_CatalogService, the CG_Discovery and the CG_CatalogManager interfaces which are intended to support lightweight clients and servers that support HTTP version 1.0 type functionality and do not necessarily maintain information required to enable persistent sessions.

Interfaces

To support service discovery the WMT Registry Server must support the following interfaces:

<GetCapabilities>

A server must be able to describe its capabilities. Specifically, it must indicate which query languages, namespaces, return schema and return formats it supports. This service is similar to the

the non-session oriented portions of the CG_CatalogService Interface in the OGC Catalog Service Specification

<GetDescriptor>

A registry server must be able to respond to client queries in at least one of a number of predefined query languages and return the results of those queries in at least one of a number of predefined formats. This interface is similar to the CG_Discovery Interface in the OGC Catalog Service Specification

To support Open Server Registration and Periodic Content Harvesting the Web Registry Server may support the following interfaces:

<RegisterService>

Web Registry Server may allow service provider's to register their servers online, request periodic update thru the harvesting of capabilities documents from the server or the adhoc update of the descriptors data base by transaction or in bulk

Request Semantics

This service is intended to comply to chapters 4 and 5 of OGC WMT2 Basic Services Model. HTTP defines two methods of exchanging information with a web server; the GET method and the POST method.

GET Method

With the GET method, a request is encoded into the URI that is sent to the web server. The main limitations of GET are: The length of the URI has a hard limit (typically set by the browser being used) which limits the size of a web feature server request. It is difficult (if not impossible) for complicated MIME messages to be sent from the client to the server using the GET method.

POST Method

A POST request can be arbitrarily long and complex which means that client applications can generate long and complex query structures

The following table correlates WRS interfaces and request semantics:

Interface	Request Semantics
<GetDescriptors>	POST + limited GET
<RegisterService>	POST + GET
<GetCapabilities>	POST + GET

The main body of this document defines the interfaces in a form suitable for packaging as an HTTP POST request. The prototype implementations which are discussed in Annex A use the GET syntax.

<GETCAPABILITIES> INTERFACE

A WRS must be able to describe its capabilities. Specifically, it must indicate which interfaces, query languages, namespaces, return schema and return formats it supports. The Normative specification for the <GetCapabilities> Interface is given in the WMT Basic Services Model. This section provides the details of this interface for a WRS. This service is similar to the non-session oriented portions of the CG_CatalogService Interface in the OGC Catalog Service Specification.

. The normative portions of BSM 1.0 **must** be considered an integral and mandatory part of this WRS specification. The purpose of the GetCapabilities operation is specified in BSM 1.0 [17]. Refer to that document for additional mandatory information regarding GetCapabilities.

For the particular case of a Web Registry Server, the response of a GetCapabilities request is general information about the service itself and specific information about the available attribute namespaces, query languages and element sets.

GetCapabilities Request

Overview

The general form of a GetCapabilities request is defined in BSM 1.0 [17]. When making this request of a WRS, which may offer other OGC Web Services as well, it is necessary to indicate that the client seeks information about the WRS in particular. Thus, the SERVICE parameter of the request **must** have the value "WRS" as shown in the table below.

<i>URL Component</i>	Required/ Optional	Description
http://server_address/path/script?	R	URL prefix of server
VERSION=version	O	Request version
REQUEST=GetCapabilities	R	Request name
SERVICE=WRS	R	Service type

Table ?? - The GetCapabilities Request for a WMS

Request Parameters

Prefix

The role of the URL prefix is specified in BSM 1.0 [17]. The prefixes for GetCapabilities, GetMap and GetFeature info **may** be different.

VERSION

This parameter is specified in BSM 1.0 [17].

REQUEST

This nature of this parameter is specified in BSM 1.0 [17]. For GetCapabilities, the value "GetCapabilities" **must** be used.

SERVICE

*This nature of this parameter is specified in BSM 1.0 [17]. When invoking GetCapabilities on a WRS the value "WRS" **must** be used.*

GetCapabilities Response

(Editor's note Annex B is TBS)

BSM 1.0 [17] specifies general rules about the GetCapabilities response.

In the particular case of a Web Repository Server complying with version 1.0.0 of this standard, the Extensible Markup Language (XML)[9] response **must** be valid according to the XML Schema in Appendix B of this document. The Schema specifies the required and optional content of the response and how the content is formatted.

< GETDESCRIPTORS > INTERFACE

Request

The GetDescriptors request document is described by the following DTD. Parameters which require enumerations or defaults are expressed using attributes because of DTD limitations. This DTD is appropriate for use with an HTTP Post request. Annex A.1 shows the HTTP Get form of this request used by the WMT2 Prototype implementations

```
<!ELEMENT getDescriptors (querySpec, presentationSpec) >
<!ATTLIST OGC_Web_Service
    version CDATA #FIXED "0.0.2"
    updateSequence CDATA "0">

<!-- The querySpec provides metadata for completely specifying the query
requested. -->
<!ELEMENT querySpec (queryExpression, attributeSetName)>
<!ATTLIST querySpec
    queryLanguage CDATA ("OGC_Common"|"OGC_Filter"|"SFSQL") #REQUIRED
    queryScope CDATA ("local"|"cascade") #DEFAULT="local"
    catalogType CDATA ("product"|"collection"|"service")
        DEFAULT="service">
<!--if queryLanguage is OGC_Filter the queryExpression contains XML Markup--!>
<!ELEMENT queryExpression (#PCDATA|ANY) >
<!ELEMENT attributeSetName (#PCDATA) >
<!-- The presentationSpec provides metadata for completely specifying the
returndocument -->
<!--as the GetDescriptors response. -->
<!ELEMENT presentationSpec (elementSchema,resultPosition)>
<!ATTLIST presentationSpec
    maxRecs CDATA #DEFAULT "0">
<!-- The element schema must be one of those specified in the getCapabilities
response. This will be easier in XML/Schema -->
<!ELEMENT elementSchema (#PCDATA)
<!ATTLIST elementSchema
    elementSetName ("Brief"|"Summary"|"Full")
    returnFormat (XML|Other)>
<!ELEMENT resultPosition EMPTY>
<!ATTLIST resultPosition
    startPosition CDATA #DEFAULT "0"
    sortKey NMTOKEN #IMPLIED>>
```

URL Component	Require Description Option
---------------	-------------------------------

URL Component	Required Option	Description
http://server_address/path/script?	R	URL prefix of server.
VERSION=version	R	Request version.
REQUEST=GetDescriptors	R	Request name.
queryLanguage	R	Query Language of Query Expression to follow OGC standards. Allowed values are OGC Common, OGC Filter, SQL. Must be in list of query languages in the response to GetCapabilities from this Service.
attributeSetName	R	Url of attribute names used in Query expression. Must be in list of attribute namespaces list in the response to GetCapabilities
queryExpression	R	String expressing constraints used to select descriptors to return
maxRecs	O	Maximum number of descriptors to be returned. If not specified returns all matching descriptors or service maximum defined in GetCapabilities response
catalogType	O	One of Collection, Product or Service for general search descriptors to search
QueryScope	O	Local or Cascade. Default is Local
elementSchema	O	Schema used for result sets. Must be in list of element schemas list in the response to GetCapabilities. If not listed in GetCapabilities response
elementSetName	O	Element set name to determine which descriptor should be returned. Current allowed values are Brief, Full, and Summary. Must be in list of query languages in the response to GetCapabilities from this Service
startPosition	O	The startPosition parameter identifies the first record entry to be returned. If not specified the default is 1
sortKey	O	The sortField parameter specifies how the descriptor data is to be sorted prior to presentation.
returnFormat	O	The returnFormat parameter specifies the encoding standard to be used for returning the result set. Currently the only allowed value is XML

GetDescriptors Response

The GetDescriptors response is an XML document which contains information on the success of the query and descriptors of objects at one of three levels of details. The DTD which appears below defines those common elements giving background and status of the query. The three DTDs in annex B demonstrate the differing levels of detail in the brief, summary and full element sets

Currently the only available element sets are based on the FGDC and ISO TC211 metadata standards

```
<!ENTITY % DistinguishedName "(nameValue,nameNameSpace)" >

<!ENTITY % CI_OnLineResource "(linkage,protocol?,applicationProfile?,
                                onlineResourceName?,onlineResourceDescription?,functionCode?)" >

<!--The top level holder for a response from the catalog.-->
<!ELEMENT searchResponse (searchParameter*,searchStatus,searchResults) >

<!ATTLIST searchResponse
            DTD_Version CDATA #FIXED "1.1.0" >

<!ELEMENT searchParameter (#PCDATA) >

<!ATTLIST searchParameter
            name CDATA #REQUIRED >

<!--This tag contains a number of attributes that detail the status of the
search.
-->
<!ELEMENT searchStatus EMPTY >

<!--elementSetName - The element set that has been returned (e.g., brief,
summary, full)
numberOfRecords - The number of hits returned in this result.
schema - The type of response returned (e.g., OGCService, FGDC)
success - Was the search successful (true, false)
timestamp - The date and time at the completion of the search
-->
<!ATTLIST searchStatus
            elementSetName CDATA #FIXED "brief"
            success CDATA "true"
            numberOfRecords CDATA #IMPLIED
            schema CDATA #FIXED "ISO19119"
            timestamp CDATA #IMPLIED >

<!--The holder for the results from the catalog.-->
<!ELEMENT searchResults (ISO19119*) >
```

<REGISTERSERVICE> INTERFACE

A WRS may want to enable service providers to register their services and have the WRS ingest their current capabilities document and create appropriate service descriptors in the WRS descriptor storage. The WRS may also provide the capability to automatically update the descriptor base for a service on a predefined schedule or based on adhoc requests. This set of operations is provided by the Register Service Interface. These operations are similar in functionality to the CG_CatalogManager interface .

RegisterService Request

<i>URL Component</i>	<i>Require</i>	<i>Description</i>
	<i>Option</i>	
http://server_address/path/script?	R	URL prefix of server.
VERSION=version	R	Request version.
REQUEST=RegisterService	R	Request name.
ServiceAddr	R	URL of the Service GetCapability Operation
ServiceOwnerContactInfo	O	Contact point for notification of events
HarvestFrequency	O	Frequency for catalog to check for and harvest new versions of the Service capabilities document
Push updates allowed (editor's note allow push update we will probably new service	O	May a service provider push his update to the WRS in what form. New Capability object or a WFS Transaction stream
oVendor-specific parameters	O	Optional experimental parameters.

Response

TBD- Concept of a message sent to the ServiceOwner on email or an event flag

Annex A: WMT2 Testbed Implementation

Overview

The Stateless Catalog prototypes made many simplifying assumptions. Including:

- Support of one Web Service, the WMS
- Support of one query Language, OGC_Common
- Support of a few searchable attribute set
- No support of queryScope, sortField

Annex A.1: Request Syntax

It is Searchable via HTTP GET, with the following CGI parameters:

Name	Description	Allowed Values
ESN	element set name	'b', 's', 'f' (for brief, summary and full)
RS	results set format	'XML'
QUERY	ogc simple query	See Below
DOCID	id of service	String
SCHEMA	Schema used for result sets	'FGDC', 'ISO19119'
'MAXRECS'	m,ax number of records to be returned.	integer

For example:

<http://ogc.compusult.nf.ca/cgi-bin/catquery-iso/catquery?RS=XML&QUERY=SRS%3D%27EPSG%3A4326%27%0D%0AAND+title+like+%27%25COASTLINES%25%27%0D%0AAND+%28Format%3D%27gif%27+OR+format%3D%27jpeg%27%29%0D%0AAND+ServiceType%3D%27Web+Mapping+Layer%27+%0D%0AAND+Intersects%28LatLongBoundingBox%2C%0D%0A%09%09Envelope%28-160%2C+-50%2C+50%2C+10%29%29%0D%0A%09%09%09&SCHEMA=ISO19119&ESN=b>

The URL (GET or POST) syntax supported is:

<http://host[:port]>http://host[:port]/path/script?[DOCID=docid]&Query=ogc_common_query&am
p;ESN=esn&RS=rs&MAXRECS=maxrecs

With the parameters as defined below. The additional CGI parameter 'MAXRECS' allows a limit to be specified on the number of records returned from a search.

The basic processing model is that clients would do a search with a 'brief' or 'summary' element (in the ESN parameter) to retrieve a condensed version of the service of interest and then, if needed, could retrieve the full element set through the DOCID parameter (extracted from the brief or summary element set).

E.g.

```
<http://lennier.gsfc.nasa.gov:8080/wmt/search.pl?QUERY=title+like+%27%25coastlines%25%27+and+ServiceType%3D%27Web+Mapping+Layer%27&ESN=b&amp;RS=XML&amp;MAXRECS=1>http://lennier.gsfc.nasa.gov:8080/wmt/search.pl?QUERY=title+like+%27%25coastlines%25%27+and+ServiceType%3D%27Web+Mapping+Layer%27&ESN=b&amp;RS=XML&amp;MAXRECS=1
```

```
<http://lennier.gsfc.nasa.gov:8080/wmt/search.pl?DOCID=S_F_NASA_GLOBE>http://lennier.gsfc.nasa.gov:8080/wmt/search.pl?DOCID=S_F_NASA_GLOBE
```

XML records are returned as a result of the search, DTD's of which are contained in:

```
<http://coverage.opengis.org/ip2000/docs/iso19119-20001108.zip>http://coverage.opengis.org/ip2000/docs/iso19119-20001108.zip
```

Annex B: ISO 19115 DTDs

BRIEF DTD

```
<!ENTITY % DistinguishedName "(nameValue,nameNameSpace)" >

<!ENTITY % CI_OnLineResource "(linkage,protocol?,applicationProfile?,
                                onlineResourceName?,onlineResourceDescription?,functionCode?)" >

<!--The top level holder for a response from the catalog.-->
<!ELEMENT searchResponse (searchParameter*,searchStatus,searchResults) >

<!ATTLIST searchResponse
            DTD_Version CDATA #FIXED "1.1.0" >

<!ELEMENT searchParameter (#PCDATA) >

<!ATTLIST searchParameter
            name CDATA #REQUIRED >

<!--This tag contains a number of attributes that detail the status of the
search.
-->
<!ELEMENT searchStatus EMPTY >

<!--elementSetName - The element set that has been returned (e.g., brief,
summary, full)
numberOfRecords - The number of hits returned in this result.
schema - The type of response returned (e.g., OGCService, FGDC)
success - Was the search successful (true, false)
timestamp - The date and time at the completion of the search
-->
<!ATTLIST searchStatus
            elementSetName CDATA #FIXED "brief"
            success CDATA "true"
```

```

        numberOfRecords CDATA      #IMPLIED
        schema CDATA      #FIXED "ISO19119"
        timestamp CDATA      #IMPLIED >

<!--The holder for the results from the catalog.-->
<!ELEMENT searchResults (ISO19119*) >

<!--The top level tag for a result created with the ISO 19119 schema.-->
<!ELEMENT ISO19119 (title,serviceName?,serviceType,serviceTypeVersion,
        onLineResource?) >

<!--relevanceRank - The relative value of this hit, higher = better
serviceId - The unique identifier for this service. Can be used to later
retrieve additional information.
timestamp - The catalog date for this service record.-->
<!ATTLIST ISO19119
        relevanceRank CDATA      " "
        timestamp CDATA      #IMPLIED
        serviceId CDATA      #REQUIRED >

<!ELEMENT title (#PCDATA) >

<!ELEMENT serviceName (#PCDATA) >

<!ELEMENT serviceType %DistinguishedName; >

<!ELEMENT serviceTypeVersion (#PCDATA) >

<!ELEMENT onLineResource %CI_OnLineResource; >

<!ELEMENT nameValue (#PCDATA) >

<!ELEMENT nameNameSpace (#PCDATA) >

<!ELEMENT linkage EMPTY >

<!ATTLIST linkage
        xmlns:xlink CDATA      #FIXED "http://www.w3.org/1999/xlink"

        xlink:type CDATA      #FIXED "simple"
        xlink:href CDATA      #REQUIRED >

<!ELEMENT protocol (#PCDATA) >

<!ELEMENT applicationProfile (#PCDATA) >

<!ELEMENT onlineResourceName (#PCDATA) >

<!ELEMENT onlineResourceDescription (#PCDATA) >

<!ELEMENT functionCode EMPTY >

<!ATTLIST functionCode
        value (access | additionalInformation | download | order |
search)
        #REQUIRED >

```

Summary DTD

```
<!ENTITY % DistinguishedName "(nameValue,nameNameSpace)" >

<!ENTITY % CI_OnLineResource "(linkage,protocol?,applicationProfile?,
                               onlineResourceName?,onlineResourceDescription?,functionCode?)" >

<!ENTITY % ValueType "(dataType | instanceValue | range | enumValues)" >

<!ENTITY % typedDataValue
"(valueTitle?,valueDescription?,valueOnLineResource?,
 value)" >

<!--The top level holder for a response from the catalog.-->
<!ELEMENT searchResponse (searchParameter*,searchStatus,searchResults) >

<!ATTLIST searchResponse
          DTD_Version CDATA #FIXED "1.1.0" >

<!ELEMENT searchParameter (#PCDATA) >

<!ATTLIST searchParameter
          name CDATA #REQUIRED >

<!--This tag contains a number of attributes that detail the status of the
search.
-->
<!ELEMENT searchStatus EMPTY >

<!--elementSetName - The element set that has been returned (e.g., brief,
summary, full)
numberOfRecords - The number of hits returned in this result.
schema - The type of response returned (e.g., OGCService, FGDC)
success - Was the search successful (true, false)
timestamp - The date and time at the completion of the search
-->
<!ATTLIST searchStatus
          elementSetName CDATA #FIXED "summary"
          success CDATA "true"
          numberOfRecords CDATA #IMPLIED
          schema CDATA #FIXED "ISO19119"
          timestamp CDATA #IMPLIED >

<!--The holder for the results from the catalog.-->
<!ELEMENT searchResults (ISO19119*) >

<!--The top level tag for a result created with the ISO 19119 schema.-->
<!ELEMENT ISO19119 (title,serviceName?,serviceType,serviceTypeVersion,
                   onLineResource?,keywords*,typeProperty*,accessProperties,
                   legalConstraints*,securityConstraints*,DCP+) >

<!--relevanceRank - The relative value of this hit, higher = better
serviceId - The unique identifier for this service. Can be used to later
retrieve additional information.
```

```

timestamp - The catalog date for this service record.-->
<!ATTLIST ISO19119
    relevanceRank CDATA " "
    timestamp CDATA #IMPLIED
    serviceId CDATA #REQUIRED >

<!ELEMENT title (#PCDATA) >

<!ELEMENT serviceName (#PCDATA) >

<!ELEMENT serviceType %DistinguishedName; >

<!ELEMENT serviceTypeVersion (#PCDATA) >

<!ELEMENT onLineResource %CI_OnLineResource; >

<!ELEMENT keywords (keyword*,typeCode?,thesaurusName?) >

<!ELEMENT typeProperty (typeName,typeValue) >

<!ELEMENT accessProperties
(fees?,plannedAvailableDateTime?,orderingInstructions?,
    turnaround?) >

<!ELEMENT legalConstraints
(useLimitation*,propertyRightsCode*,useConstraintsCode*,
    otherConstraints*) >

<!ELEMENT securityConstraints (useLimitation*,classificationCode,userNote?,
    classificationSystem?,handlingDescription?) >

<!ELEMENT DCP (invocationName,connectPoint) >

<!ATTLIST DCP
    type (HTTPGet | HTTPPost) #REQUIRED >

<!ELEMENT keyword (#PCDATA) >

<!ELEMENT typeCode EMPTY >

<!ATTLIST typeCode
    KeyType (discipline | place | stratum | temporal | theme)
    #REQUIRED >

<!ELEMENT thesaurusName (#PCDATA) >

<!ELEMENT typeName %DistinguishedName; >

<!ELEMENT typeValue %ValueType; >

<!ELEMENT fees (#PCDATA) >

<!ELEMENT plannedAvailableDateTime (#PCDATA) >

<!ELEMENT orderingInstructions (#PCDATA) >

<!ELEMENT turnaround (#PCDATA) >

```

```

<!ELEMENT useLimitation  (#PCDATA) >

<!ELEMENT propertyRightsCode  EMPTY  >

<!ATTLIST propertyRightsCode
          Restrict (copyright | patent | patentPending | license |
                    intellectualPropertyRights | trademark)  #REQUIRED  >

<!ELEMENT useConstraintsCode  EMPTY  >

<!ATTLIST useConstraintsCode
          Restrict (copyright | patent | patentPending | license |
                    intellectualPropertyRights | trademark)  #REQUIRED  >

<!ELEMENT otherConstraints  (#PCDATA) >

<!ELEMENT classificationCode  EMPTY  >

<!ATTLIST classificationCode
          Classification (unclassified | codeWord | confidential | secret
                        |
                        restricted | topsecret)  #REQUIRED  >

<!ELEMENT userNote  (#PCDATA) >

<!ELEMENT classificationSystem  (#PCDATA) >

<!ELEMENT handlingDescription  (#PCDATA) >

<!ELEMENT invocationName  (#PCDATA) >

<!ELEMENT connectPoint  %CI_OnLineResource; >

<!ELEMENT nameValue  (#PCDATA) >

<!ELEMENT nameNameSpace  (#PCDATA) >

<!ELEMENT linkage  EMPTY  >

<!ATTLIST linkage
          xmlns:xlink  CDATA  #FIXED "http://www.w3.org/1999/xlink"

          xlink:type  CDATA  #FIXED "simple"
          xlink:href  CDATA  #REQUIRED  >

<!ELEMENT protocol  (#PCDATA) >

<!ELEMENT applicationProfile  (#PCDATA) >

<!ELEMENT onlineResourceName  (#PCDATA) >

<!ELEMENT onlineResourceDescription  (#PCDATA) >

<!ELEMENT functionCode  EMPTY  >

<!ATTLIST functionCode

```

```

        value (access | additionalInformation | download | order |
search)
        #REQUIRED >

<!ELEMENT dataType EMPTY >

<!ATTLIST dataType
        type (string | number) #REQUIRED >

<!ELEMENT instanceValue %typedDataValue; >

<!ELEMENT range (minimumValue,maximumValue) >

<!ELEMENT enumValues (%typedDataValue;)* >

<!ELEMENT minimumValue %typedDataValue; >

<!ELEMENT maximumValue %typedDataValue; >

<!ELEMENT valueTitle (#PCDATA) >

<!ELEMENT valueDescription (#PCDATA) >

<!ELEMENT valueOnLineResource %CI_OnLineResource; >

<!ELEMENT value (#PCDATA) >

<!ATTLIST value
        type (string | number) #REQUIRED >

```

Full DTD

```

<!ENTITY % CI_Citation "(title,alternateTitle*,date+,edition?,editionDate?,
        identifier*,identifierType*,citedResponsibleParty*,
        presentationFormCode*,seriesName?,issueIdentification?,
        otherCitationDetails?,collectionTitle?,page?,ISBN?,ISSN?)" >

<!ENTITY % CI_ResponsibleParty
"(individualName*,organisationName*,positionName*,
        contactInfo*,roleCode+)" >

<!ENTITY % DistinguishedName "(nameValue,nameNameSpace)" >

<!ENTITY % ValueType "(dataType | instanceValue | range | enumValues)" >

<!ENTITY % CI_OnLineResource "(linkage,protocol?,applicationProfile?,
        onlineResourceName?,onlineResourceDescription?,functionCode?)" >

<!ENTITY % typedDataValue
"(valueTitle?,valueDescription?,valueOnLineResource?,
        value)" >

<!--The top level holder for a response from the catalog.-->
<!ELEMENT searchResponse (searchParameter*,searchStatus,searchResults) >

<!ATTLIST searchResponse

```

```

                DTD_Version  CDATA      #FIXED "1.1.0" >

<!--ELEMENT searchParameter  (#PCDATA) >

<!--ATTLIST searchParameter
        name  CDATA      #REQUIRED  >

<!--This tag contains a number of attributes that detail the status of the
search.
-->
<!--ELEMENT searchStatus      EMPTY  >

<!--elementSetName - The element set that has been returned (e.g., brief,
summary, full)
numberOfRecords - The number of hits returned in this result.
schema - The type of response returned (e.g., OGCService, FGDC)
success - Was the search successful (true, false)
timestamp - The date and time at the completion of the search
-->
<!--ATTLIST searchStatus
        elementSetName  CDATA      #FIXED "full"
        success  CDATA      "true"
        numberOfRecords  CDATA      #IMPLIED
        schema  CDATA      #FIXED "ISO19119"
        timestamp  CDATA      #IMPLIED  >

<!--The holder for the results from the catalog.-->
<!--ELEMENT searchResults  (ISO19119*) >

<!--The top level tag for a result created with the ISO 19119 schema.-->
<!--ELEMENT ISO19119
(citation,abstract,purpose?,credit?,statusCode*,pointOfContact*,
resourceSpecifiedUsage*,serviceTypeVersion,serviceType,typeProperty*,
accessProperties,legalConstraints*,securityConstraints*,quality?,
keywords*,operationMetadata*) >

<!--relevanceRank - The relative value of this hit, higher = better
serviceId - The unique identifier for this service. Can be used to later
retrieve additional information.
timestamp - The catalog date for this service record.-->
<!--ATTLIST ISO19119
        relevanceRank  CDATA      " "
        timestamp  CDATA      #IMPLIED
        serviceId  CDATA      #REQUIRED  >

<!--ELEMENT citation  %CI_Citation; >

<!--ELEMENT abstract  (#PCDATA) >

<!--ELEMENT purpose  (#PCDATA) >

<!--ELEMENT credit  (#PCDATA) >

<!--ELEMENT statusCode  EMPTY  >

```

```

<!ATTLIST statusCode
    progressCode (completed | historicalArchive | obsolete | onGoing
    |
    planned | required | underdevelopment)    #REQUIRED  >

<!ELEMENT pointOfContact    %CI_ResponsibleParty; >

<!ELEMENT resourceSpecifiedUsage    (specifiedUsage,usageDateTime?,
    userDetirminedLimitations?,userContactInfo+) >

<!ELEMENT serviceTypeVersion    (#PCDATA) >

<!ELEMENT serviceType    %DistinguishedName; >

<!ELEMENT typeProperty    (typeName,typeValue) >

<!ELEMENT accessProperties
(fees?,plannedAvailableDateTime?,orderingInstructions?,
    turnaround?) >

<!ELEMENT legalConstraints
(useLimitation*,propertyRightsCode*,useConstraintsCode*,
    otherConstraints*) >

<!ELEMENT securityConstraints    (useLimitation*,classificationCode,userNote?,
    classificationSystem?,handlingDescription?) >

<!ELEMENT quality    (TBD_ServiceQuality) >

<!ELEMENT keywords    (keyword*,typeCode?,thesaurusName?) >

<!ELEMENT operationMetadata    (operationName,operationDescription,parameter*,
    dependsOn?,DCP+) >

<!ELEMENT specifiedUsage    (#PCDATA) >

<!ELEMENT usageDateTime    (#PCDATA) >

<!ELEMENT userDetirminedLimitations    (#PCDATA) >

<!ELEMENT userContactInfo    (individualName*,organisationName*,positionName*,
    contactInfo*,roleCode+) >

<!ELEMENT typeName    %DistinguishedName; >

<!ELEMENT typeValue    %ValueType; >

<!ELEMENT fees    (#PCDATA) >

<!ELEMENT plannedAvailableDateTime    (#PCDATA) >

<!ELEMENT orderingInstructions    (#PCDATA) >

<!ELEMENT turnaround    (#PCDATA) >

<!ELEMENT useLimitation    (#PCDATA) >

```

```

<!ELEMENT propertyRightsCode    EMPTY    >

<!ATTLIST propertyRightsCode
    Restrict (copyright | patent | patentPending | license |
    intellectualPropertyRights | trademark)    #REQUIRED    >

<!ELEMENT useConstraintsCode    EMPTY    >

<!ATTLIST useConstraintsCode
    Restrict (copyright | patent | patentPending | license |
    intellectualPropertyRights | trademark)    #REQUIRED    >

<!ELEMENT otherConstraints    (#PCDATA)    >

<!ELEMENT classificationCode    EMPTY    >

<!ATTLIST classificationCode
    Classification (unclassified | codeWord | confidential | secret
    |
    restricted | topsecret)    #REQUIRED    >

<!ELEMENT userNote    (#PCDATA)    >

<!ELEMENT classificationSystem    (#PCDATA)    >

<!ELEMENT handlingDescription    (#PCDATA)    >

<!ELEMENT TBD_ServiceQuality    (#PCDATA)    >

<!ELEMENT keyword    (#PCDATA)    >

<!ELEMENT typeCode    EMPTY    >

<!ATTLIST typeCode
    KeyType (discipline | place | stratum | temporal | theme)
    #REQUIRED    >

<!ELEMENT thesaurusName    (#PCDATA)    >

<!ELEMENT operationName    %DistinguishedName;    >

<!ELEMENT operationDescription    (#PCDATA)    >

<!ELEMENT parameter    (parameterName,parameterType,parameterDescription?,
    permittedValues)    >

<!ATTLIST parameter
    optional (yes | no)    #REQUIRED
    repeatable (true | false)    #REQUIRED
    direction (in | out | inout)    #REQUIRED    >

<!ELEMENT dependsOn    (operationName*)    >

<!ELEMENT DCP    (invocationName,connectPoint,parameter*)    >

<!ATTLIST DCP
    type (HTTPGet | HTTPPost)    #REQUIRED    >

```

```

<!ELEMENT title    (#PCDATA) >

<!ELEMENT alternateTitle    (#PCDATA) >

<!ELEMENT date    (#PCDATA) >

<!ATTLIST date
            dateType (creation | publication | revision)    #REQUIRED >

<!ELEMENT edition    (#PCDATA) >

<!ELEMENT editionDate    (#PCDATA) >

<!--"editionDate" has a domain of Date which is defined in another standard-->
<!ELEMENT identifier    (#PCDATA) >

<!ELEMENT identifierType    (#PCDATA) >

<!ELEMENT citedResponsibleParty    %CI_ResponsibleParty; >

<!ELEMENT presentationFormCode    EMPTY >

<!ATTLIST presentationFormCode
            value (document | hardcopyMap | image | model | profile |
rasterMap |
            table | vectorMap | view)    #REQUIRED >

<!ELEMENT seriesName    (#PCDATA) >

<!ELEMENT issueIdentification    (#PCDATA) >

<!ELEMENT otherCitationDetails    (#PCDATA) >

<!ELEMENT collectionTitle    (#PCDATA) >

<!ELEMENT page    (#PCDATA) >

<!ELEMENT ISBN    (#PCDATA) >

<!ELEMENT ISSN    (#PCDATA) >

<!--count of ("individualName" + "organisationName" + "positionName") > 0-->
<!ELEMENT individualName    (#PCDATA) >

<!ELEMENT organisationName    (#PCDATA) >

<!ELEMENT positionName    (#PCDATA) >

<!ELEMENT contactInfo    (phone?,address?,onLineResource?,hoursOfService?,
            contactInstructions?) >

<!ELEMENT roleCode    EMPTY >

<!ATTLIST roleCode
            value (contentProvider | custodianSteward | owner | user |
distributor | metadataProvider | originator | pointOfContact |

```

```

principalInvestigator | processor | publisher) #REQUIRED >

<!ELEMENT nameValue (#PCDATA) >

<!ELEMENT nameNameSpace (#PCDATA) >

<!ELEMENT parameterName %DistinguishedName; >

<!ELEMENT parameterType EMPTY >

<!ATTLIST parameterType
    type (string | number) #REQUIRED >

<!ELEMENT parameterDescription (#PCDATA) >

<!ELEMENT permittedValues (onLineResource?, (%ValueType;)* ) >

<!ELEMENT invocationName (#PCDATA) >

<!ELEMENT connectPoint %CI_OnLineResource; >

<!ELEMENT phone (voice*, facsimile*, other*, otherType*) >

<!ELEMENT address
    (deliveryPoint*, city?, administrativeArea?, postalCode?, country?,
    electronicMailAddress*) >

<!ELEMENT onLineResource %CI_OnLineResource; >

<!ELEMENT hoursOfService (#PCDATA) >

<!ELEMENT contactInstructions (#PCDATA) >

<!ELEMENT dataType EMPTY >

<!ATTLIST dataType
    type (string | number) #REQUIRED >

<!ELEMENT instanceValue %typedDataValue; >

<!ELEMENT range (minimumValue, maximumValue) >

<!ELEMENT enumValues (%typedDataValue;)* >

<!-- "other" is mandatory if "voice" and "facsimile" not provided
"otherType" is mandatory if "other" is provided-->
<!ELEMENT voice (#PCDATA) >

<!ELEMENT facsimile (#PCDATA) >

<!ELEMENT other (#PCDATA) >

<!ELEMENT otherType (#PCDATA) >

<!ELEMENT deliveryPoint (#PCDATA) >

<!ELEMENT city (#PCDATA) >

```

```

<!ELEMENT administrativeArea (#PCDATA) >

<!ELEMENT postalCode (#PCDATA) >

<!ELEMENT country (#PCDATA) >

<!ELEMENT electronicMailAddress (#PCDATA) >

<!ELEMENT minimumValue %typedDataValue; >

<!ELEMENT maximumValue %typedDataValue; >

<!ELEMENT linkage EMPTY >

<!ATTLIST linkage
            xmlns:xlink CDATA #FIXED "http://www.w3.org/1999/xlink"
            xlink:type CDATA #FIXED "simple"
            xlink:href CDATA #REQUIRED >

<!ELEMENT protocol (#PCDATA) >

<!ELEMENT applicationProfile (#PCDATA) >

<!ELEMENT onlineResourceName (#PCDATA) >

<!ELEMENT onlineResourceDescription (#PCDATA) >

<!ELEMENT functionCode EMPTY >

<!ATTLIST functionCode
            value (access | additionalInformation | download | order |
search)
            #REQUIRED >

<!ELEMENT valueTitle (#PCDATA) >

<!ELEMENT valueDescription (#PCDATA) >

<!ELEMENT valueOnLineResource %CI_OnLineResource; >

<!ELEMENT value (#PCDATA) >
<!ATTLIST value
            type (string | number) #REQUIRED >

```

Annex B: Get Capabilities Response

Annex C: References

[need to update this list]

- [1] OpenGIS Project Document 97-009: "WWW Mapping Framework", Allan Doyle, BBN Corporation
- [2] OpenGIS Project Document 98-060: "User Interaction with Geospatial data", Adrian Cuthbert, Laser-Scan
- [3] OpenGIS Project Document 98-068: "A Web Mapping Scenario", Kenn Gardels, University of California, Berkeley
- [4] OpenGIS "A Request for Technology In Support of a Web Mapping Technology Testbed", October 28, 1998
- [5] OpenGIS "Request For Quotation and Call For Participation in the OGC Web Mapping Testbed Initial Operating Capability and Demonstration", March 8, 1999
- [6] OpenGIS Specification for Simple Features for OLE/COM
- [7] OpenGIS Catalog Interface Implementation Specification (Version 1.0)
- [8] Fielding et. al., "RFC 2616: Hypertext Transfer Protocol – HTTP 1/1," June 1999, Internet Society Network Working Group, <http://www.ietf.org/rfc/rfc2616.txt>
- [9] Bray, Paoli, Sperberg-McQueen, eds., "Extensible Markup Language (XML) 1.0", 10 February, 1998, World Wide Web Consortium
- [10] OpenGIS Project Document 99-086: "Standardize Spatial Reference System (SRS) Identifiers", Arliss Whiteside, BAE SYSTEMS Mission Solutions
- [11] Borenstein N., and N. Freed, "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies", RFC 1521, Bellcore, Innosoft, September 1993, <http://www.ietf.org/rfc/rfc1521.txt>.
- [12] Berners-Lee, T., Fielding, N., and Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax", IETF RFC 2396, <http://www.ietf.org/rfc/rfc2396.txt>
- [13] European Petroleum Survey Group, "EPSG Geodesy Parameters", <http://www.petroconsultants.com/products/geodetic.html>
- [14] Web Mapping Testbed, "WM Automatic Projections", <http://www.digitalearth.gov/wmt/auto.html>
- [15] Bradner, Scott, "RFC 2119 Key words for use in RFCs to Indicate Requirement Levels," March 1997, <http://www.ietf.org/rfc/rfc2119.txt>
- [16] Whitehead, E. et. Al., "XML Media Types," July 1998, <http://www.ietf.org/rfc/rfc2376.txt>
- [17] National Center for Supercomputing Applications, "The Common Gateway Interface," undated, <http://hoohoo.ncsa.uiuc.edu/cgi/>
- [18] XPath specification.