

建設情報連携のためのポータル標準インタフェースガイドライン(案)
第1版

平成19年10月

国土交通省

はじめに

公共事業では、ライフサイクルの各段階の業務を支援するために、さまざまな情報システムが整備・運用されている。これらのシステムは、それぞれの目的に応じて独立に運用され効果を上げているが、複数の情報システムを連携して利用できれば、さらに大きな効果が得られると期待される。しかしシステム間の連携を図るための適切な管理基準や制度が存在しないために、分野横断的な情報連携が困難なことがある。このため、建設分野において流通するデータの標準化、および分散システムの検索・利用技術の確立は、業界全体の効率化推進に特に大きな効果があると期待される。

情報を有効に利活用するためには、「情報の蓄積と管理」が不可欠である。国土交通省では、直轄事業のすべての業務・工事を対象に電子納品を実施している。また、地方自治体でも電子納品が展開されてきており、様々な建設情報が蓄積されている。今後、蓄積された様々な建設情報を効率よく検索・収集・利活用していくためには、情報システムの基盤整備が特に重要な課題と言える。

蓄積された建設情報を有効活用するには、情報システムの連携による相互運用性を高めていく必要がある。相互運用性を高める方策として、既存のデータベースの構造に大きな変化を加えることなく、何らかの標準的なインタフェースを用いて分散環境にある情報システムを連携させる手法が考えられる。建設情報の分野横断的な連携利用の実現のためには、既存の規格や事例を踏まえた、建設分野全体を対象とした分散システムの連携と、データ検索・利用技術の仕組みを開発することが望まれる。

本書は、建設分野で流通する情報を、効率よく検索・収集・利活用できるようにするため、情報システムの連携による相互運用性を高めるための「標準的なインタフェース」として、インターネット/イントラネット環境で動作するアプリケーションプログラムが実装すべきインタフェースの考え方を示すとともに、インタフェースの仕様を定めている。

本書の策定にあたっては、国土交通省が(財)日本建設情報総合センターに業務を委託し、建設情報標準化委員会 電子地図/建設情報連携小委員会(柴崎亮介小委員長)にご検討・ご審議をいただいた。多大なご協力をいただいた関係各位に深く謝意を表する次第である。

平成 19 年 10 月

国土交通省

<目次>

【本編】

1. ガイドラインの概要.....	1
1-1. 目的.....	1
1-2. 本ガイドライン策定の背景と標準インタフェースの意義.....	2
1-3. 適用範囲.....	5
1-4. 利用者と利用方法.....	6
1-5. 引用規格.....	9
1-5-1. 国際的な標準化団体による標準規格.....	9
1-5-2. 既存の標準インタフェース仕様との関係.....	10
1-6. 用語の定義.....	12
2. 標準インタフェースの考え方.....	15
2-1. 本ガイドラインにおける標準インタフェースの位置づけ.....	15
2-2. 標準インタフェースの定義.....	20
2-3. 標準インタフェースの種類.....	21
2-4. 標準インタフェースの定義方法.....	24
2-5. 関数仕様の考え方.....	25
2-6. 関数仕様の実装方式.....	26
2-7. 標準インタフェースを利用したアプリケーションプログラムの運用形態.....	27
2-8. インタフェース公開・管理の考え方.....	28
2-9. インタフェース接続の考え方.....	30
3. 標準インタフェース関数仕様.....	32
3-1. 関数の一覧.....	33
3-2. 関数の説明.....	35
3-2-1. 共通.....	35
3-2-2. 地理情報に関するインタフェース.....	36
3-2-3. 地名辞典に関するインタフェース.....	38
3-2-4. メタデータ検索に関するインタフェース.....	53
3-2-5. 構造体.....	58
4. 標準インタフェース実装仕様.....	82
4-1. POST/GET 利用のための実装仕様.....	82
4-1-1. 地理情報に関するインタフェース.....	82
4-1-2. 地名辞典に関するインタフェース.....	82
4-1-3. メタデータ検索に関するインタフェース.....	82
4-2. SOAP と WSDL 利用のための実装仕様.....	83

4-2-1.	共通事項	83
4-2-2.	各サービスに共通するインタフェース	86
4-2-3.	地理情報に関するインタフェース	89
4-2-4.	地名辞典（空間参照系管理）に関するインタフェース	90
4-2-5.	地名辞典（地名辞典管理）に関するインタフェース	100
4-2-6.	メタデータ検索に関するインタフェース	110
5.	関数仕様作成に関する取り決め	111
5-1.	関数仕様作成に関する取り決め	111
5-2.	関数と引数の命名規則	116
6.	実装仕様作成に関する取り決め	118
6-1.	実装仕様作成に関する取り決め	118
7.	標準インタフェースの運用に関する取り決め	120
7-1.	インタフェース公開・管理方法	120
7-2.	インタフェース接続方式	122
8.	標準インタフェース公開・管理仕様	123
8-1.	アプリケーションプログラムの内容に関する公開・管理仕様	124
8-2.	アプリケーションプログラムで利用できる関数等の技術的な公開・管理仕様	127

【附属書】

1. 参考とすべきインタフェース.....	1
1-1. 地物単位で地理情報を取得するインタフェース（参考）.....	2
1-1-1. 関数仕様.....	2
1-1-2. 詳細情報の入手先.....	3
1-2. SOAP による地理情報等の標準インタフェース（参考）.....	4
1-2-1. 関数仕様.....	5
1-2-2. 詳細情報の入手先.....	10
2. ガイドラインの適用事例.....	11
2-1. システム実装の事例.....	12
2-1-1. 概要.....	12
2-2. 連携システムの詳細.....	14
2-2-2. 実装の手順（例）.....	17
2-3. インタフェース策定の事例.....	23
2-3-1. 河川 GIS・河川アプリケーション標準インタフェースガイドラインにお ける事例.....	23
2-3-2. 電子納品保管管理システムにおける事例.....	25

1. ガイドラインの概要

1-1. 目的

「建設情報連携のための標準インタフェースガイドライン」(以下、「本ガイドライン」という)は、インターネット/イントラネット環境で動作するアプリケーションプログラムが実装すべきインタフェースの考え方を示すとともに、インタフェースの仕様/取り決めについて定める。

なお本ガイドラインでは、各節の要点を枠内に記述し、本文を枠外に記述する。

【解説】

本ガイドラインでは、国及び地方公共団体などの建設情報を作成、利用する各機関・各部署で採用・開発する情報システムを相互に連携するための考え方と方法論を示している。方法論については、「仕様」と「取り決め」として取りまとめている。本ガイドラインに記載している内容は、概ね以下のとおりである。

- インタフェースの考え方
- インタフェースを利用したシステムの開発と運用
- インタフェースに関わる仕様
- インタフェースに関わる取り決め

1-2. 本ガイドライン策定の背景と標準インターフェースの意義

現在稼働中のアプリケーションプログラムの多くは、システム間の連携が困難であり、データの連携ができないだけでなく、利用者にとっても様々なシステムを操作しなければならない不便な状況を作っている。そこで、異なるアプリケーションプログラム間において円滑な情報共有を実現するために、アプリケーションプログラム間の情報連携の標準的なルールとして「標準インターフェース」を策定し、これを導入することとした。標準インターフェースの導入により、アプリケーションプログラムの相互運用性が高まり、分野横断的な情報共有が可能となる。

【解説】

現在稼働中のアプリケーションプログラムの多くは、開発者の違い、開発言語の違い、プログラムの処理方法の違い、データ仕様の違いにより、システム間の連携が困難であり、データの連携ができないだけでなく、利用者にとっても様々なシステムを操作しなければならない不便な状況を作っている（図 1-1）。

例えば、国道事務所における業務を例にとると、ある道路の盛土区間で亀裂等の変状が見つかった場合、当該区間の状況は道路管理システムで、地質等の状況は地盤情報データベースから、試験等の竣工時の状況は電子納品保管管理システムから、それぞれ検索して入手する必要がある（図 1-2）。

データの取得に関する処理方法を標準化し、それを標準インターフェースとして公開することで、道路管理システムを通じて地盤情報データベースに対して当該区間のボーリング柱状図を要求したり、電子納品保管管理システムに対して当該区間の竣工図書を要求したりすることが可能となる（図 1-3）。これにより、今まで困難であったデータの入手・利用が容易になるだけでなく、新規のシステム開発の場合にすでにあるシステムやデータを利用することが可能となり、データ整備やシステム開発のコストの縮減につながる効果が見込まれる。

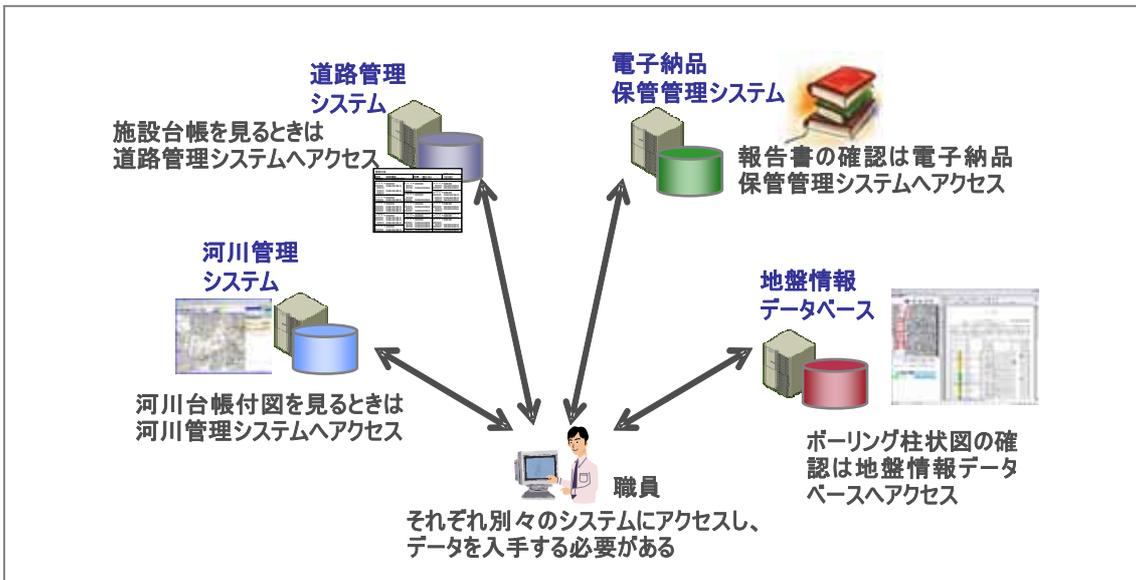


図 1-1 現在のシステム利用状況

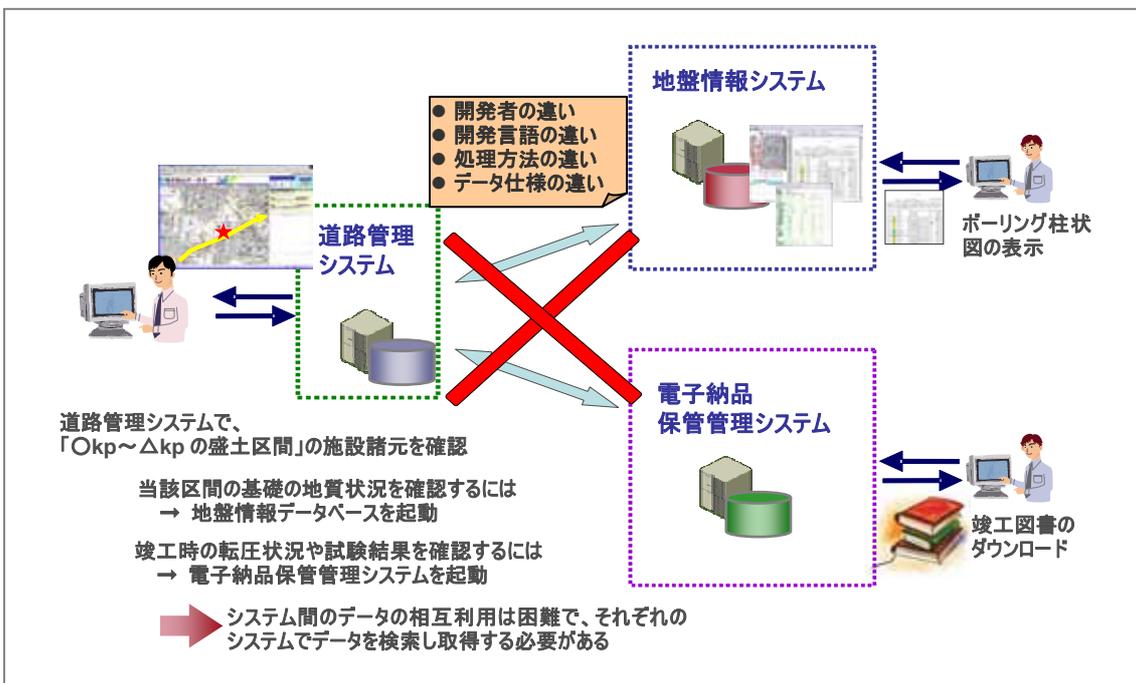


図 1-2 現在のシステム利用の例

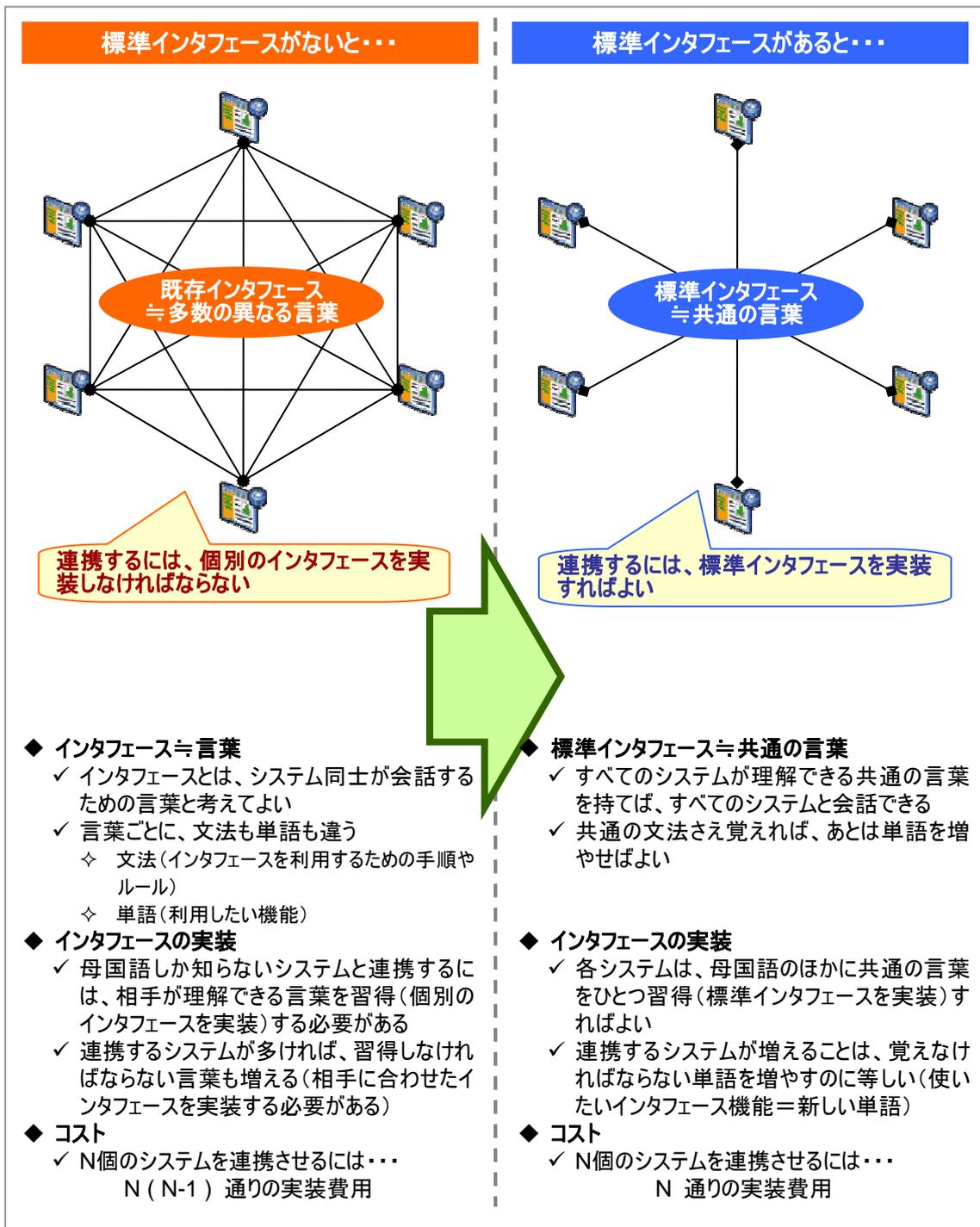


図 1-3 標準インタフェースによる連携のイメージと効果

1-3. 適用範囲

本ガイドラインは、建設情報を取り扱うアプリケーションプログラムの開発及び保守・運用管理に適用する。このうち、データの共用性を考慮し、インターネット/イントラネット環境で動作するアプリケーションプログラムを対象とする。

なお、上記アプリケーションプログラムには、新規に開発するもの、既存のもの(ただし、機能追加や改修の予定があるもの)の両方を含む。

【解説】

インターネット/イントラネット環境で動作するアプリケーションプログラムは、多くの利用者を対象としたものであり、共用性の高いものである。本ガイドラインは、このような共用性の高いアプリケーションプログラムのうち、「建設情報」を取り扱うものに対して適用する。本ガイドラインでは、アプリケーションプログラムを開発・保守・運用管理するにあたっての標準的なルールを定めている。

1-4. 利用者と利用方法

本ガイドラインの利用者は、以下のとおりとする。

- システム開発の企画者
- システム開発の発注者
- システム開発者
- システム管理者

各利用者は、情報システムの企画～運用の各フェーズにおいて、本ガイドラインを参照し、情報システムの管理において自らが担う役割を全うするものとする。

【解説】

本ガイドラインの利用者と利用方法を、表 1-1 の通り定義する。

表 1-1 本ガイドラインの利用者

No.	利用者	役割
1	システム開発の企画者 新規にシステム開発の企画・調整をする立場にある者 稼動中のシステムに対して、システムの機能追加やリニューアルを検討する立場にある者	システム開発の企画者は、ガイドラインの概要や位置づけを把握し、標準インタフェースの考え方を理解した上で、本ガイドラインに従ったシステム開発を企画・検討する
2	システム開発の発注者 システム開発を発注する際の仕様書を作成する立場にある者	システム開発の発注者は、標準インタフェースの考え方を理解し、標準インタフェースの関数仕様を理解した上で、本ガイドラインに従ったシステム開発の発注仕様書を作成する
3	システム開発者 システムを開発する立場にある者(外部委託企業等を含む)	システム開発者は、標準インタフェースの関数仕様を理解した上で、標準インタフェース実装仕様に従ったシステム開発を行う
—	システム利用者 システムを利用する者	本ガイドラインの対象外
4	システム管理者 システムを運用・管理する立場にある者	システム管理者は、標準インタフェースの普及・運用に関する考え方にしたがって、標準インタフェースに対応したシステムの保守・管理・運用を行う

本ガイドラインの利用者は、システム運用の各フェーズにおいて、システムに対して何らかの影響を及ぼす役割を持つ者を想定しており、所謂「システムの利用者」は対象外とする。表 1-1 の内容を図示すると、図 1-4 のようになる。

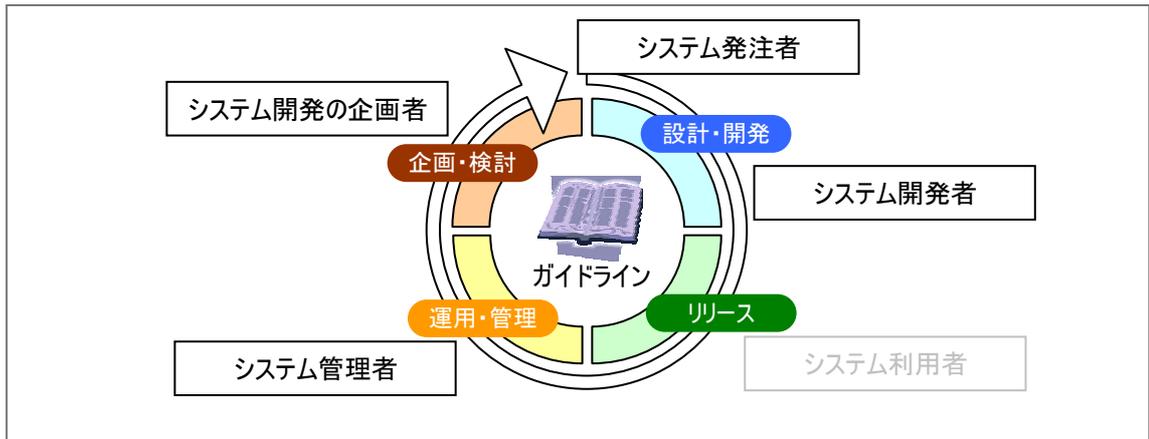


図 1-4 システム運用の各フェーズにおける本ガイドラインの利用者

それぞれの利用対象者が参照すべき本ガイドラインの内容は、表 1-2 に示すとおりである。

表 1-2 利用対象者が参照する本ガイドラインの項目

本ガイドラインの目次	企画者	発注者	開発者	管理者
1.ガイドラインの概要・位置づけ				
1.1.目的	○	○	○	○
1.2.本ガイドライン策定の背景と標準インタフェースの意義	○	○	○	○
1.3.適用範囲	○	○	○	○
1.4.利用者と利用方法	○	○	○	○
1.5.引用規格	○	○	○	○
1.6.用語の定義	○	○	○	○
2.標準インタフェースの考え方				
2.1.本ガイドラインにおける標準インタフェースの位置づけ	○	○	○	○
2.2.標準インタフェースの定義	○	○	○	
2.3.標準インタフェースの種類		○	○	
2.4.標準インタフェースの定義方法		○	○	
2.5.関数仕様の考え方		○	○	
2.6.関数仕様の実装方式		○	○	
2.7.標準インタフェースを利用したアプリケーションプログラムの運用形態		○	○	○
2.8.インタフェース公開・管理の考え方		○	○	○
2.9.インタフェース接続の考え方		○	○	○

本ガイドラインの目次		企画者	発注者	開発者	管理者
3.標準インタフェース関数仕様					
3.1.関数の一覧			○		
3.2.関数の説明			○		
4.標準インタフェース実装仕様					
4.1.POST/GET 利用のための実装仕様			○		
4.2.SOAPとWSDL 利用のための実装仕様			○		
5.関数仕様作成に関する取り決め					
5.1.関数仕様作成に関する取り決め			○		
5.2.関数と引数の命名規則			○		
6.実装仕様作成に関する取り決め					
6.1.実装仕様作成に関する取り決め		○	○	○	
7.標準インタフェースの運用に関する取り決め					
7.1.インタフェース公開・管理方法					○
7.2.インタフェース接続方式					○
8.標準インタフェース公開・管理仕様					
8.1.アプリケーションプログラムの内容に関する公開・管理仕様					○
8.2.アプリケーションプログラムで利用できる関数等の技術的な公開・管理仕様					○
附属書					
1.参考とすべきインタフェース					
1.1.地物単位で地理情報を取得するインタフェース(参考)	○	○	○		
1.2.SOAPによる地理情報等の標準インタフェース(参考)	○	○	○		
2.ガイドラインの適用事例					
2.1.システム実装の事例	○	○	○		
2.2.インタフェース策定の事例	○	○			

1-5. 引用規格

本ガイドラインは、以下に掲げる標準規格を引用する。これにより、本ガイドラインを構成する。

1-5-1. 国際的な標準化団体による標準規格

【W3C : World Wide Web Consortium】

- Simple Object Access Protocol (SOAP) 1.1, W3C Note 08 May 2000
- Web Services Architecture, W3C Working Group Note 11 February 2004
- Web Services Description Language (WSDL) 1.1, W3C Note 15 March 2001

【ISO : International Organization for Standardization】

- ISO19128 Web Map Server interface(WMS)

【OGC : Open Geospaial Consortium, Inc.】

- OGC Web Services Common Specification Version: 1.1.0 with Corrigendum 1 2007-02-09
- OGC Filter Encoding Implementation Specification Version: 1.1.0 2005-05-03
- OGC Catalogue Services Specification Version: 2.0.0 with Corregendum 2005-05-20

【JIS : 日本工業規格】

- JIS X7112 地理識別子による空間参照

【その他】

- Basic Profile Version 1.0, 16 April 2004, The Web Services-Interoperability Organization (WS-I)

1-5-2. 既存の標準インタフェース仕様との関係

本ガイドラインを作成するにあたり、以下に挙げる既存の標準インタフェースを大いに参考とした。また、いくつかの既存標準インタフェースからは、その一部を引用している。具体的な引用箇所を、表 1-3 に示す。

- 減災情報共有プロトコル Ver. 1.00.028s 平成 18 年 9 月
独立行政法人 防災科学技術研究所/独立行政法人 産業技術総合研究所
- 平成 18 年度情報家電活用基盤整備事業（時空間情報利活用システム推進事業）
「PI の標準化と実装検証」 平成 19 年 2 月
財団法人 日本情報処理開発協会 データベース振興センター
- 地理情報共用 Web システム標準インタフェースガイドライン 第 0.03 版 平成 13 年 3 月
国土交通省国土計画局
- 河川 GIS・河川アプリケーション標準インタフェースガイドライン 第 1.0 版
2006 年 6 月 27 日
国土交通省河川局

表 1-3 既存の標準インタフェースからの引用箇所

本ガイドラインの目次	他の既存インタフェースからの引用			
	減災	PI	WMS	河川
1. ガイドラインの概要・位置づけ				
1.1. 目的				
1.2. 本ガイドライン策定の背景と標準インタフェースの意義				
1.3. 適用範囲				
1.4. 利用者と利用方法				
1.5. 引用規格				
1.6. 用語の定義				○
2. 標準インタフェースの考え方				
2.1. 本ガイドラインにおける標準インタフェースの位置づけ				
2.2. 標準インタフェースの定義				○
2.3. 標準インタフェースの種類				
2.4. 標準インタフェースの定義方法				○
2.5. 関数仕様の考え方				○
2.6. 関数仕様の実装方式				○

本ガイドラインの目次	他の既存インタフェースからの引用			
	減災	PI	WMS	河川
2.7.標準インタフェースを利用したアプリケーションプログラムの運用形態				○
2.8.インタフェース公開・管理の考え方				○
2.9.インタフェース接続の考え方				○
3.標準インタフェース関数仕様				
3.1.関数の一覧		△	△	△
3.2.関数の説明		△	△	△
4.標準インタフェース実装仕様				
4.1.POST/GET 利用のための実装仕様		△	△	△
4.2.SOAPとWSDL 利用のための実装仕様		△	△	△
5.関数仕様作成に関する取り決め				
5.1.関数仕様作成に関する取り決め				○
5.2.関数と引数の命名規則				○
6.実装仕様作成に関する取り決め				
6.1.実装仕様作成に関する取り決め				○
7.標準インタフェースの運用に関する取り決め				
7.1.インタフェース公開・管理方法				○
7.2.インタフェース接続方式				○
8.標準インタフェース公開・管理仕様				
8.1.アプリケーションプログラムの内容に関する公開・管理仕様				○
8.2.アプリケーションプログラムで利用できる関数等の技術的な公開・管理仕様				○
附属書				
1.参考とすべきインタフェース				
1.1.地物単位で地理情報を取得するインタフェース(参考)	△			
1.2.SOAPによる地理情報等の標準インタフェース(参考)				△
2.ガイドラインの適用事例				
2.1.システム実装の事例				
2.2.インタフェース策定の事例				△
減災： 減災情報共有プロトコル PI： PIの標準化と実装検証 WMS： 地理情報共有 Web システム標準インタフェースガイドライン 河川： 河川 GIS・河川アプリケーション標準インタフェースガイドライン (○…引用、△…一部引用)				

1-6. 用語の定義

本ガイドラインで利用する用語を、以下のとおり定義する。

(i) アプリケーションプログラム

プログラム言語の要素を論理的に組み合わせ、PC システムで機械、プロセスなどを制御するために必要なプログラムの集合体
(JIS B3500 プログラマブルコントローラ用語)

(ii) インターネット【Internet】

複数のネットワークを相互に接続して構成されるネットワーク。ここでは、イントラネットと区別し、TCP/IP 技術を基礎とした世界中を接続するネットワークの意味で用いる。

(改訂電子情報通信用語辞典：電子情報通信学会編)

(iii) イン트라ネット【Intranet】

インターネット向けに開発された技術やインフラを LAN に適用したシステム。
(情報処理技術者用語辞典：日経 BP 社)

(iv) インタフェース【Interface】

ハードウェアやソフトウェアを相互に接続するための仕様（規格）や取り決め（規約）。

(情報処理技術者用語辞典：日経 BP 社 を基に定義)

(v) Web サービス【Web Service】

インターネットプロトコルをプロトコルとするネットワーク上で、共通の操作で機械間の相互作用ができるように構成されたソフトウェア・システム

(Web Services Architecture :W3C Working Group Note、11 February 2004 を基に定義)

(vi) サービス【service】

アプリケーションなどが所定のインタフェースを通して命令を受けることにより実行される機能で、他から区別して一まとまりものとして認識されるもの

(ISO19119 Geographic information – Services を基に定義)

(vii) クライアント

二つ以上の情報処理機器をネットワークでつないで利用するもののうち、処理を依頼する側。ユーザ（利用者）が操作する端末。

（改訂電子情報通信用語辞典：電子情報通信学会編）

(viii) サーバ

二つ以上の情報処理機器をネットワークでつないで利用するもののうち、共通に利用される専門処理機能をもつもの。

（改訂電子情報通信用語辞典：電子情報通信学会編）

(ix) POST/GET

HTTP プロトコルで規定されているリクエストのメソッドの一部。

(x) 略語

GIS	Geographic Information System (地理情報システム ¹⁾)
HTML	Hypertext Markup Language (ハイパテキストマーク付け言語 ²⁾)
HTTP	Hypertext Transfer Protocol (ハイパテキスト転送プロトコル ³⁾)
ISO	International Organization for Standardization (国際標準化機構 ⁴⁾)
NPO	Nonprofit Organization (行政・企業とは別に社会的活動をする非営利の民間組織 ⁵⁾)
OGC	Open Geospatial Consortium、 Inc.
SOAP	Simple Object Access Protocol (XML によるデータ交換のための通信プロトコル ⁶⁾)
W3C	World Wide Web Consortium (1994 年 10 月に設立された World Wide Web に利用される技術の標準化を進める国際的な非営利団体 ⁶⁾)
WFS	Web Feature Service
WMS	Web Map Server、 Web Map Service
WSDL	Web Service Description Language (Web サービスのサービス仕様を記述 ⁶⁾)
XML	Extensible Markup Language (拡張可能なマーク付け言語 ⁷⁾)

注 ¹⁾ : 地理情報科学事典 : 地理情報システム学会編

注 ²⁾ : JIS X 4156:2000

注 ³⁾ : TS X 0085:2004

注 ⁴⁾ : 改訂電子情報通信用語辞典 : 電子情報通信学会編

注 ⁵⁾ : 広辞苑第五版 : 岩波書店

注 ⁶⁾ : 情報処理技術者用語辞典 : 日経 BP 社

注 ⁷⁾ : JIS X 4159:2002

2. 標準インタフェースの考え方

2-1. 本ガイドラインにおける標準インタフェースの位置づけ

本ガイドラインでいう「標準インタフェース」とは、異なるアプリケーションプログラム間において情報を共有し、新たなサービスを作成するために利用するためのインタフェースである。

本ガイドラインにおける標準インタフェースは、既存の標準インタフェースにおいて策定されている内容から共用性・汎用性の高いものを抽出し、再利用可能となるよう再構成したものと、既存の標準インタフェースでは未検討であるが、多くの分野で応用可能なものを定義している。

【解説】

(1) 既存インタフェースについて

標準インタフェースについては、本ガイドライン以外にも、(独)防災科学技術研究所、(財)日本情報処理開発協会データベース振興センター、国土交通省国土計画局、国土交通省河川局等においても、各機関が関係する分野を対象とした検討が実施されている。

- 減災情報共有プロトコル Ver. 1.00.028s 平成 18 年 9 月
独立行政法人 防災科学技術研究所/独立行政法人 産業技術総合研究所
- 平成 18 年度情報家電活用基盤整備事業（時空間情報利活用システム推進事業）
「PI の標準化と実装検証」 平成 19 年 2 月
財団法人 日本情報処理開発協会 データベース振興センター
- 地理情報共用 Web システム標準インタフェースガイドライン 第 0.03 版 平成 13 年 3 月
国土交通省国土計画局
- 河川 GIS・河川アプリケーション標準インタフェースガイドライン 第 1.0 版
2006 年 6 月 27 日
国土交通省河川局

それぞれの標準インタフェースについて、特徴、概要、適用範囲 / 分野等のカテゴリで整理すると、表 2-1 のようになる。

表 2-1 各標準インタフェースの関係

整理項目	河川 I/F	減災 I/F	地理 Web	PI 標準
適用範囲(対象)	河川分野のデータ取得	防災(減災)分野の情報共有	地理情報全般	PI(Place Identifier: 場所を特定する名前)
記述内容	技術仕様+運用指針等のガイドライン	技術仕様	技術仕様+運用指針等のガイドライン	技術仕様
採用している引用規格(地理情報)	WMS (ISO19128)	WFS (OGC 実装仕様)	WMS (ISO19128)	ISO19112
採用している引用規格(プロトコル)	HTTP、SOAP	SOAP	HTTP	HTTP、SOAP
他の分野への適用可能性	<ul style="list-style-type: none"> ✓ 地理情報に関する I/F ✓ プロトコルの仕様 ✓ 規格引用の考え方 ✓ 拡張の考え方 	<ul style="list-style-type: none"> ✓ 地理情報に関する I/F ✓ プロトコルの仕様 ✓ 規格引用の考え方 	<ul style="list-style-type: none"> ✓ 地理情報に関する I/F ✓ プロトコルの仕様 ✓ 規格引用の考え方 	<ul style="list-style-type: none"> ✓ 地理情報に関する I/F ✓ プロトコルの仕様 ✓ 規格引用の考え方

【表題凡例】	
河川 I/F:	河川 GIS・河川アプリケーション標準インタフェースガイドライン
減災 I/F:	減災情報共有プロトコル
地理 Web:	地理情報共用 Web システムガイドライン
PI 標準:	PI の標準化と実証検証
技術仕様:	通信プロトコルや関数仕様等、インタフェースを利用したアプリケーションを作成するのに必要な技術文書という位置づけ(主に開発者向け)
運用指針:	インタフェースを実際に普及・利用するための考え方や、運用上のルール等の施策までを含んだ文書という位置づけ

(2) 本ガイドラインにおける標準インタフェースの位置づけ

本ガイドラインにおいて策定した標準インタフェースは、図 2-1 に示すように、既存の標準インタフェースのうち、他分野に適用可能な部分に関しては、それを包含し、既存の標準インタフェースでは規定されていない部分に関しては、新たに検討・策定したものである。したがって、既存のインタフェース仕様で定められている内容は尊重し、今後、現在定められていない分野（道路、港湾、下水道など）においても同様の考え方で適用できるようなインタフェース仕様となっている。

		河川 I/F	地理 Web	減災 I/F	PI 標準	etc...							建設 I/F
汎 仕 様	地理情報	○	○	○	○	→	→	→	→	→	→	→	●
	メタデータ検索	※ここでいう「メタデータ検索」とは、クリアリングハウスを利用した横断検索を指す										◎	
	地名辞典				○	→	→	→	→	→	→	→	●
技 術 仕 様		○	○	○	○	→	→	→	→	→	→	→	●
考 え 方 等	運用ルール	○	○	○	○	→	→	→	→	→	→	→	●
	拡張ルール	○	○	○	○	→	→	→	→	→	→	→	●
	開発手法	○	○	○	○	→	→	→	→	→	→	→	●

既存の標準インタフェースは、適用範囲や策定期間が異なること等の理由から、相互に連携して策定されていない状況である。
記述内容には、重複・類似する箇所も存在する。

本件における標準インタフェースは建設情報という広い対象を想定しているため、既存のインタフェースを包括した汎用的な記述となる。

【凡例】

<p>河川 I/F: 河川 GIS・河川アプリケーション標準インタフェースガイドライン</p> <p>減災 I/F: 減災情報共有プロトコル</p> <p>地理 Web: 地理情報共用 Web システムガイドライン</p> <p>PI 標準: PI の標準化と実証検証</p> <p>建設 I/F: 本件における標準インタフェース</p>	<p>既存 I/F で定義されている事項に関しては、整理・統合が可能なものは整理し、既存 I/F のものをベースに策定。 ○⇒●</p> <p>既存 I/F で定義されていないものは、新規に策定。 ⇒◎</p>
--	---

図 2-1 既存の標準インタフェースと本件の標準インタフェースの策定内容

(3) 本ガイドラインにおける標準インターフェースと今後の利用方法

本ガイドラインで定める標準インターフェースは、以下の3種類である。

- 1) 標準インターフェース導入の考え方等、分野を問わない普遍的な概念
- 2) 共用性の高い技術仕様
- 3) 汎用性の高いインターフェース仕様

これらは、様々な建設情報を連携して利用する上で必要な事項を定めたものであるが、個々の内容についてはすでに一般的な地理空間情報を利用するために定められたものである。

本ガイドラインにおいて汎用的なインターフェースを策定したことにより、上記に示したような異なるインターフェース間の重複は解消される。各応用分野では図 2-2 に示すように、本ガイドラインにおいて策定した標準インターフェースをベースとし、特定分野の利用に特化したインターフェースを定めることができる。

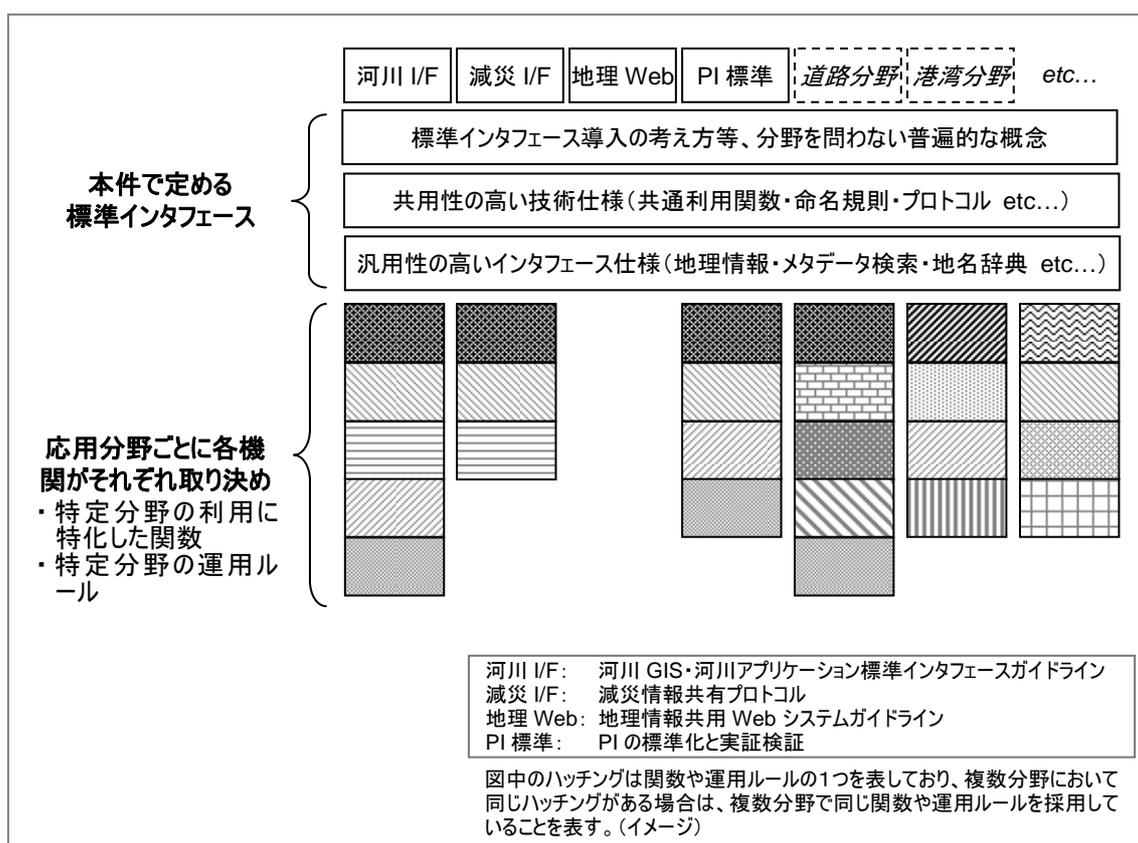


図 2-2 本ガイドラインの標準インターフェースと他の標準インターフェースの関係

(4) 本ガイドラインの位置づけと各分野のメリット

以上のように、本ガイドラインにおいて定義する標準インタフェースとは、特定の分野ではなく、建設情報全般のシステム開発において汎用的に利用可能なものである。以下に、本ガイドラインの位置づけ、および各分野に本ガイドラインを適用した際のメリットを整理した。

【本ガイドラインで定める標準インタフェース】

- 地理情報、メタデータ検索、地名辞典等、汎用性の高い部分を策定
- 標準インタフェースの考え方等、適用分野に左右されない普遍的な概念を定義

【他の分野における標準インタフェース】

- 本ガイドラインの標準インタフェースをベースとし、応用分野ごとに必要な部分を策定
- 河川、減災等の既存分野だけでなく、道路、港湾等の分野へも展開予定

【各分野におけるメリット】

- 各分野のシステムから、他分野のシステムの持つ情報を収集・利用することが可能となる
 - ◇ 他分野の最新の情報を直接入手でき、データの受け渡しが効率化する
- 標準インタフェースの運用について、各々の分野に特化した関数や運用ルールのみを管理(維持・更新)すればよく、それぞれの管理対象を明確化できる
 - ◇ 分野横断的な部分のメンテナンスにかかる費用・時間が削減できる

2-2. 標準インターフェースの定義

本ガイドラインで適用するインターフェースとは、インターネット/イントラネット環境で動作するアプリケーションプログラム間を相互接続するための仕様と取り決めを指す。

【解説】

本ガイドラインで適用するインターフェースは、関数と実装方式だけを取り決めるものではなく、それを運用するときのルールも含む。本ガイドラインでは、表 2-2 で “ ” 印をつけた項目について、仕様と取り決めを定める。

表 2-2 本ガイドラインで定めるインターフェースに関する仕様と取り決め

内容		仕様	取り決め
インターフェース	関数について	○	○
	実装方式について	○	○
	接続方式について	△	○
運用	公開・管理について	○	○

○:本ガイドラインで定める項目

△:すでにある仕様・取り決めを採用する項目

本ガイドラインでは、標準インターフェース作成に関する取り決めとして、「関数仕様作成に関する取り決め」、「実装仕様作成に関する取り決め」、「接続方式に関する取り決め」、「運用に関する取り決め」を定める。それらの取り決めに基づき、「関数仕様」「実装仕様」「公開・管理仕様」を定める。

2-3. 標準インターフェースの種類

本件における標準インターフェースは、その機能によって以下の3種類を定義する。

- (1) 地理情報に関するインターフェース
- (2) 地名辞典に関するインターフェース
- (3) メタデータ検索に関するインターフェース

【解説】

本件における標準インターフェースは、その機能によって以下に大分できる。それぞれの種類について概要を説明する

(1) 地理情報に関するインターフェース

ここでいう「地理情報に関するインターフェース」とは、建設情報を共有する場面を想定したとき、地理情報に関するサービスとして以下の機能を実現するために必要なインターフェースである。

表 2-3 地理情報に関するインターフェース

No.	機能	具体例
1	地図を表示する(座標指定)	地名辞典で地名を検索し、地名に該当する座標値を得たので、その座標値を基に地図上で該当位置を表示させる
2	地図を表示する(縮尺指定)	適切な表示縮尺で地図を表示させる
3	地図を表示する(表示する地物を指定)	検索結果の地物をハイライト表示させたり、表示する地図に含まれる地物を指定する
4	地図を移動させる	地図を移動させる(スクロールバー、ドラッグ&ドロップ、8方向ボタン等で実現)
5	地図の凡例を表示する	表示されている地図に含まれる地物の凡例を表示する

(2) 地名辞典に関するインタフェース

ここでいう「地名辞典に関するインタフェース」とは、建設情報を共有する場面を想定したとき、地名辞典に関するサービスとして以下の機能を実現するために必要なインタフェースである。

表 2-4 地名辞典に関するインタフェース

No.	機能	具体例
1	地名(地理識別子)を基に、地名に該当する座標を検索(地理識別子と座標の変換)	「赤坂」「東京タワー」「日本武道館」等の地名(地理識別子)にあたるキーワードを基にし、地名に該当する座標を検索する
2	座標を基に、付近の地名(地理識別子)を検索(座標と地理識別子の変換)	GIS 等で取得した座標や座標範囲の値から、座標値の近傍や座標範囲内に含まれる地名(地理識別子)を検索する
3	地名(地理識別子)を基に、地名に関連する別の地名を検索	「千代田区」という地名をキーワードとし、千代田区の下位住所の検索(「神田」や「九段南」など)

通常は、上記の処理を行った後、「地図上で位置を表示する」等の処理を行うが、地名辞典が提供するサービスではなく、地名辞典が実現すべきインタフェースには該当しない為、その後の処理に関してはここでは定義しない

(3) メタデータ検索に関するインタフェース

ここでいう「メタデータ検索に関するインタフェース」とは、建設情報を共有する場面を想定したとき、メタデータ検索に関するサービスとして以下の機能を実現するために必要なインタフェースである。

メタデータ検索とは、特定のデータベースや特定の情報を検索するということではなく、様々な情報がメタデータとして整備され、クリアリングハウスのような仕組みにより、横断的にメタデータの検索を実現する仕組みのことを指す。

表 2-5 メタデータ検索に関するインタフェース

No.	機能	具体例
1	建設情報の検索(キーワード)	「スーパー堤防」「舗装」「環境」等のキーワードを利用し、キーワードに関連する建設情報(電子納品成果)を検索する
2	建設情報の検索(地理的範囲)	上記に加え、地図のインタフェース等を組み合わせ、地理的範囲(座標範囲)も考慮した検索を行う
3	検索結果の表示(候補一覧表示)	キーワードや地理的範囲に該当する建設情報の候補一覧を表示する
4	検索結果の表示(メタデータの表示)	検索結果の候補一覧を1件ずつ選択することで、それぞれのメタデータを確認できる。

2-4. 標準インターフェースの定義方法

標準インターフェースの定義方法を以下に示す。

- (1) 建設情報の連携を実現するうえで必要とされる機能を抽出する。
- (2) 同種の機能をとりまとめ、関数を整理する。
- (3) ISO で提示されているインターフェースについては、それを採用する。
- (4) ISO で提示されているインターフェースだけでは不十分と判断したものについては、独自のインターフェースを定義する。
- (5) 既存の標準インターフェースで、該当する機能が規定されているかどうかを調査し、既存の標準インターフェースで取り扱っている機能は、それを引用する。

【解説】

上記の定義方法を定義するにあたり、基本的な考え方としては、以下の考え方に基づき、既存の標準インターフェースや ISO 等の標準規格を最大限に利用するものとする。

- ISO あるいは JIS による規格を「引用すべき規格」と位置づけ、インターフェースとして定められている既存の標準規格が存在するものに関しては、それを採用する
- ISO 等の標準化機関が策定する標準規格には、「正式な版」と「検討中の版」が存在する場合がある。検討中の版に関しては、今後細部の仕様が変更になる可能性もあるため、引用する際は可能な限り正式な版を採用することとする
- 同様に、ISO や JIS に次ぐ標準化機関として、OGC があるが、ISO ほどの強制力を持たない「業界主導の実装仕様」であることから、該当するインターフェースの規格が ISO で存在しない場合に、OGC の実装仕様を採用することとする

2-5. 関数仕様の考え方

- (1) 標準インタフェースの関数仕様は、本ガイドラインにおいて定める。アプリケーションプログラムの開発・提供を行う者で、本ガイドラインに定めた機能に該当するサービスを開発・提供する場合は、本ガイドラインで定めた関数仕様に従わなければならない。
- (2) 本ガイドラインで定める関数仕様に該当しない関数を用いる必要が生じた場合は、本ガイドラインで定める取り決めに従い、アプリケーションプログラムの開発・提供者が関数仕様を作成する。

【解説】

アプリケーションプログラムの開発・提供者は、開発・提供しようとするアプリケーションプログラムのサービスが、本ガイドラインで定めた関数仕様に該当する場合には、本ガイドラインで定めた関数仕様に従いアプリケーションプログラムの開発・提供を行わなければならない。

開発・提供しようとするアプリケーションプログラムのサービスが、本ガイドラインで定める関数仕様に該当しない場合には、アプリケーションプログラムの開発・提供者は、本ガイドライン第5章・第6章で定める取り決めに従い、関数仕様と実装仕様を作成しなければならない。また作成した関数仕様と実装仕様は、他のアプリケーションプログラムからの利用も可能とする必要があるため、本ガイドライン第8章により公開しなければならない。

2-6. 関数仕様の実装方式

標準インタフェース関数仕様の実装方式として、以下の2種類を定める。

方式1) HTTP による POST/GET を利用した実装

方式2) HTTP による SOAP と WSDL を利用した実装

【解説】

(1) HTTP による POST/GET を利用した実装方式について

標準インタフェース関数仕様の実装方式は、接続方式を考慮する必要がある。アプリケーションプログラム間での通信プロトコルは、インターネット/イントラネット環境に一般的に利用されている HTTP を採用し、接続方式としてはさらに HTTP による POST/GET の2種類の接続方式を採用する。したがって、それぞれの接続方式に対応する実装方式を定める。HTTP のバージョンは 1.0 または 1.1 とするが、推奨は 1.1 とする。

(2) HTTP による SOAP と WSDL を利用した実装方式について

アプリケーションプログラムにおいて種々の機能を実現するためには、(1) の HTTP による POST/GET を利用した接続方式だけでは不十分な場合がある。その場合は、HTTP による SOAP と WSDL を利用した実装方式を採用する。

たとえば、地理情報に関するインタフェースは、ISO19128 を採用することで、方式1の接続方式となるが、地名辞典に関するインタフェースやメタデータ検索に関するインタフェースに関しては、方式2の SOAP と WSDL を利用した接続方式を採用している。

方式2の実装方式の策定にあたっては、OGC における実装仕様を参考にしつつ、OGC が仕様策定時に参照した上位の標準技術 (W3C における Web サービス等) を活用している。

2-7. 標準インターフェースを利用したアプリケーションプログラムの運用形態

標準インターフェースによるアプリケーションプログラムは、次に示す運用形態を想定する。

- (1) 国(各省庁内)や各地方公共団体内など、単一の機関内における運用
- (2) 国の研究機関など省庁をまたいだ機関も含めた運用
- (3) 大学、NPO、民間企業など、一般利用も含めた運用

【解説】

標準インターフェースに準拠したアプリケーションプログラムは、「建設情報」を取り扱うあらゆる機関での運用を想定する。

建設情報を取り扱う機関の代表的なものは国や地方公共団体であり、まずは単一の機関における業務アプリケーションプログラムでの運用を想定する。将来的には、国、地方公共団体、教育・研究機関、NPO、民間企業等の一般の機関にまでその運用範囲が広がる。

このような運用形態を踏まえ、本ガイドラインでは以下の内容を定める。

- インタフェースの公開、管理に関する仕様・取り決め
- インタフェースの接続に関する仕様・取り決め

2-8. インタフェース公開・管理の考え方

(1) インタフェースの公開

- アプリケーションプログラムは、本ガイドラインで定めた関数仕様、実装仕様に基づくインタフェースを公開する。公開したインタフェースを介して、複数の異なるアプリケーションプログラムに対してサービスを提供する。
- 逆にアプリケーションプログラムは、公開されたインタフェースを介して、複数の異なるアプリケーションプログラムに対して、サービスの提供を依頼する。

(2) インタフェースの管理

- アプリケーションプログラムを管理するため、アプリケーションプログラムの開発者は、公開するインタフェースの関数仕様、実装仕様、ならびにアプリケーションプログラムが提供するサービスの内容を文書として作成し、アプリケーションプログラムの管理者に提出する。

【解説】

(1) インタフェースの公開

製品として提供・販売されている GIS 等アプリケーションプログラムのなかには、インタフェース仕様を公開しているものがあり、アプリケーションプログラムの開発者は、公開されたインタフェース仕様に従い、その GIS 等アプリケーションプログラムを利用するアプリケーションプログラムの開発を行うことができる。このとき、インタフェース仕様を公開しているアプリケーションプログラムのことを、特に「サービス」と位置づける。

ただし、そこで公開されているインタフェース仕様は、そのアプリケーションプログラムにのみ適用される独自の仕様であり、たとえそのアプリケーションプログラムと同種の機能を提供するアプリケーションプログラムがあったとしても、機関 A が開発したアプリケーションプログラムが、機関 B のアプリケーションプログラムを利用することは不可能であった。この結果、同様のサービスを提供するアプリケーションプログラムであっても、製品によってインタフェースが異なるためにその都度アプリケーションプログラムを開発するという非効率な状況にあった。このような非効率な状況を避けるため、同様のサービスを提供する場合は、本ガイドラインに基づく共通のインタフェース仕様に基づいた実装を行うものとする。

(2) インタフェースの管理

標準インタフェースが公開され、アプリケーションプログラム間を自由に接続できるようになったとしても、どこにどのようなインタフェースを提供するサービスがあるの

か分からなければ、そのインタフェースを利用することはできない。また、サービスの存在を発見できたとしても、公開されているインタフェースの仕様が不明な場合、インタフェースへの接続はできず、利用することができない。

このような課題を克服するためには、どこにどのようなインタフェースを提供するサービスが存在しているかという情報と、そのインタフェースに接続するためのインタフェース仕様を管理する必要がある。その方法のひとつとして、アプリケーションプログラム管理サーバを設け運用する方法が考えられる。サービスの管理とそれに基づいたインタフェースへの接続の概念図を、図 2-3 に示す。サービスを管理する単位としては、事務所単位、地方整備局単位のほか、将来的には外部機関による管理も考えられる。

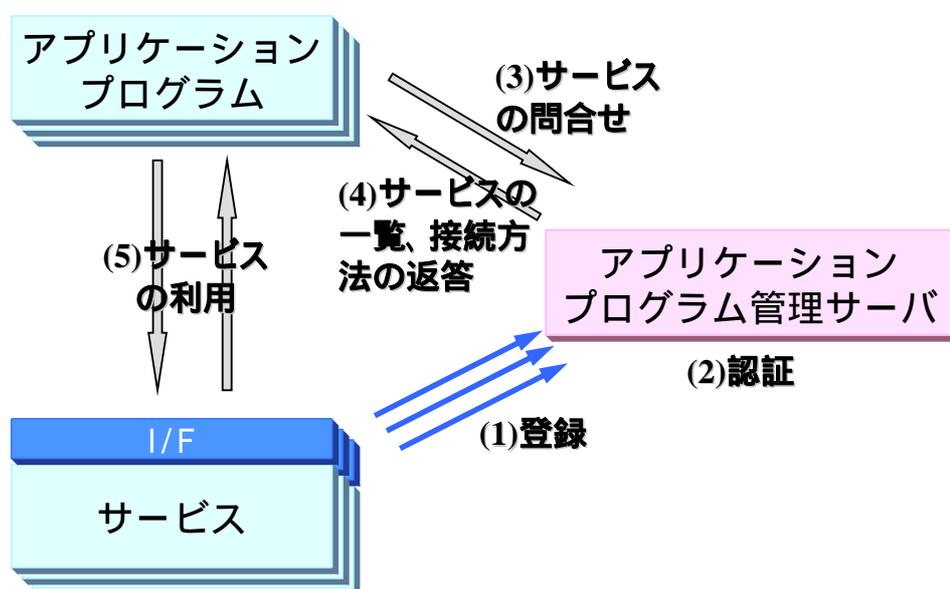


図 2-3 インタフェース管理方法概念図（参考）

2-9. インタフェース接続の考え方

アプリケーションプログラム間での接続は、インタフェースを介して行う。

アプリケーションプログラム間での接続に用いる通信プロトコルは、HTTP とし、以下のうちいずれかの方式とする。

方式1) POST/GET により行う場合

方式2) SOAP により行う場合

アプリケーション間で接続を行う場合には、基本的に以下の手順で行う。

- (1) アプリケーションプログラムとサービスを提供するアプリケーションプログラム間の接続認証
- (2) アプリケーションプログラムによる具体的な命令の送信
- (3) サービスを提供するアプリケーションプログラムによる命令に対する処理結果の送信

【解説】

アプリケーションプログラム間での接続に用いる通信プロトコルは、インターネット/イントラネット環境に一般的に利用されている HTTP を採用する。

HTTP による通信では、HTML 文書の表示等、一般的にクライアントからサーバへの要求に POST/GET メソッドを使う方法が利用されている。また Web サービスにおいても、HTTP などにより XML 形式のデータをやり取りする SOAP の利用が現在の主流である。

そこで本ガイドラインでは、一般的に利用されている POST/GET による接続方式(方式1)と、Web サービスを念頭に、SOAP による接続方式(方式2)の2種類を採用する。

アプリケーションプログラム間では、基本的には次に示す手順に則り接続を行う(図2-4)。

(1) アプリケーションプログラムとサービスを提供するアプリケーションプログラム間の接続認証

- アプリケーションプログラムから、サービスを提供するアプリケーションプログラムに対して、接続の依頼を行う。
- サービスを提供するアプリケーションプログラムは、それに対して接続の認証を行う。
- サービスを提供するアプリケーションプログラムは、認証結果を問合せしたアプリケーションプログラムに返す。

(2) アプリケーションプログラムによる具体的な命令の送信

- アプリケーションプログラムから、サービスを提供するアプリケーション

プログラムに対して、サービスに対する具体的な命令（関数、引数、戻り値の型）を送信する。

(3) サービスを提供するアプリケーションプログラムによる命令に対する処理結果の送信

- サービスを提供するアプリケーションプログラムは、命令に対する処理を行い、その結果について処理を依頼したアプリケーションプログラムに返す。

あるアプリケーションプログラム（サービス利用側）から、処理の命令を受け取ったサービス提供側のアプリケーションプログラムは、その命令が接続認証されたアプリケーションプログラムからのものなのかどうか、どのアプリケーションプログラムからの命令で結果をどこに返すのかを把握しておかなければならない。したがって接続においては、(a)認証に関する情報と、(b)命令の内容や命令に対する処理結果に関する情報により通信を行う。

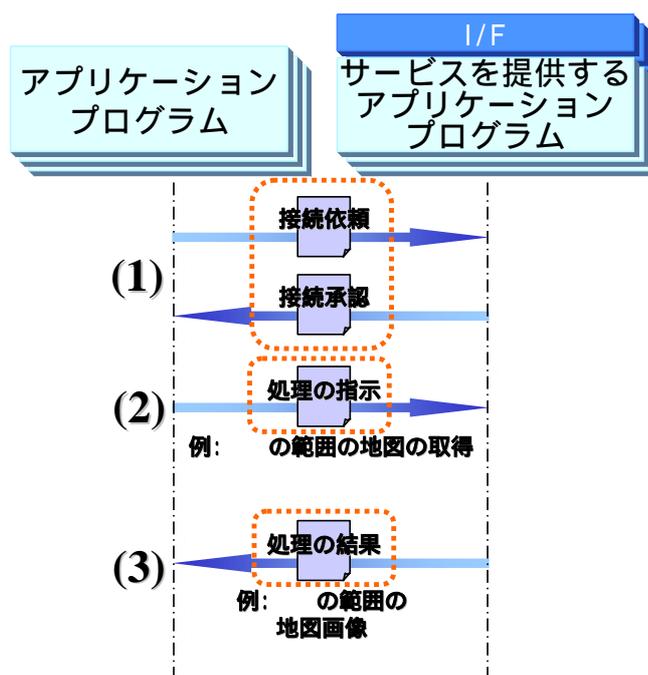


図 2-4 インタフェース接続の手順

SOAP による接続方式（方式 2）においては、SOAP の仕様の解釈の違いから異なるプラットフォームで実装されたインタフェースが接続できないことがあることが知られている。そこで、本ガイドラインでは、相互運用のための指針として、WS-I Basic Profile 1.0 を適用し、実装プラットフォームに依存しないインタフェースを定義する。

3. 標準インタフェース関数仕様

地名辞典サービス、地図サービス、検索サービス等を連携する上で必要となるインタフェースについて、「関数仕様」を定め、解説する。

関数仕様では、以下の項目を定める。

- 関数名
- 戻り値
- 機能
- 関連項目
- 引数
- 説明

それぞれのインタフェースにおいて、必要となる関数は、以下のとおりである。

- A) 提供するサービスの内容を取得するための関数
- B) 情報を検索するための関数
- C) 情報を取得するための関数
- D) 情報を登録するための関数

3-1. 関数の一覧

以下に、本ガイドラインで定義する関数の一覧を示す。

表 3-1 本ガイドラインで定義する関数の一覧

インタフェース種類	種別	必要な機能	機能を実現するための関数
共通	A	サービスメタデータの取得	GetCapabilities
地理情報	B	地図画像の取得	GetMap
	B	(地物の属性情報の取得)	(GetFeatureInfo)
地名辞典 (空間参照系管理)	B	地理識別子による空間参照系の検索	SearchLRS
	C	地理識別子による空間参照系の取得	GetLRS
	D	(地理識別子による空間参照系を新規登録する)	(AddLRS)
	D	(地理識別子による空間参照系を更新する)	(UpdateLRS)
	D	(地理識別子による空間参照系を削除する)	(DeleteLRS)
	B	場所型の検索	SearchLocationType
	C	場所型の取得	GetLocationType
	D	(場所型の新規登録)	(AddLocationType)
	D	(場所型の更新)	(UpdateLocationType)
	D	(場所型の削除)	(DeleteLocationType)
	B	場所型関連の検索	SearchLocationTypeAssociation
	D	(場所型関連の新規登録)	(AddLocationTypeAssociation)
	D	(場所型関連の削除)	(DeleteLocationTypeAssociation)
	地名辞典 (地名辞典管理)	D	(地名辞典の新規登録)
D		(地名辞典の更新)	(UpdateGazetteer)
D		(地名辞典の削除)	(DeleteGazetteer)
B		場所インスタンスの PI 部分の検索	SearchPI
C		場所インスタンスの取得	GetLocationInstance
D		(場所インスタンスの新規登録)	(AddLocationInstance)

インタフェース種類	種別	必要な機能	機能を実現するための関数
	<i>D</i>	<i>(場所インスタンスの更新)</i>	<i>(UpdateLocationInstance)</i>
	<i>D</i>	<i>(場所インスタンスの削除)</i>	<i>(DeleteLocationInstance)</i>
	B	複数場所インスタンス関連の検索	SearchLocationInstanceAssociation
	<i>D</i>	<i>(場所インスタンス関連の新規登録)</i>	<i>(AddLocationInstanceAssociation)</i>
	<i>D</i>	<i>(場所インスタンス関連の削除)</i>	<i>(DeleteLocationInstanceAssociation)</i>
メタデータ検索	C	メタデータで使われる型定義の取得	DescribeRecord
	<i>C</i>	<i>メタデータの属性の定義域の取得</i>	<i>(GetDomain)</i>
	B	検索結果一覧の取得	GetRecords
	C	IDによる検索結果の取得	GetRecordById
	<i>D</i>	<i>メタデータの登録、更新、削除</i>	<i>(Transaction)</i>
	<i>C</i>	<i>メタデータ情報の収集</i>	<i>(Harvest)</i>
<p>【種別凡例】</p> <p>A: 提供するサービスの内容を取得するための関数</p> <p>B: 情報を検索するための関数</p> <p>C: 情報を取得するための関数</p> <p>D: 情報を登録するための関数</p>			

※ 表中のカッコ (斜体) はオプション扱いの関数であり、実装は必須ではない。それ以外の関数に関しては、当該インタフェースを実装する場合は実装を必須とした。本ガイドラインでは、上記に挙げた全ての関数に関して関数仕様を定義したが、実装仕様に関しては必須扱いの関数のみの提示とした。

3-2. 関数の説明

以下より、表 3-1 で示した各関数の概要を示す。

3-2-1. 共通

3-2-1-1. GetCapabilities

機能	サービスメタデータを取得する	
引数	型	名前
	String	version
	説明	
	準拠する標準インタフェースのバージョン番号 該当するバージョンのサービスメタデータのみを取得する 指定なしの場合は、最新バージョン	
	String	service
	サービスタイプ 要求するサービスの識別子 例) 地理情報 “WMS” 地名辞典(空間参照系管理) “SRS” 地名辞典(地名辞典管理) “Gazetteer” メタデータ検索 “CSW”	
String	request	要求名 ”GetCapabilities”でなければならない
String	format	サービスメタデータの出力フォーマット 応答をどのようなフォーマット(MIME Type)で返すかを指定する 省略された場合は、”text/xml”とみなされる
String	updateSequence	サービスメタデータの連番 サービス呼び出し側がこの連番に該当するサービスメタデータをキャッシュしている場合に指定する
戻り値	各 ServiceMetadata	service の内容に応じたサービスメタデータ
関連項目	各 ServiceMetadata 構造体 (※引数”service”において指定したサービスタイプにより、構造体の内容は異なる)	
	サービスタイプ	サービスメタデータ
	WMS	WMSServiceMetadata
	SRS	SRSServiceMetadata
	Gazetteer	GazetteerServiceMetadata
	CSW	CSWServiceMetadata

3-2-2. 地理情報に関するインターフェース

3-2-2-1. GetMap

機能	地図画像を取得する		
引数	型	名前	説明
	String	version	準拠する標準インターフェースのバージョン番号 指定なしの場合は、最新バージョン
	String	request	要求名 ”GetMap”を用いなければならない
	String[]	layers	描画レイヤのリスト
	String[]	styles	描画スタイルのリスト
	String	crs	座標参照系
	Integer[]	bBox	座標範囲
	Integer	width	画像の幅
	Integer	height	画像の高さ
	String	format	画像フォーマット image/jpeg、image/png などの MIME タイプ
	Boolean	transparent	背景透明フラグ true: 透過する false: 透過しない
	String	bgColor	背景色 0x で始まる RGB の 16 進数表現
	String	exceptions	例外報告フォーマット サービスメタデータに記述されているフォーマットの なかから、例外報告フォーマットを指定できる 省略した場合、XML となる
	DateTime	time	時間範囲
Integer	elevation	目的のレイヤの標高 サービスメタデータに有効な標高が記述されている 場合に記述することができる	
Integer	dimension	該当する他の次元値 サービスメタデータに有効な次元が記述されている 場合に記述することができる	
戻り値	地図画像	引数”format”で指定した MIME タイプの画像	
関連項目			

3-2-2-2. GetFeatureInfo

機能	地物の属性情報を表示する		
引数	型	名前	説明
	String	version	準拠する標準インタフェースのバージョン番号 指定なしの場合は、最新バージョン
	String	request	要求名 ”GetFeatureInfo”を用いなければならない
	GetMap 要求 のパラメータ	—	直前の GetMap 要求で送信したパラメータのうち、version と request を除くすべて
	String[]	query_layers	問合せ対象の一つ以上のレイヤのリスト
	String	info_format	地図情報の出力フォーマット サービスメタデータに記述されている”text/xml”などの MIME タイプ
	Integer	feature_count	情報を取得する対象の地物の個数
	Integer	i	地物の水平方向のピクセル単位の座標値
	Integer	j	地物の垂直方向のピクセル単位の座標値
	String	exceptions	例外報告フォーマット サービスメタデータに記述されているフォーマットのなかから、例外報告フォーマットを指定できる 省略した場合、XML となる
戻り値	地物の情報	info_format で指定した形式で符号化された地物の属性情報	
関連項目			

3-2-3. 地名辞典に関するインタフェース

3-2-3-1. 空間参照系管理

1) SearchLRS

機能	地理識別子による空間参照系を検索する	
引数	型	名前
	String	name
	GeographicExtent	domainOfValidity
	String	theme
	String	overallOwner
	String[]	locationType
	AuthResult	authResult
説明	名称 地理的範囲 分類 管理者 関連する場所型の名称 複数指定された場合は AND 検索を行う 認証結果 ユーザ認証を必要としないサービスの場合は null を指定する	
戻り値	LRS[]	
説明	空間参照系が 1 件もヒットしなかった場合は要素数 0 の配列を返す。	
関連項目	GeographicExtent 構造体、LRS 構造体、AuthResult 構造体	

2) GetLRS

機能	地理識別子による地理識別子による空間参照系を取得する	
引数	型	名前
	String	name
	AuthResult	authResult
説明	名称 認証結果 ユーザ認証を必要としないサービスの場合は null を指定する	
戻り値	LRS	
説明	指定した名称の空間参照系が存在しない場合は null を返す。	
関連項目	LRS 構造体、AuthResult 構造体	

3) AddLRS

機能	地理識別子による空間参照系を新規登録する	
引数	型	名前
	LRS	lrs
	String[]	locationType
	AuthResult	authResult
	SRSEexamResult	examResult
説明	登録対象の空間参照系が地理的境界またはポリゴンを持つ場合、その空間参照系は登録済空間参照系の名称である必要がある。 処理に失敗した場合、空間参照系データベースを本関数呼び出し前の状態に戻すこと。	
戻り値	boolean	
関連項目	LRS 構造体、AuthResult 構造体、SRSEexamResult 構造体	

4) UpdateLRS

機能	地理識別子による空間参照系を更新する	
引数	型	名前
	LRS	lrs
	String[]	locationType
	AuthResult	authResult
	SRSEexamResult	examResult
説明	更新対象の空間参照系が地理的境界またはポリゴンを持つ場合、その空間参照系は登録済空間参照系の名称である必要がある。 処理に失敗した場合、空間参照系データベースを本関数呼び出し前の状態に戻すこと。	
戻り値	boolean	
関連項目	LRS 構造体、AuthResult 構造体、SRSEexamResult 構造体	

5) DeleteLRS

機能	地理識別子による空間参照系を削除する	
引数	型	名前
	String[]	lrs
	AuthResult	authResult
戻り値	boolean	
説明	1 件でも空間参照系の削除に失敗した場合は false を返す。 処理に失敗した場合、空間参照系データベースを本関数呼び出し前の状態に戻すこと。	
関連項目	AuthResult 構造体	

6) SearchLocationType

機能	登録されている場所型を検索する		
引数	型	名前	
	String	name	
			名称 場所型の名称を指定する。一部のみの指定も可能である。名称を特定しない場合は null を指定する。
	String	theme	
			分類 一部のみの指定も可能である。分類を特定しない場合は null を指定する
	PiType	identification	
			PI タイプ PI タイプの各要素の値は一部のみの指定が可能
	GeographicExtent	territoryOfUse	
			地理的範囲
String	owner		
		管理者	
String[]	lrs		
		関連する空間参照系の名称 空間参照系を限定しない場合は null を指定する 複数した場合は OR 検索を行う。	
String[]	parentType		
		親の場所型の名称 親関連の条件を指定しない場合は null を指定する	
String[]	childType		
		子の場所型の名称 子関連の条件を指定しない場合は null を指定する	
AuthResult	authResult		
		認証結果 ユーザ認証を必要としないサービスの場合は null を指定する	

戻り値	LocationType[]
説明	<p>lrs に地理識別子による空間参照系の名称を指定することで、特定の空間参照系に関連する場所型を検索することが可能である。</p> <p>また、場所型の Nesting 関連の条件を指定する場合は、parentType または childType に関連する場所型の名称を指定する。</p> <p>地理的範囲が指定された場合は、場所型が持つ地理的範囲と一部でも重なるものを返す。</p> <p>条件に合致する場所型が存在しない場合にはサイズ 0 の配列を返す。</p>
関連項目	PIType 構造体、GeographicExtent 構造体、LocationType 構造体、AuthResult 構造体

7) GetLocationType

機能	登録されている場所型を取得する	
引数	型	名前
	String[]	name
	AuthResult	authResult
		<p>認証結果</p> <p>ユーザ認証を必要としないサービスの場合は null を指定する</p>
戻り値	LocationType[]	
説明	条件に合致する場所型が存在しない場合にはサイズ 0 の配列を返す。	
関連項目	LocationType 構造体、AuthResult 構造体	

8) AddLocationType

機能	場所型を新規登録する	
引数	型	名前
	LocationType	locationType
	AuthResult	authResult
	LocationType-ExamResult	examResult
		<p>登録対象</p> <p>null 指定不可</p>
		<p>認証結果</p>
		<p>審査結果</p>
戻り値	boolean	
説明	処理に失敗した場合、空間参照系データベースを本関数呼び出し前の状態に戻すこと。	
関連項目	LocationType 構造体、AuthResult 構造体、LocationTypeExamResult 構造体	

9) UpdateLocationType

機能	登録されている場所型を更新する	
引数	型	名前
	LocationType	locationType
	AuthResult	authResult
	LocationType-ExamResult	examResult
	更新対象 登録済みの場所型であること。 null 指定不可	
	認証結果	
	審査結果	
戻り値	boolean	
説明	処理に失敗した場合、空間参照系データベースを本関数呼び出し前の状態に戻すこと。	
関連項目	LocationType 構造体、AuthResult 構造体、LocationTypeExamResult 構造体	

10) DeleteLocationType

機能	登録されている場所型を削除する	
引数	型	名前
	String[]	name
	AuthResult	authResult
	削除対象の場所型の名称 登録済みの場所型であること。	
	認証結果	
戻り値	boolean	
説明	処理に失敗した場合、空間参照系データベースを本関数呼び出し前の状態に戻すこと。	
関連項目	AuthResult 構造体	

11) SearchLocationTypeAssociation

機能	登録されている場所型関連を検索する。	
引数	型	名前
	String	childLocationType
	String	parentLocationType
	AuthResult	authResult
		説明
		子場所型の名称
		親場所型の名称
		認証結果
		ユーザ認証を必要としないサービスの場合は null を指定する
戻り値	LocationTypeAssociation[]	
説明	子場所型を指定しない場合は親場所型で指定された場所型のすべての子場所型を返す。 親場所型を指定しない場合は子場所型で指定された場所型のすべての親場所型を返す。	
関連項目	LocationTypeAssociation 構造体、AuthResult 構造体	

12) AddLocationTypeAssociation

機能	場所型関連を新規に登録する。	
引数	型	名前
	LocationTypeAssociation[]	association
	AuthResult	authResult
		説明
		登録対象
		認証結果
		ユーザ認証を必要としないサービスの場合は null を指定する
戻り値	boolean	
説明	1件でも登録に失敗した場合には例外 UpdateFailure を発生する。 処理に失敗した場合、空間参照系管理ベースを本関数呼び出し前の状態に戻すこと。	
関連項目	LocationTypeAssociation 構造体、AuthResult 構造体	

13) DeleteLocationTypeAssociation

機能	場所型関連を削除する。		
引数	型	名前	説明
	LocationTypeAssociation	association	削除対象
	AuthResult	authResult	認証結果 ユーザ認証を必要としないサービス の場合は null を指定する
戻り値	boolean		
説明	削除に失敗した場合は false を返す。 処理に失敗した場合、空間参照系管理サービスを本関数呼び出し前の状態に戻す こと。		
関連項目	LocationTypeAssociation 構造体、AuthResult 構造体		

3-2-3-2. 地名辞典管理

1) AddGazetteer

機能	地名辞典を新規登録する	
引数	型	名前
	Gazetteer	gazetteer
	String	lrs
	説明	登録対象 座標参照系は登録済みのものであること
		参照する空間参照系の名称 登録済みの空間参照系であること
	AuthResult	authResult
	説明	認証結果
戻り値	boolean	
説明	処理に成功した場合、地名辞典サービスメタデータの連番を更新する。 処理に失敗した場合、地名辞典データベースを本関数呼び出し前の状態に戻すこと。	
関連項目	Gazetteer 構造体、AuthResult 構造体	

2) UpdateGazetteer

機能	登録されている地名辞典を更新する	
引数	型	名前
	Gazetteer	gazetteer
	AuthResult	authResult
	説明	更新対象 座標参照系は登録済みのものであること
		認証結果
戻り値	boolean	
説明	処理に成功した場合、地名辞典サービスメタデータの連番を更新する。 処理に失敗した場合、地名辞典データベースを本関数呼び出し前の状態に戻すこと。	
関連項目	Gazetteer 構造体、AuthResult 構造体	

3) DeleteGazetteer

機能	登録されている地名辞典を削除する	
引数	型	名前
	String[]	name
	AuthResult	authResult
	説明	削除対象の地名辞典の名称
		認証結果
戻り値	boolean	
説明	1件でも削除に失敗した場合は false を返す。 処理に失敗した場合、地名辞典データベースを本関数呼び出し前の状態に戻すこと。	
関連項目	AuthResult 構造体	

4) SearchPI

機能	登録されている場所インスタンスの PI 部分を検索する		
引数	型	名前	説明
	RouteCondition	routeCondition	他サービス呼び出し条件
	String	gazetteer	地名辞典の名称 地名辞典を特定しない場合は null を指定する
	String	locationType	場所型の名称 場所型を特定しない場合は null を指定する
	String[]	geographicIdentifier	地理識別子 地理識別子を特定しない場合は null を指定する
	GeographicExtent	geographicExtent	地理的範囲 地理的範囲を限定しない場合は null を指定する
	TemporalExtent	temporalExtent	時間範囲 時間範囲を限定しない場合は null を指定する
	String	administrator	管理者 管理者を特定しない場合は null を指定する
	String[]	searchWord	検索語句 場所インスタンスの PI の値または代替地理識別子を対象とした全文検索用フリーワード。 語句が複数指定された場合は、地理識別子、または代替地理識別子のいずれかがすべての検索語句を含むものを検索する。 検索語句を設定しない場合は null を指定する。
	AuthResult	authResult	認証結果 ユーザ認証を必要としないサービスの場合は null を指定する
戻り値	PI[]		

<p>説明</p>	<p>指定した条件に合致する PI を検索する。 地理的範囲で地理的境界またはポリゴンを指定した場合は、一部でも地理的範囲が交差する PI を検索する。 条件に合致する PI が存在しない場合はサイズ 0 の配列を返す。 (A) routeCondition.cascadingLevel > 0 の場合 他の地名辞典管理サービスの SearchPI 関数を呼び出す場合は routeCondition.cascadingLevel の値を 1 だけ減じ、かつ routeCondition.routeHistory の末尾に自サーバを追加しなければならない。また、routeCondition.routeHistory で指定された呼び出し経路に自サービスが含まれている場合はサイズ 0 の配列を返す。 (B) routeCondition.cascadingLevel = 0 の場合 他の地名辞典管理サービスの SearchPI 関数を呼び出すことはできない。また、routeCondition.routeHistory で指定された呼び出し経路に自サービスが含まれている場合はサイズ 0 の配列を返す。 (C) routeCondition.cascadingLevel < 0 の場合 他の地名辞典管理サービスの SearchPI 関数は routeCondition.routeDefinition の順に呼び出さなければならない。</p>
<p>関連項目</p>	<p>GeographicExtent 構造体、TemporalExtent 構造体、PI 構造体、RouteCondition 構造体、AuthResult 構造体</p>

5) GetLocationInstance

機能	登録されている場所インスタンスを取得する	
引数	型	名前
	RouteCondition	routeCondition
	String	gazetteer
	PI[]	pi
	AuthResult	authResult
戻り値	LocationInstance[]	
説明	<p>指定した PI に合致する場所インスタンスを取得する。 地名辞典を特定して場所インスタンスを取得することもできる。 PI が複数指定された場合は、指定された順に対応する場所インスタンスを返す。PI に対応する場所インスタンスが存在しない場合は、配列中の当該要素に null を格納して返す。 条件に合致する場所インスタンスが存在しない場合はサイズ 0 の配列を返す。</p> <p>(A) routeCondition.cascadingLevel > 0 の場合 他の地名辞典管理サービスの GetLocationInstance 関数を呼び出す場合は routeCondition.cascadingLevel の値を 1 だけ減じ、かつ routeCondition.routeHistory の末尾に自サーバを追加しなければならない。また、routeCondition.routeHistory で指定された呼び出し経路に自サービスが含まれている場合はサイズ 0 の配列を返す。</p> <p>(B) routeCondition.cascadingLevel = 0 の場合 他の地名辞典管理サービスの GetLocationInstance 関数を呼び出すことはできない。また、routeCondition.routeHistory で指定された呼び出し経路に自サービスが含まれている場合はサイズ 0 の配列を返す。</p> <p>(C) routeCondition.cascadingLevel < 0 の場合 他の地名辞典管理サービスの GetLocationInstance 関数は routeCondition.routeDefinition の順に呼び出さなければならない。</p>	
関連項目	PI 構造体、LocationInstance 構造体、RouteCondition 構造体、AuthResult 構造体	

6) AddLocationInstance

機能	場所インスタンスを新規登録する		
引数	型	名前	説明
	String	gazetteer	地名辞典の名称
	LocationInstance[]	locationInstance	場所インスタンス 代表点、地理的境界、ポリゴンのいずれかを持つ場合、座標参照系は登録済みのものである必要がある。
	AuthResult	authResult	認証結果
戻り値	Boolean		
説明	1 件でも登録に失敗した場合には例外 UpdateFailure を発生する。 処理に失敗した場合、地名辞典データベースを本関数呼び出し前の状態に戻すこと。		
関連項目	LocationInstance 構造体、AuthResult 構造体		

7) UpdateLocationInstance

機能	登録されている場所インスタンスを更新する		
引数	型	名前	説明
	String	gazetteer	地名辞典の名称
	LocationInstance[]	locationInstance	場所インスタンス 代表点、地理的境界、ポリゴンのいずれかを持つ場合、座標参照系は登録済みのものである必要がある。
	AuthResult	authResult	認証結果
戻り値	boolean		
説明	1 件でも更新に失敗した場合には false を発生する。 処理に失敗した場合、地名辞典データベースを本関数呼び出し前の状態に戻すこと。		
関連項目	LocationInstance 構造体、AuthResult 構造体		

8) DeleteLocationInstance

機能	登録されている場所インスタンスを削除する		
引数	型	名前	説明
	String	name	地名辞典の名称
	String[]	geographicIdentifier	削除対象場所インスタンスの地理識別子
	AuthResult	authResult	認証結果
戻り値	Boolean		
説明	1 件でも削除に失敗した場合には false を返す。 処理に失敗した場合、地名辞典データベースを本関数呼び出し前の状態に戻すこと。		
関連項目	AuthResult 構造体		

9) SearchLocationInstanceAssociation

機能	複数場所インスタンス関連を検索する	
引数	型	名前
	RouteCondition	routeCondition
	String	sourceGazetteer
	String	sourceGeographicIdentifier
	String	targetGazetteer
	String	targetGeographicIdentifier
	String[]	phase
	AuthResult	authResult
戻り値	LocationInstanceAssociation[]	

<p>説明</p>	<p>条件に合致する場所インスタンス関連が存在しない場合はサイズ 0 の配列を返す。</p> <p>(A) routeCondition.cascadingLevel > 0 の場合 他の地名辞典管理サービスの SearchLocationInstanceAssociation 関数を呼び出す場合は routeCondition.cascadingLevel の値を 1 だけ減じ、かつ routeCondition.routeHistory の末尾に自サーバを追加しなければならない。また、routeCondition.routeHistory で指定された呼び出し経路に自サービスが含まれている場合はサイズ 0 の配列を返す。</p> <p>(B) routeCondition.cascadingLevel = 0 の場合 他の地名辞典管理サービスの SearchLocationInstanceAssociation 関数を呼び出すことはできない。また、routeCondition.routeHistory で指定された呼び出し経路に自サービスが含まれている場合はサイズ 0 の配列を返す。</p> <p>(C) routeCondition.cascadingLevel < 0 の場合 他の地名辞典管理サービスの SearchLocationInstanceAssociation 関数は routeCondition.routeDefinition の順に呼び出さなければならない。</p> <p>Nesting 関連を検索する場合は source を child に、target を parent に読み替える。</p>
<p>関連項目</p>	<p>LocationInstanceAssociation 構造体、 RouteCondition 構造体、 AuthResult 構造体</p>

10) AddLocationInstanceAssociation

機能	場所インスタンス関連を新規登録する	
引数	型	名前
	LocationInstanceAssociation[]	association
	AuthResult	authResult
説明	登録対象 親および子の場所インスタンスは登録済みのものでなければならない	
戻り値	boolean	
説明	1件でも登録に失敗した場合は false を返す。 処理に失敗した場合、地名辞典データベースを本関数呼び出し前の状態に戻すこと。	
関連項目	LocationInstanceAssociation 構造体、AuthResult 構造体	

11) DeleteLocationInstanceAssociation

機能	登録されている場所インスタンス関連を削除する	
引数	型	名前
	LocationInstanceAssociation	association
	AuthResult	authResult
説明	削除対象	
戻り値	boolean	
説明	1件でも削除に失敗した場合は false を返す。 処理に失敗した場合、地名辞典データベースを本関数呼び出し前の状態に戻すこと。	
関連項目	LocationInstanceAssociation 構造体、AuthResult 構造体	

3-2-4. メタデータ検索に関するインタフェース

3-2-4-1. DescribeRecord

機能	レコードの概要(レコードが記述される形式の詳細)を取得する	
引数	型	名前
	String	request
	String	service
	String	version
	String	namespace
	String	typeName
	String	outputFormat
	String	schemaLanguage
戻り値	DescribeRecordResponse	schemaLanguage で指定したスキーマ言語に基づくレコードの構造
関連項目	DescribeRecordResponse 構造体	

3-2-4-2. GetDomain

機能	メタデータの各項目に対する定義域を取得する	
引数	型	名前
	String	request
	String	service
	String	version
	String	parameterName
String	propertyName	
説明	要求名 "GetDomain"でなければならない	
説明	サービスタイプ "CSW"でなければならない	
説明	準拠する標準インタフェース(CSW)のバージョン番号 該当するバージョンのサービスメタデータのみを取得する 指定なしの場合は、最新バージョン	
説明	パラメータ名	
説明	プロパティ名	
戻り値	GetDomainResponse	指定したパラメータあるいはプロパティのデータ構造を示した XML Schema
関連項目	GetDomainResponse 構造体	

3-2-4-3. GetRecords

機能	検索結果一覧の取得		
引数	型	名前	説明
	String	request	要求名 "GetRecords"でなければならない
	String	service	サービスタイプ "CSW"でなければならない
	String	version	バージョン番号
	String	namespace	レコードが参照する名前空間のリスト
	String	resultType	以下のいずれかより選択: - hits - results - validate
	String	outputFormat	出力ファイルフォーマット。 "text/xml" 等
	String	outputSchema	出力ファイルフォーマット(XML)が参照するスキーマ名
	Integer	startPosition	検索開始位置(カーソル位置)
	Integer	maxRecords	検索結果の件数の上限値
	String	typeName	コレクション ID
	String	elementName / elementSetName	要素名 or 要素体名
	String	constraintLanguage	検索条件式の形式 CQL_TEXT or FILTER(OGC Filter)
	String	constraint	検索条件式(クエリ)
	String[]	sortBy	ソート対象のフィールドと順番のリスト
	Boolean	distributedSearch	検索クエリが既存であるかどうかの論理値 規定値は FALSE
Integer	hopCount	検索クエリの上限值 規定値は"2"	
URL	responseHandler	操作要求の応答状況をモニタするハンドラ	
戻り値	GetRecordsResponse		
関連項目	GetRecordsResponse 構造体		

3-2-4-4. GetRecordById

機能	IDによる検索結果(メタデータ)の取得	
引数	型	名前
	String	version
	String[]	id
戻り値	Metadata	検索結果レコード
関連項目	Metadata 構造体	

3-2-4-5. Transaction

機能	レコードを追加・更新・削除する	
引数	型	名前
	Transaction	transaction
	URI	requested
	Boolean	verboseResponse
戻り値	TransactionResponse	実行したトランザクションの結果を報告する
関連項目	Transaction 構造体、 TransactionResponse 構造体	

3-2-4-6. Harvest

機能	レコードの更新状況の収集		
引数	型	名前	説明
	String	request	要求名 "Harvest"でなければならない
	String	service	サービスタイプ "CSW"でなければならない
	String	version	準拠する標準インタフェース(CSW)のバージョン番号 指定なしの場合は、最新バージョン
	String[]	namespace	レコードが参照する名前空間のリスト
	URI	source	リソースとなるメタデータを示す URI
	URI	resourceType	リソースとなるメタデータの構造を示す URI 例) リソースとなるメタデータを規定する XML Schema を参照する URI など
	String	resourceFormat	リソースのフォーマット 収集 (Harvest) するリソースのフォーマット (MIME Type) を指定する 省略された場合は、"text/xml"とみなされる
	URL	responseHandler	操作要求の応答状況をモニタするハンドラ
	TemporalExtent	harvestInterval	時間範囲
戻り値	TransactionResponse		
関連項目	TemporalExtent 構造体、 TransactionResponse 構造体		

3-2-5. 構造体

3-2-5-1. 地理情報

1) WMSServiceMetadata

説明	地理情報サービスのメタデータ		
要素	型	名前	説明
	String	name	サービスの名称。 規定では"WMS"でなければならない。
	String	title	サービス提供元を示す名前。 WMS サーバ名などを記述する。
	String	abstract	サービスの概要。 サービス提供元あるいはWMS サーバを簡潔に説明した文章。
	String[]	keywordList	キーワードのリスト。 提供するサービスを端的に表すキーワードのリスト。
	URL	onlineResource	オンラインリソース。 WMS における要求を送信可能な具体的 URL。
	ResponsibleParty	contactInformation	問合せ先情報。 サービス提供元や WMS サーバ開発元など、サービスに関する問合せが可能な連絡先等の情報。 ※ ISO19115 で規定される"CI_ResponsibleParty"クラスを引用しており、住所、電話番号等の情報を含む。
	String	fees	サービスの利用における課金の有無。 ※有料である場合は、必ず示されなければならない。
	Integer	layerLimit	一回の GetMap 要求で指定できるレイヤーの上限数。 ※上限が存在する場合は、必ず示されなければならない。
	Integer	maxWidth	WMS サーバが生成する地図画像サイズ(幅)の上限値。 ※上限が存在する場合は、必ず示されなければならない。

	Integer maxHeight	WMS サーバが生成する地図画像サイズ (高さ)の上限値。 ※上限が存在する場合は、必ず示されな なければならない。
	OperationMet request adata[]	GetCapabilities 、 GetMap 、 GetFeatureInfo それぞれに関する詳細 情報
	String[] exception	例外報告フォーマットに指定可能なフォー マットの情報 例外報告を行う際に指定可能なフォー マット
	LayerMetadat layer a	地図画像生成元となるレイヤーに関する 情報
関連項目	ResponsibleParty 構造体、OperationMetadata 構造体、LayerMetadata 構造体	

2) LayerMetadata

説明	地図画像生成元となるレイヤーに関する情報		
要素	型	名前	説明
	String	name	レイヤーの名称
	String	title	レイヤーのタイトル ※GetMap 要求では、Name ではなく Title によりレイヤー名を指定するため、Title は必須である。
	String	abstract	レイヤーの概要 レイヤーに含まれる地物の情報や、主題の名称
	String[]	keywordList	キーワードリスト レイヤーを端的にあらわすキーワードのリスト
	String	crs	レイヤーに適用される座標参照系 レイヤーCRS の座標参照系識別子
	Double[]	EX_Geographic BoundingBox	地理境界ボックス レイヤー全体の東西南北の境界緯度経度 ※絶対緯度経度による記述
	GeographicBoundingBox[]	boundingBox	境界ボックス レイヤー全体の境界座標 ※座標参照系を含み、座標参照系ごとの記述が可能
		dimension	次元 レイヤーが属する次元。※さらに子要素を持つクラスである
	InfoURL	attribution	情報提供元の属性情報
	InfoURL	authorityURL	作者のリソース情報
	String	identifier	レイヤーを識別する ID(識別子)
	InfoURL	metadataURL	メタデータのリソース情報
	InfoURL	dataURL	実データのリソース情報
	InfoURL	featureListURL	地物リストのリソース情報
	InfoURL	style	描画スタイルのリソース情報
Integer	minScaleDenominator	最小スケールの分母 最小表示スケールが存在する場合、その分母の数値	
Integer	maxScaleDenominator	最大スケールの分母 最大表示スケールが存在する場合、その分母の数値	

	Layer layer	子レイヤの情報 ※レイヤーは、子レイヤーを持つことができ、階層構造を構成できる
関連項目	GeographicBoundingBox 構造体、InfoURL 構造体、Layer 構造体	

3) InfoURL

説明	画像リソースの URL 情報		
要素	型	名前	説明
	String	name	名称
	String	title	リソースのタイトル
	String	abstract	リソースの概要
	Integer	width	画像の幅(pixel) ※リソースが画像の場合
	Integer	height	画像の高さ(pixel) ※リソースが画像の場合
	String	format	リソースの MIME タイプ
	URL	onlineResource	リソースの URL
	InfoURL	url	リソースの URL 情報
関連項目			

3-2-5-2. 地名辞典(空間参照系管理)

1) SRSServiceMetadata

用途	空間参照系管理サービスのメタデータ	
要素	型	名前
	String	version
	boolean	canUpdateLRS
	boolean	canUpdateCRS
	boolean	canUpdateConversionInfo
	boolean	canUpdateLocationType
	boolean	needsExam
	String	updateSequence
説明	更新連番は任意の文字列であるが、キャッシュ機能を実現するために、サービスは更新連番の値の大小を判断できることが必要である。	
関連項目		

2) LRS

用途	地理識別子による空間参照系	
要素	型	名前
	String	name
	GeographicExtent	domainOfValidity
	String	theme
	String	overallOwner
説明		
関連項目	GeographicExtent 構造体	

3) SRSExamResult

用途	空間参照系審査結果		
要素	型	名前	説明
	String	name	空間参照系名称
	String	examNo	審査結果番号
説明			
関連項目			

4) PIType

用途	PIタイプ		
要素	型	名前	説明
	String	schemaUrl	PI スキーマ文書の URL
	String	encodingRuleUrl	PI 符号化規則文書の URL
	String	generationRuleUrl	PI 符号化規則作成規則文書の URL
説明			
関連項目			

5) LocationType

用途	場所型		
要素	型	名前	説明
	String	name	名称
	String	theme	分類
	PIType	identification	PIタイプ
	String	definition	定義
	GeographicExtent	territoryOfUse	地理的範囲
	String	owner	管理者
	String	name	名称
	String	theme	分類
説明			
関連項目	PIType 構造体、GeographicExtent 構造体		

6) LocationTypeAssociation

用途	場所型関連		
要素	型	名前	説明
	String	childLocationType	子場所型の名称 null 指定不可
	String	parentLocationType	親場所型の名称 null 指定不可
説明			
関連項目			

7) LocationTypeExamResult

用途	場所型審査結果		
要素	型	名前	説明
	String	name	場所型名称
	String	examNo	審査結果番号
説明			
関連項目			

3-2-5-3. 地名辞典(地名辞典管理)

1) GazetteerServiceMetadata

用途	地名辞典サービスメタデータ		
要素	型	名前	説明
	String	version	地名辞典サービスのバージョン 本仕様によるサービスメタデータでは 1.0 を指定する。
	Gazetteer[]	gazetteer	地名辞典情報
	GazetteerStatistics[]	gazetteerStatistics	地名辞典統計情報
	boolean	canUpdateLocationInstance	場所インスタンスの登録可否
	boolean	canUpdateLocationInstanceAssociation	場所インスタンス関連の登録可否
	String	updateSequence	連番
説明			
関連項目	Gazetteer 構造体、GazetteerStatistics 構造体		

2) Gazetteer

用途	地名辞典		
要素	型	名前	説明
	String	name	名称
	String[]	alias	別名
	String	scope	スコープ
	GeographicExtent	territoryOfUse	地理的範囲
	String	custodian	管理者
	String	crs	座標参照系
	TemporalExtent	date	時間範囲
説明			
関連項目	GeographicExtent 構造体、TemporalExtent 構造体		

3) GazetteerStatistics

用途	地名辞典統計情報		
要素	型	名前	説明
	String	gazetteer	地名辞典名称
	String	locationType	場所型名称
	Integer	count	総データ件数
説明			
関連項目			

4) PI

用途	PI		
要素	型	名前	説明
	String	value	地理識別子 null は指定不可
	String	lrs	地理識別子による空間参照系の名称 null は指定不可
説明	現実世界のある場所を一意に識別するための識別子。		
関連項目			

5) LocationInstance

用途	場所インスタンス		
要素	型	名前	説明
	PI	PI	PI 地理識別子と空間参照系の組 null は指定不可
	String	locationType	場所型名称 場所型の実体は空間参照系管理サービスが保持する null は指定不可
	String[]	alternative-GeographicIdentifier	代替地理識別子 地理識別子の別名 代替地理識別子がない場合は null を指定する
	GeographicExtent	geographicExtent	地理的範囲
	Position	position	位置 代表点の座標
	String	administrator	管理者 場所インスタンスの特質を定義する責任がある組織の名称 null は指定不可
	TemporalExtent	temporalExtent	時間範囲 時間範囲を限定しない場合は null を指定する
	String	gazetteer	地名辞典の名称
説明			
関連項目	PI 構造体、GeographicExtent 構造体、Position 構造体、TemporalExtent 構造体		

6) LocationInstanceAssociation

用途	場所インスタンス関連		
要素	型	名前	説明
	String	sourceGazetteer	関連元場所インスタンスの地名辞典の名称
	String	sourceGeographicIdentifier	関連元場所インスタンスの地理識別子
	String	targetGazetteer	関連先場所インスタンスの地名辞典の名称
	String	targetGeographicIdentifier	関連先場所インスタンスの地理識別子
	String	phase	位相の名称
説明	phase には“Nesting”または“Equivalent”を指定することができる。		
関連項目			

7) RouteCondition

用途	地名辞典管理サービス呼び出し経路条件		
要素	型	名前	説明
	Integer	cascadingLevel	他サービス呼び出し可能レベル 0 以下: 他サービス呼び出さない 1 以上: 他サービスを最大 N 段階呼び出すことができる
	Route[]	routeHistory	他サービス呼び出し経路履歴
	Route[]	routeDefinition	他サービス呼び出し経路定義
説明			
関連項目			

8) Route

用途	地名辞典管理サービスの呼び出し経路		
要素	型	名前	説明
	String	serviceName	地名辞典管理サービスの名称
説明			
関連項目			

3-2-5-4. メタデータ検索

1) CSWServiceMetadata

用途	カタログサービスのサービスメタデータ		
要素	型	名前	説明
	String	name	サービスの名称。 規定では"CWS"でなければならない。
	String	title	サービス提供元を示す名前。 CWS サーバ名などを記述する。
	String	abstract	サービスの概要。 サービス提供元あるいは CWS サーバを簡潔に説明した文章。
	String[]	keywordList	キーワードのリスト。 提供するサービスを端的に表すキーワードのリスト。
	URL	onlineResource	オンラインリソース。 CWS における要求を送信可能な具体的な URL。
	Responsible Party	contactInformation	問合せ先情報。 サービス提供元や CWS サーバ開発元など、サービスに関する問合せが可能な連絡先等の情報。 ※ ISO19115 で規定される"CI_ResponsibleParty"クラスを引用しており、住所、電話番号等の情報を含む。
	String	fees	サービスの利用における課金の有無。 ※有料である場合は、必ず示されなければならない。
	OperationMetadata[]	operationsMetadata	CWS サーバが提供可能な操作についてのメタデータ
Filter_Capabilities	filter_Capabilities	CWS サーバが処理可能な Transaction クエリに関するメタデータ ※OGC Filter1.0 に基づく	
説明	カタログサービスが提供可能なサービスのメタデータを返す		
関連項目	ResponsibleParty 構造体、OperationsMetadata 構造体、Filter_Capabilities 構造体		

2) DescribeRecordResponse

用途	検索対象メタデータの構造		
要素	型	名前	説明
	URI	targetNamespace	スキーマの規定名前空間
	String	parentSchema	親スキーマのファイル名称
	String	schemaLanguage	スキーマの言語
説明	検索対象メタデータの構造を明らかにするために必要な XML Schema の情報を返す		
関連項目			

3) Filter_Capabilities

用途	CWS サーバが処理可能な Transaction クエリに関するメタデータ		
要素	型	名前	説明
	Logical operators	and	CSW サーバが処理可能な論理演算
		or	
		not	
	Comparison operators	propertyIsEqualTo	CSW サーバが処理可能な比較演算
		propertyIsNotEqualTo	
		propertyIsLessThan	
		propertyIsGreaterThan	
		propertyIsLessThanOrEqualTo	
		propertyIsGreaterThanOrEqualTo	
Spatial operators	BBox	CSW サーバが処理可能な空間演算	
説明	CWS サーバが処理可能な Transaction クエリに関するメタデータの情報を返す		
関連項目			

4) GetDomainResponse

用途	メタデータ属性値の項目 (Property) と定義域 (Values)		
要素	型	名前	説明
	String	parameterName	項目名
	String[]	value	項目名に対応する値のリスト
説明	検索対象メタデータの各属性値の情報を返す		
関連項目			

5) GetRecordsResponse

用途	検索結果		
要素	型	名前	説明
	String	requestId	リクエスト ID
	String	status	検索結果ステータス
	String	resultSetId	検索結果セット ID
	String	elementSet	要素セット名称
	String	recordSchema	レコードの書式 (XML Schema 等のスキーマを特定する名称)
	Integer	numberOfRecordsMatched	検索ヒット総数
	Integer	numberOfRecordsReturned	検索結果に含まれる総数
	Integer	nextRecord	次の検索結果リストの始まりのレコード位置
Metadata	resultRecord	検索結果レコード	
説明	検索結果メタデータの概要情報を返す		
関連項目	Metadata 構造体		

6) Metadata

用途	検索結果メタデータの詳細		
要素	型	名前	説明
		any	検索結果メタデータレコードにより異なる
説明	✓ 検索結果レコードである Metadata 構造体は、GetRecordResponse 構造体におけるパラメータ recordSchema で示されるスキーマによってその構造が決定される。		
関連項目			

7) Transaction

(i) Transaction.Insert

用途	トランザクションクエリ(Insert)		
要素	型	名前	説明
	String	transactionType	“update”
	String	requestId	リクエスト ID
	Record	record	処理対象レコード
	Boolean	verboseResponse	レスポンスの種類 TRUE: 通常のレスポンスを要求 FALSE: CSW サーバ固有のレスポンスを 要求
String	handle	操作要求の応答状況	
説明	Insert 処理のリクエストクエリを記述		
関連項目	Record 構造体		

Record 構造体は、処理対象レコードの構造によりその都度異なる

(ii) Transaction.Update

用途	トランザクションクエリ(Update)		
要素	型	名前	説明
	String	transactionType	"update"
	String	requestId	リクエストID
	Record	record	処理対象レコード
	Boolean	verboseResponse	レスポンスの種類 TRUE: 通常のレスポンスを 要求 FALSE: CSW サーバ固有 のレスポンスを要求
	String	recordProperty	レコードプロパティ ※Record がリクエストされな い場合
	String	typeNameNames	処理対象レコードの種類 例) "Service"、 "Dataset"、 Dataset Collection"、 "Application" 等
	Codelist	constraintLanguage	規定値"Filter"
	String	constraint	OGC Filter1.0.0に基づく符 号化規則によるクエリ式
	String	handle	操作要求の応答状況
説明	Update 処理のリクエストクエリを記述		
関連項目	Record 構造体		

Record 構造体は、処理対象レコードの構造によりその都度異なる

(iii) Transaction.Delete

用途	トランザクションクエリ(Delete)		
要素	型	名前	説明
	String	transactionType	"delete"
	String	requestId	リクエスト ID
	Record	record	処理対象レコード
	Boolean	verboseResponse	レスポンスの種類 TRUE: 通常のレスポンスを要求 FALSE: CSW サーバ固有のレスポンスを要求
	String	recordProperty	レコードプロパティ ※Record がリクエストされない場合
	String	typeNameNames	処理対象レコードの種類 例) "Service"、"Dataset"、Dataset Collection"、"Application" 等
	Codelist	constraintLanguage	規定値"Filter"
	String	constraint	OGC Filter1.0.0 に基づく符号化規則によるクエリ式
String	handle	操作要求の応答状況	
説明	Delete 処理のリクエストクエリを記述		
関連項目	Record 構造体		

Record 構造体は、処理対象レコードの構造によりその都度異なる

8) TransactionResponse

用途	トランザクションの結果		
要素	型	名前	説明
	String	requestId	リクエスト ID
	Integer	totalInserted	Insert 成功数(※Insert の場合)
	Integer	totalUpdated	Update 成功数(※Update の場合)
	Integer	totalDeleted	Delete 成功数(※Delete の場合)
	String	handleRef	操作 ID
	Record	briefRecord	Insert/Update/Delete した内容の詳細
説明	トランザクションの結果を報告する		
関連項目	Record 構造体(※Record 構造体は、Insert/Update/Delete したレコードの構造をそのまま報告する)		

3-2-5-5. 共通仕様

1) AuthResult

用途	ユーザ認証結果		
要素	型	名前	説明
	String	token	認証トークン
説明	サービスが発行する、ユーザ認証の結果を一意に特定する文字列を保持する。この文字列を認証トークンと呼ぶことにする。 関数呼び出しの際にユーザ認証結果を受け取ることで、サービスは利用ユーザを特定でき、ユーザごとに機能の利用を制限することができる。		
関連項目			

2) ContactAddress

説明	住所情報		
要素	型	名前	説明
	String	addressType	住所の型
	String	address	住所
	String	city	市町村
	String	stateOrProvince	州、郡、都道府県
	String	postCode	郵便番号
	String	country	国
関連項目			

3) ContactPersonPrimary

説明	問合せ先名称情報		
要素	型	名前	説明
	String	contactPerson	問合せ先個人名
	String	contactOrganization	問合せ先組織名
関連項目			

4) Datum

用途	原子		
要素	型	名前	説明
	String	datumID	原子の識別子
	String[]	alias	原子の別名
	String	type	原子の型
	String	point	地球に固定するために使用する点
	DateTime	realizationEpoch	原子の実現時期
	Extent	validArea	有効な地域
	String	scope	有効な応用分野
	String	remark	注釈
	PrimeMeridian	primeMeridian	本初子午線情報
Ellipsoid	ellipsoid	楕円体情報	
説明			
関連項目	Extent 構造体、PrimeMeridian 構造体、Ellipsoid 構造体		

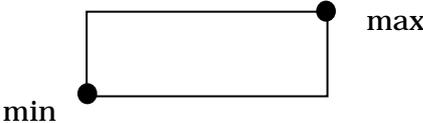
5) Ellipsoid

用途	楕円体		
要素	型	名前	説明
	String	ellipsoidID	原子に対する楕円体の識別子
	String[]	alias	別名
	Double	semiMajorAxis	楕円体の長半径の長さ
	boolean	ellipsoidShape	楕円体の計上
	InverseFlattening	inverseFlattening	楕円体の逆扁平率
	String	remark	注釈
説明			
関連項目	InverseFlattening 構造体		

6) Extent

用途	範囲情報		
要素	型	名前	説明
	String	description	記述
	GeographicExtent[]	geographicElement	地理的範囲
	TemporalExtent[]	temporalElement	時間範囲
	VerticalExtent[]	verticalElement	鉛直範囲
説明			
関連項目	GeographicExtent 構造体、TemporalExtent 構造体、VerticalExtent 構造体		

7) GeographicBoundingBox

用途	地理的境界		
要素	型	名前	説明
	double[]	min	最小座標
	double[]	max	最大座標
	String	crs	座標参照系
説明	<p>矩形で領域を定義する。min と max は下図で示す位置の座標を指定する。</p>  <p>座標配列には座標参照系で定義された順番で値を格納する。 各座標は以下の条件を満たす必要がある。 $\min[i] \leq \max[i]$ (i: 配列のインデックス)</p>		
関連項目			

8) GeographicExtent

用途	地理的範囲		
要素	型	名前	説明
	Polygon	boundingPolygon	ポリゴン
	GeographicBoundingBox	boundingBox	矩形
	String	description	記述
説明	boundingPolygon、 boundingBox、 description のいずれか1つのみ null 以外の値を指定する。		
関連項目	Polygon 構造体、GeographicBoundingBox 構造体		

9) InverseFlattening

用途	楕円体扁平率		
要素	型	名前	説明
	Double	inverseFlattening	逆扁平率
	boolean	isInfinite	
説明			
関連項目			

10) OperationMetadata

説明	サービスが提供可能な操作 (Operation) の詳細情報		
要素	型	名前	説明
	String	operation	提供可能な操作の名称
	String[]	format	応答のファイル形式 (MIME Type)
	URL[]	onlinerResource	Get メソッド/POST メソッドの要求先 URL (※ POST メソッドの URL は、POST メソッドを実装している場合のみ)
	String	Parameter.name	操作が持つパラメータの名称
	String	Parameter.value	操作が持つパラメータの名称に対応する値
	String	extendedCapabilities	本体持つ操作に対し、実装者が拡張した内容
関連項目			

11) Polygon

用途	ポリゴン		
要素	型	名前	説明
	Ring	exterior	外周 null を指定してはならない
	Ring[]	interior	内周 穴あきポリゴンを定義する場合に指定する。穴あきポリゴンでない場合は null を指定する。
説明	1 つの外周と複数の内周からなる多角形を定義する。 外周や内周が内部でもつ座標参照系はすべて同一のものでなくてはならない。		
関連項目	Ring 構造体		

12) Position

用途	位置		
要素	型	名前	説明
	double[]	coord	座標
	String	crs	座標参照系
説明	座標配列には空間参照系で定義された順番で値を格納する。		
関連項目			

13) PrimeMeridian

用途	本初子午線		
要素	型	名前	説明
	String	meridianID	本初子午線の識別子
	Double	GreenwichLongitude	本書子午線グリニッジ経度
	String	remark	注釈
説明			
関連項目			

14) ResponsibleParty

説明	問合せ先情報		
要素	型	名前	説明
	ContactPersonPrimary	contactPersonPrimary	問合せ先名称情報
	String	contactPosition	問合せ先の役職
	ContactAddresses	contactAddress	問合せ先住所情報
	String	contactVoiceTelephone	問合せ先電話番号
	String	contactFacsimileTelephone	問合せ先 FAX 番号
	String	contactElectronicMailAddresses	問合せ先電子メールアドレス
関連項目	ContactPersonPrimary 構造体、ContactAddress 構造体		

15) Ring

用途	リング		
要素	型	名前	説明
	Position[]	coords	座標列
説明	交差しない閉図形を定義する。始点と終点は一致する必要がある。座標列は 3 つ以上の点を含まなければならない。		
関連項目	Position 構造体		

16) TemporalExtent

用途	時間範囲		
要素	型	名前	説明
	DateTime	begin	開始時点 終了時点以前のすべての期間を指定する場合は null を指定する
	DateTime	end	終了時点 開始時点以後のすべての期間を指定する場合は null を指定する
説明	begin、end のいずれか1つは null でない値を指定しなければならない。 また、begin、end ともに値を指定する場合は $begin \leq end$ でなければならない。		
関連項目			

17) VerticalExtent

用途	鉛直範囲		
要素	型	名前	説明
	Double	minimumValue	最低値
	Double	maximumValue	最大値
	String	unitOfMeasure	測定単位
	Datum	verticalDatum	鉛直原子
説明			
関連項目	Datum 構造体		

3-2-5-6. 例外

1) InvalidUpdateSequence

用途	サービスメタデータの更新連番の異常
説明	更新連番の大小関係が異常な場合や大小関係を評価できない場合に発生する。 例外の表現方法は処理系ごとに定義する共通仕様に従う。
関連項目	GetCapabilities 関数

2) InvalidArgument

用途	関数の引数に不正な値が指定された場合の例外
説明	null を許可しない引数に null が指定された場合に発生する。 例外の表現方法は処理系ごとに定義する共通仕様に従う。
関連項目	すべての関数

3) InvalidValue

用途	構造体に不正な値が指定された場合の例外
説明	null を許可しない構造体要素に null が指定された場合に発生する。ただし、関数の引数に指定された構造体の場合は InvalidArgument を発生する。 例外の表現方法は処理系ごとに定義する共通仕様に従う。
関連項目	すべての関数

4) UpdateFailure

用途	データベースの追加、更新、削除に失敗した場合の例外
説明	各種更新系の処理で失敗した場合に発生する。 例外の表現方法は処理系ごとに定義する共通仕様に従う。
関連項目	データベースのトランザクション処理を行うすべての関数

4. 標準インタフェース実装仕様

3章で示した関数仕様に対し、アプリケーションプログラムとして実装を行う場合に必要となる各関数の機能の説明、利用条件、リクエストの内容、レスポンスの内容、例外処理、および各構造体実装時の注意事項について示す。

4-1節では、POST/GET 利用のための実装仕様について示す。

4-2節では、SOAP と WSDL 利用のための実装仕様について示す。

4-1. POST/GET 利用のための実装仕様

4-1-1. 地理情報に関するインタフェース

地理情報に関するインタフェースにおける POST/GET 利用のための実装仕様は、ISO19128 Web Map Server Interface に準ずる。POST/GET 利用のためのインタフェースを実装する場合には、ISO19128 Web Map Server Interface のレスポンスの規則、座標系、リクエストパラメータの規則、各関数の記述規則等に従うものとする。

ISO19128 において規定している Web Map Server Interface のプロトコルバージョンは、「1.3.0」である。

4-1-2. 地名辞典に関するインタフェース

地名辞典に関するインタフェースにおける POST/GET 利用の実装仕様は、特に定めがない。

本ガイドラインにおいては、地名辞典に関するインタフェースに関しては、4-2-4 に示す SOAP と WSDL 利用の実装を推奨する。

4-1-3. メタデータ検索に関するインタフェース

メタデータ検索に関するインタフェースにおける POST/GET 利用の実装仕様は、OGC Catalogue Service Specification において規定されている、HTTP を通信プロトコルとした実装仕様である Catalogue Services for the Web (CSW) に準ずる。POST/GET 利用のためのインタフェースを実装する場合には、CSW のレスポンスの規則、リクエストパラメータの規則、各関数の記述規則等に従うものとする。

CSW において規定しているプロトコルバージョンは、「2.0.0」である。

4-2. SOAP と WSDL 利用のための実装仕様

4-2-1. 共通事項

4-2-1-1. トランスポートプロトコル

SOAP1.1 では、通信プロトコルとして、HTTP のほか、SMTP、FTP などの利用が可能であるが、本ガイドラインでは、HTTP での SOAP 通信を実装することとする。

4-2-1-2. 例外処理

無効なリクエストが発行された場合、実装されていないサービスが要求された場合などは、例外処理として、Fault 要素を含む SOAP メッセージを送付する。Fault 要素には子要素として detail 要素を設け、以下のアプリケーション固有のメッセージを付与するものとする。

```
<someFault>
  <message>メッセージ</message>
  <code>コード</code>
</someFault>
( someFault は例外の内容を表す任意の文字列 )
```

代表的な例として以下のものを定義する。

コード	メッセージ
999	この関数は未実装です。
001	パラメータが不正です。
002	セッションがタイムアウトしました。

使用するすべてのコードはインタフェース実装ごとに ServiceMetadata 構造体で定義する。

4-2-1-3. 座標参照系

本ガイドラインで取り扱う座標系は、JIS X 7115 の 付属書 2 (規定) 「日本における座標参照系の表記」に従い、測地原子と座標系の組み合わせで表記し、表 4-1 のとおりとする。

表 4-1 本ガイドラインで取り扱う座標系

必須	座標参照系	内容
	JGD2000 / (B, L)	日本測地系 2000 / 緯度経度
	JGD2000 / (B, L, h)	日本測地系 2000 / 緯度経度および高さ ^{※1)}
○	JGD2000, TP / (B, L), H	日本測地系 2000 / 緯度経度および高さ ^{※2)}
	JGD2000 / (X, Y, Z)	日本測地系 2000 / 3次元座標
	WGS84 / (B, L)	世界測地系 / 緯度経度
	WGS84 / (B, L, h)	世界測地系 / 緯度経度および高さ ^{※1)}
	WGS84, TP / (B, L), H	世界測地系 / 緯度経度および高さ ^{※2)}
	WGS84 / (X, Y, Z)	世界測地系 / 3次元座標
	JGD2000 / [Zone No.] ^{※3)} (X, Y)	日本測地系 2000 / 平成 14 年国土交通省告示第 9 号に定める平面直角座標系
	JGD2000 / [Zone No.] ^{※4)} (E, N)	日本測地系 2000 / UTM 座標系
	JGD2000, TP / [Zone No.] ^{※3)} (X, Y), H	日本測地系 2000 / 平成 14 年国土交通省告示第 9 号に定める平面直角座標系(高さを含む) ^{※2)}
	JGD2000, TP / [Zone No.] ^{※4)} (E, N), H	日本測地系 2000 / UTM 座標系(高さを含む) ^{※2)}
	WGS84 / [Zone No.] ^{※3)} (X, Y)	世界測地系 / 平成 14 年国土交通省告示第 9 号に定める平面直角座標系
	WGS84 / [Zone No.] ^{※4)} (E, N)	世界測地系 / UTM 座標系
	WGS84, TP / [Zone No.] ^{※3)} (X, Y), H	世界測地系 / 平成 14 年国土交通省告示第 9 号に定める平面直角座標系(高さを含む) ^{※2)}
	WGS84, TP / [Zone No.] ^{※4)} (E, N), H	世界測地系 / UTM 座標系(高さを含む) ^{※2)}

1 : 楕円体高

2 : 東京湾平均海面からの高さ

3 : 平成 14 年国土交通省告示第 9 号に定める平面直角座標系で用いられる 1~19 までのゾーン No.のいずれか

4 : UTM 座標系で用いられる 1~60 までのゾーン No.のいずれか

4-2-1-4. データタイプ

エンコーディングは、XML Schema Part2 に準拠する。なお本ガイドライン中で取り扱われている変数のデータ型は、同規格に準拠したデータ型（XML Schema DataType）とする（表 4-2）。

表 4-2 データ型

ガイドライン表記データ型	XML Schema DataType
整数	long
実数	double
文字	string
日時	dateTime
論理	boolean
バイト	base64Binary

4-2-1-5. 文字エンコーディング

SOAP メッセージの文字エンコーディングは Basic Profile 1.0 で規定されている以下のものを採用する。

- UTF-8
- UTF-16

4-2-2. 各サービスに共通するインタフェース

4-2-2-1. GetCapabilities

1) 機能説明

サービス提供者（以下、プロバイダ）が提供するサービスの内容（サービスメタデータ）を取得する。主な用途は、以下のとおり。

- サービス利用者（以下、リクエスタ）が取扱可能な標準インタフェースのバージョンと、サービス提供者（以下、プロバイダ）が提供可能な標準インタフェースのバージョンの互換性確認を行う。
- プロバイダが提供するサービスに関しての利用条件を把握する。
- なおサービスメタデータは、プロバイダが提供するサービスの内容についてを示したものである。サービスメタデータの内容に関わるサービス更新がされた場合（地図サービスにおけるレイヤ追加など）は、メタデータが更新され、追隨して連番が加算される。

2) 利用条件

リクエスタは、サービス接続直後にバージョンを取得し互換性を確認する。

3) リクエストの内容

リクエストパラメータは、以下のとおりである。

型	名前	入力規則	説明
String	version	指定しない場合は Null を入力する	リクエスタが利用する標準インタフェースのバージョン番号を指定する
String	service	必須	要求するサービスの識別子を指定する
String	request	必須	要求名"GetCapabilities"を指定する
String	format	指定しない場合は Null を入力する	サービスメタデータの出カフォーマットを指定する
String	updateSequence	指定しない場合は Null を入力する	リクエスタがすでにキャッシュしているサービスメタデータの連番を指定する

(i) version

リクエスタが使用可能な共通インタフェースのバージョン番号を指定する。バージョン番号を指定しない場合は、プロバイダが提供可能なバージョンのうち、最新のバージョンを指定したことと同様の動作となる。

リクエスタが指定したバージョン番号とプロバイダが返すバージョン番号の関係は、以下のとおり。

リクエストが指定するバージョン番号	プロバイダが返すバージョン番号
指定なし	提供可能なバージョンのうち最新のバージョン番号を返す
指定あり (プロバイダが提供可能なバージョン以下のバージョンを指定した場合)	指定されたバージョン以下での最新のバージョン番号を返す ・指定したバージョンが存在する場合は、指定したバージョン番号を返す ・指定したバージョン以下のバージョンが存在しない場合は、提供可能なバージョンのうち最も古いバージョン番号を返す
指定あり (プロバイダが提供可能なバージョンより大きいバージョンを指定した場合)	提供可能なバージョンのうち最新のバージョン番号を返す

【書式】

共通インタフェースバージョン番号は、小数点一桁までの数字で定義される。

(ii) service

要求するサービスの識別子を指定する。

【書式】

service は、以下のいずれかの値によって定義される。

- 地理情報 “WMS”
- 地名辞典(空間参照系管理) “SRS”
- 地名辞典(地名辞典管理) “Gazetteer”
- メタデータ検索 “CSW”

(iii) request

要求名は、” GetCapabilities ” でなければならない。

【書式】

“GetCapabilities”を指定する。

(iv) format

レスポンスの書式を指定する。省略された場合は、” text/xml ” とみなされる。

【書式】

format は、有効な MIME-Type の文字列で定義される。

(v) updateSequence

リクエストがサービスメタデータをキャッシュしていた場合、キャッシュしているメタデータの連番を指定することで、プロバイダとの差異を確認する。サービスメタデータの連番を指定しない場合は、プロバイダが提供可能なバージョンのうち、最新の連番を指定したことと同様の動作となる。リクエストが指定した連番とプロバイダが返す連番などの関係は、以下のとおり。

リクエストが指定する連番	プロバイダが返す連番
指定なし	最新の連番を返す (最新のサービスメタデータを返す)
指定あり (プロバイダ保有の連番と同じの場合)	最新の連番を返す (NULL 空のサービスメタデータを返す)
指定あり (プロバイダ保有の連番以下の場合)	最新の連番を返す (最新のサービスメタデータを返す)
指定あり (プロバイダ保有の連番より大きい場合)	例外処理: 指定した連番は無効

【書式】

サービスメタデータの連番は、整数値で定義される。

4) レスポンスの内容

以下のとおりである。

型	説明
ServiceMetadata 構造体	要求内容に沿ったサービスメタデータ型のデータを送信する

サービスメタデータの内容は、構造体「ServiceMetadata」の型として送信される。ServiceMetadata は、要求するサービスの種類により異なる。以下に、本ガイドラインで規定するサービスの種類に対応する ServiceMetadata 構造体を示す。

サービスの種類	識別子	応答される ServiceMetadata 構造体
地理情報	WMS	WMSServiceMetadata
地名辞典(空間参照系管理)	SRS	SRSServiceMetadata
地名辞典(地名辞典管理)	Gazetteer	GazetteerServiceMetadata
メタデータ検索	CSW	CSWSERVICEMetadata

5) 例外処理

リクエストしたサービスメタデータの連番が、プロバイダで保有する連番よりも大きい場合など(3-2-5-6.1)に示した InvalidUpdateSequence を利用する)。

4-2-3. 地理情報に関するインタフェース

地理情報に関するインタフェースにおける SOAP と WSDL 利用の実装仕様は、本ガイドラインにおいては特に定めない。

本ガイドラインにおいては、地理情報のインタフェースに関しては、4-1-1 に示すとおり、ISO19128 に基づく POST/GET 利用の実装を推奨する。

4-2-4. 地名辞典(空間参照系管理)に関するインタフェース

4-2-4-1. SearchLRS

1) 機能説明

地理識別子による空間参照系を検索する。空間参照系が1件もヒットしなかった場合は要素数0の配列を返す。

2) 利用条件

特になし。

3) リクエストの内容

リクエストパラメータは、以下のとおりである。

型	名前	入力規則	説明
String	name	指定しない場合は Null を入力する	名称
GeographicExtent	domainOfValidity	指定しない場合は Null を入力する	地理的範囲
String	theme	指定しない場合は Null を入力する	分類
String	overallOwner	指定しない場合は Null を入力する	管理者
String[]	locationType	必須	関連する場所型の名称 複数指定された場合は AND 検索を行う
AuthResult	authResult	指定しない場合は Null を入力する	認証結果 ユーザ認証を必要としないサービスの場合は null を指定する

(i) name

地理識別子による空間参照系の名称を指定する。

【書式】

特になし。文字列で定義される。

(ii) domainOfValidity

地理識別子による空間参照系が含まれる地理的範囲の情報を指定する。

【書式】

GeographicExtent 構造体を用いて、東西南北の境界座標を定義する。

(iii) theme

地理識別子による空間参照系が持つ分類（主題）を指定する。

【書式】

特になし。文字列で定義される。

(iv) overallOwner

地理識別子による空間参照系の管理者の名称を指定する。

【書式】

特になし。文字列で定義される。

(v) locationType

地理識別子による空間参照系が持つ場所型の名称を指定する。複数指定可能であり、複数指定は配列として指定する。複数指定された場合は、AND 検索となる。

【書式】

場所型の名称（String）の配列として定義される。

(vi) authResult

認証結果を指定する。

ユーザ認証を必要としないサービスの場合は null を指定する。

【書式】

AuthResult 構造体によって定義される。

4) レスポンスの内容

以下のとおりである。

型	説明
LRS 構造体の配列	リクエスト内容に応じた地理識別子による空間参照系のリストを送信する

【関連する構造体】

- GeographicExtent 構造体
- AuthResult 構造体

5) 例外処理

定義域以外の値が指定された場合など（3-2-5-6 に示す構造体により報告する）。

4-2-4-2. GetLRS

1) 機能説明

地理識別子による地理識別子による空間参照系を取得する

2) 利用条件

特になし。

3) リクエストの内容

リクエストパラメータは、以下のとおりである。

型	名前	入力規則	説明
String	name	指定しない場合は Null を入力する	名称
AuthResult	authResult	指定しない場合は Null を入力する	認証結果 ユーザ認証を必要としないサービスの場合は null を指定する

(i) name

地理識別子による空間参照系の名称を指定する。

【書式】

特になし。文字列で定義される。

(ii) authResult

認証結果を指定する。

ユーザ認証を必要としないサービスの場合は null を指定する。

【書式】

AuthResult 構造体によって定義される。

4) レスポンスの内容

以下のとおりである。

型	説明
LRS 構造体	リクエスト内容に応じた地理識別子による空間参照系を送信する

【関連する構造体】

➤ AuthResult 構造体

5) 例外処理

定義域以外の値が指定された場合など（3-2-5-6 に示す構造体により報告する）。

4-2-4-3. SearchLocationType

1) 機能説明

登録されている場所型を検索する。

Irs に地理識別子による空間参照系の名称を指定することで、特定の空間参照系に関連する場所型を検索することが可能である。

また、場所型の Nesting 関連の条件を指定する場合は、parentType または childType に関連する場所型の名称を指定する。

地理的範囲が指定された場合は、場所型が持つ地理的範囲と一部でも重なるものを返す。

条件に合致する場所型が存在しない場合にはサイズ 0 の配列を返す。

2) 利用条件

特になし。

3) リクエストの内容

リクエストパラメータは、以下のとおりである。

型	名前	入力規則	説明
String	name	指定しない場合は Null を入力する	名称
String	theme	指定しない場合は Null を入力する	分類
PIType	identification	必須	PI タイプ
GeographicExtent	territoryOfUse	指定しない場合は Null を入力する	地理的範囲
String	owner	必須	管理者
String[]	Irs	指定しない場合は Null を入力する	関連する空間参照系の名称
String[]	parentType	指定しない場合は Null を入力する	親の場所型の名称
String[]	childType	指定しない場合は Null を入力する	子の場所型の名称
AuthResult	authResult	指定しない場合は Null を入力する	認証結果

(i) name

場所型の名称を指定する。一部のみの指定も可能である。名称を特定しない場合は null を指定する。

【書式】

場所型の名称は、文字列型によって定義される。

(ii) theme

場所型の分類を指定する。一部のみの指定も可能である。分類を特定しない場合は null を指定する。

【書式】

場所型の分類は、文字列型によって定義される。

(iii) identification

PI タイプの名称を指定する。PI タイプの各要素の値は一部のみの指定が可能。

【書式】

PI タイプの名称は、PIType 型によって定義される。

(iv) territoryOfUse

場所型が含まれる地理的範囲を指定する。

【書式】

地理的範囲は、GeographicExtent 型によって定義される。

(v) owner

場所型の管理者の名称を指定する。

【書式】

管理者の名称は、文字列型によって定義される。

(vi) lrs

場所型が関連する地理識別子による空間参照系の名称を指定する。空間参照系を特に限定しない場合は、null を指定する。複数の名称を指定できるが、複数指定した場合は OR 検索となる。

【書式】

地理識別子による空間参照系の名称は、文字列型の配列として定義される。

(vii) parentType

親子関連の親に相当する場所型の名称を指定する。複数指定可能であるが、親関連の条件を指定しない場合は、null を指定する。

【書式】

親の場所型の名称は、文字列型の配列として定義される。

(viii) childType

親子関連の子に相当する場所型の名称を指定する。複数指定可能であるが、子関連の

条件を指定しない場合は、null を指定する。

【書式】

子の場所型の名称は、文字列型の配列として定義される。

(ix) authResult

認証結果を指定する。ユーザ認証を必要としないサービスの場合は、null を指定する。

【書式】

認証結果は、authResult 構造体により定義される。

4) レスポンスの内容

以下のとおりである。

型	説明
LocationType 構造体の配列	リクエスト内容に合致する場所型を、LocationType 構造体として返す。複数の場所型が合致した場合は、LocationType 構造体の配列として返す。

リクエスト時、引数に地理的範囲が指定された場合は、場所型が持つ地理的範囲と一部でも重なるものを返す。

条件に合致する場所型が存在しない場合にはサイズ 0 の配列を返す。

【関連する構造体】

- PIType 構造体
- GeographicExtent 構造体
- LocationType 構造体
- AuthResult 構造体

5) 例外処理

定義域以外の値が指定された場合など（3-2-5-6 に示す構造体により報告する）。

4-2-4-4. GetLocationType

1) 機能説明

登録されている場所型を取得する。

2) 利用条件

- ・ 場所型の名称が明らかである場合に利用する。場所型の名称が明らかでない場合は、SearchLocationType 関数を利用する。

3) リクエストの内容

リクエストパラメータは、以下のとおりである。

型	名前	入力規則	説明
String	name	必須	名称
AuthResult	authResult	指定しない場合は Null を入力する	認証結果

(i) name

場所型の名称を指定する。

【書式】

場所型の名称は、文字列型によって定義される。

(ii) authResult

認証結果を指定する。ユーザ認証を必要としないサービスの場合は、null を指定する。

【書式】

認証結果は、authResult 構造体により定義される。

4) レスポンスの内容

以下のとおりである。

型	説明
LocationType 構造体の配列	リクエスト内容に合致する場所型を、LocationType 構造体として返す。

条件に合致する場所型が存在しない場合にはサイズ 0 の配列を返す。

【関連する構造体】

- LocationType 構造体
- AuthResult 構造体

5) 例外処理

定義域以外の値が指定された場合など（3-2-5-6 に示す構造体により報告する）。

4-2-4-5. SearchLocationTypeAssociation

1) 機能説明

登録されている場所型関連を検索する。

2) 利用条件

特になし。

3) リクエストの内容

リクエストパラメータは、以下のとおりである。

型	名前	入力規則	説明
String	childLocationType	parentLocationType を指定しない場合は必須	子となる場所型関連の名称
String	parentLocationType	childLocationType を指定しない場合は必須	親となる場所型関連の名称
AuthResult	authResult	指定しない場合は Null を入力する	認証結果

(i) childLocationType

子となる場所型関連の名称を指定する。parentLocationType を指定しない場合は、必須である。

【書式】

子場所型は、文字列型として定義される。

(ii) parentLocationType

親となる場所型関連の名称を指定する。childLocationType を指定しない場合は、必須である。

【書式】

親場所型は、文字列型として定義される。

(iii) authResult

認証結果を指定する。ユーザ認証を必要としないサービスの場合は、null を指定する。

【書式】

認証結果は、AuthResult 構造体により定義される。

4) レスポンスの内容

以下のとおりである。

型	説明
LocationTypeAssociation 構造体	子場所型を指定しない場合は親場所型で指定された場所型のすべての子場所型を返す。 親場所型を指定しない場合は子場所型で指定された場所型のすべての親場所型を返す。

【関連する構造体】

- LocationTypeAssociation 構造体
- AuthResult 構造体

5) 例外処理

定義域以外の値が指定された場合など（3-2-5-6 に示す構造体により報告する）。

4-2-5. 地名辞典(地名辞典管理)に関するインタフェース

4-2-5-1. SearchPI

1) 機能説明

登録されている場所インスタンスの PI 部分を検索する。

地理的範囲で地理的境界またはポリゴンを指定した場合は、一部でも地理的範囲が交差する PI を検索する。

条件に合致する PI が存在しない場合はサイズ 0 の配列を返す。

(A) `routeCondition.cascadingLevel > 0` の場合

他の地名辞典管理サービスの `SearchPI` 関数を呼び出す場合は `routeCondition.cascadingLevel` の値を 1 だけ減じ、かつ `routeCondition.routeHistory` の末尾に自サーバを追加しなければならない。また、`routeCondition.routeHistory` で指定された呼び出し経路に自サービスが含まれている場合はサイズ 0 の配列を返す。

(B) `routeCondition.cascadingLevel = 0` の場合

他の地名辞典管理サービスの `SearchPI` 関数を呼び出すことはできない。また、`routeCondition.routeHistory` で指定された呼び出し経路に自サービスが含まれている場合はサイズ 0 の配列を返す。

(C) `routeCondition.cascadingLevel < 0` の場合

他の地名辞典管理サービスの `SearchPI` 関数は `routeCondition.routeDefinition` の順に呼び出さなければならない。

2) 利用条件

特になし。

3) リクエストの内容

リクエストパラメータは、以下のとおりである。

型	名前	入力規則	説明
RouteCondition	routeCondition	指定しない場合は Null を入力する	他サービス呼び出し条件
String	gazetteer	指定しない場合は Null を入力する	地名辞典の名称
String	locationType	指定しない場合は Null を入力する	場所型の名称
String[]	geographicIdentifier	指定しない場合は Null を入力する	地理識別子
GeographicExtent	geographicExtent	指定しない場合は Null を入力する	地理的範囲
TemporalExtent	temporalExtent	指定しない場合は Null を入力する	時間範囲
String	administrator	指定しない場合は Null を入力する	管理者
String[]	searchWord	指定しない場合は Null を入力する	検索語句
AuthResult	authResult	指定しない場合は Null を入力する	認証結果

(i) routeCondition

他サービス呼び出し条件を指定する。

【書式】

他サービス呼び出し条件は、RouteCondition 構造体によって定義される。

(ii) gazetteer

PI が含まれる地名辞典の名称を指定する。地名辞典を特定しない場合は、null を指定する。

【書式】

地名辞典の名称は、文字列型によって定義される。

(iii) locationType

PI が関連する場所型の名称を指定する。場所型を特定しない場合は、null を指定する。

【書式】

場所型の名称は、文字列型によって定義される。

(iv) geographicIdentifier

PI に含まれる地理識別子を指定する。複数指定可能であり、複数指定する場合は地配列として指定する。地理識別子を特定しない場合は、null を指定する。

【書式】

地理識別子は、文字列型の配列によって定義される。

(v) geographicExtent

PI が含まれる地理的範囲を指定する。地理的範囲を限定しない場合は、null を指定する。

【書式】

地理的範囲は、GeographicExtent 構造体により定義される。

(vi) temporalExtent

PI が持つ有効な時間範囲を指定する。時間範囲を限定しない場合は、null を指定する。

【書式】

時間範囲は、TemporalExtent 構造体により定義される。

(vii) administrator

PI の管理者の名称を指定する。管理者を特定しない場合は、null を指定する。

【書式】

管理者の名称は、文字列型により定義される。

(viii) searchWord

検索語句を指定する。検索語句は、場所インスタンスの PI の値または代替地理識別子を対象とした全文検索用フリーワードとして用いられる。語句が複数指定された場合は、地理識別子、または代替地理識別子のいずれかがすべての検索語句を含むものを検索する。

検索語句を設定しない場合は null を指定する。

【書式】

検索語句は、文字列型の配列として定義される。

(ix) authResult

認証結果を指定する。ユーザ認証を必要としないサービスの場合は、null を指定する。

【書式】

認証結果は、AuthResult 構造体により定義される。

4) レスポンスの内容

以下のとおりである。

型	説明
PI 構造体の配列	指定した条件に合致する PI を返す。複数合致した場合は、PI 構造体の配列として返す。

【関連する構造体】

- GeographicExtent 構造体
- TemporalExtent 構造体
- PI 構造体
- RouteCondition 構造体
- AuthResult 構造体

5) 例外処理

定義域以外の値が指定された場合など（3-2-5-6 に示す構造体により報告する）。

4-2-5-2. GetLocationInstance

1) 機能説明

登録されている場所インスタンスを取得する。

指定した PI に合致する場所インスタンスを取得する。

地名辞典を特定して場所インスタンスを取得することもできる。

PI が複数指定された場合は、指定された順に対応する場所インスタンスを返す。PI に対応する場所インスタンスが存在しない場合は、配列中の当該要素に null を格納して返す。

条件に合致する場所インスタンスが存在しない場合はサイズ 0 の配列を返す。

(A) routeCondition.cascadingLevel > 0 の場合

他の地名辞典管理サービスの GetLocationInstance 関数を呼び出す場合は routeCondition.cascadingLevel の値を 1 だけ減じ、かつ routeCondition.routeHistory の末尾に自サーバを追加しなければならない。また、routeCondition.routeHistory で指定された呼び出し経路に自サービスが含まれている場合はサイズ 0 の配列を返す。

(B) routeCondition.cascadingLevel = 0 の場合

他の地名辞典管理サービスの GetLocationInstance 関数を呼び出すことはできない。また、routeCondition.routeHistory で指定された呼び出し経路に自サービスが含まれている場合はサイズ 0 の配列を返す。

(C) routeCondition.cascadingLevel < 0 の場合

他の地名辞典管理サービスの GetLocationInstance 関数は routeCondition.routeDefinition の順に呼び出さなければならない。

2) 利用条件

特になし。

3) リクエストの内容

リクエストパラメータは、以下のとおりである。

型	名前	入力規則	説明
RouteCondition	routeCondition	指定しない場合は Null を入力する	他サービス呼び出し条件
String	gazetteer	指定しない場合は Null を入力する	地名辞典の名称
PI[]	pi	指定しない場合は Null を入力する	PI
AuthResult	authResult	指定しない場合は Null を入力する	認証結果

(i) routeCondition

他サービス呼び出し条件を指定する。

【書式】

他サービス呼び出し条件は、RouteCondition 構造体によって定義される。

(ii) gazetteer

PI が含まれる地名辞典の名称を指定する。地名辞典を特定しない場合は、null を指定する。

【書式】

地名辞典の名称は、文字列型によって定義される。

(iii) pi

PI を指定する。複数の PI を指定することができ、その場合は PI の配列を指定する。PI を特定しない場合は、null を指定する。

【書式】

PI は、PI 構造体によって定義される。

(iv) authResult

認証結果を指定する。ユーザ認証を必要としないサービスの場合は、null を指定する。

【書式】

認証結果は、AuthResult 構造体により定義される。

4) レスポンスの内容

以下のとおりである。

型	説明
LocationInstance 構造体の配列	検索条件に合致する場所インスタンスを、LocationInstance 構造体の配列として返す。

【関連する構造体】

- PI 構造体
- LocationInstance 構造体
- RouteCondition 構造体
- AuthResult 構造体

5) 例外処理

定義域以外の値が指定された場合など（3-2-5-6 に示す構造体により報告する）。

4-2-5-3. SearchLocationInstanceAssociation

1) 機能説明

複数場所インスタンス関連を検索する。

条件に合致する場所インスタンス関連が存在しない場合はサイズ 0 の配列を返す。

(A) routeCondition.cascadingLevel > 0 の場合

他の地名辞典管理サービスの SearchLocationInstanceAssociation 関数を呼び出す場合は routeCondition.cascadingLevel の値を 1 だけ減じ、かつ routeCondition.routeHistory の末尾に自サーバを追加しなければならない。また、routeCondition.routeHistory で指定された呼び出し経路に自サービスが含まれている場合はサイズ 0 の配列を返す。

(B) routeCondition.cascadingLevel = 0 の場合

他の地名辞典管理サービスの SearchLocationInstanceAssociation 関数を呼び出すことはできない。また、routeCondition.routeHistory で指定された呼び出し経路に自サービスが含まれている場合はサイズ 0 の配列を返す。

(C) routeCondition.cascadingLevel < 0 の場合

他の地名辞典管理サービスの SearchLocationInstanceAssociation 関数は routeCondition.routeDefinition の順に呼び出さなければならない。

Nesting 関連を検索する場合は source を child に、target を parent に読み替える。

2) 利用条件

特になし。

3) リクエストの内容

リクエストパラメータは、以下のとおりである。

型	名前	入力規則	説明
RouteCondition	routeCondition	必須	他サービス呼び出し条件
String	sourceGazetteer	指定しない場合は Null を入力する	関連元場所インスタンスの地名辞典の名称
String	sourceGeographicIdentifier	指定しない場合は Null を入力する	関連元場所インスタンスの識別子
String	targetGazetteer	指定しない場合は Null を入力する	関連先場所インスタンスの地名辞典の名称
String	targetGeographicIdentifier	指定しない場合は Null を入力する	関連先場所インスタンスの地理識別子
String[]	phase	必須	位相の名称
AuthResult	authResult	指定しない場合は Null を入力する	認証結果

(i) routeCondition

他サービス呼び出し条件を指定する。

【書式】

他サービス呼び出し条件は、RouteCondition 構造体によって定義される。

(ii) sourceGazetteer

関連元場所インスタンスの地名辞典の名称を指定する。関連元場所インスタンスの地名辞典を特定しない場合は null を指定する。

【書式】

関連元場所インスタンスの地名辞典の名称は、文字列型として定義される。

(iii) sourceGeographicIdentifier

関連元場所インスタンスが含む地理識別子を指定する。関連元場所インスタンスを特定しない場合は null を指定する。

【書式】

関連元場所インスタンスが含む地理識別子は、文字列型として定義される。

(iv) targetGazetteer

関連先場所インスタンスの地名辞典の名称を指定する。関連先場所インスタンスの地名辞典を特定しない場合は null を指定する。

【書式】

関連先場所インスタンスの地名辞典の名称は、文字列型として定義される。

(v) targetGeographicIdentifier

関連先場所インスタンスが含む地理識別子を指定する。関連先場所インスタンスを特定しない場合は null を指定する。

【書式】

関連先場所インスタンスが含む地理識別子は、文字列型として定義される。

(vi) phase

位相の名称を指定する。個々の値には、

“ Nesting ” または
“ Equivalent ” を指定する。
複数指定した場合は OR 検索を行う。

【書式】

位相の名称は複数指定することができ、文字列型の配列として定義される。

(vii) authResult

認証結果を指定する。ユーザ認証を必要としないサービスの場合は、null を指定する。

【書式】

認証結果は、AuthResult 構造体により定義される。

4) レスポンスの内容

以下のとおりである。

型	説明
LocationInstanceAssociation 構造体の配列	検索条件に合致する場所インスタンス関連を、LocationInstanceAssociation 構造体の配列として返す。 検索条件に合致する場所インスタンス関連が存在しない場合は、サイズ0の配列を返す。

【関連する構造体】

- LocationInstanceAssociation 構造体
- RouteCondition 構造体
- AuthResult 構造体

5) 例外処理

定義域以外の値が指定された場合など（3-2-5-6 に示す構造体により報告する）。

4-2-6. メタデータ検索に関するインタフェース

メタデータ検索に関するインタフェースにおける SOAP と WSDL 利用の実装仕様は、OGC Catalogue Service Specification において規定されている、HTTP を通信プロトコルとした実装仕様である Catalogue Services for the Web (CSW) に準ずる。SOAP と WSDL 利用のためのインタフェースを実装する場合には、CSW のレスポンスの規則、リクエストパラメータの規則、各関数の記述規則等に従うものとする。

本ガイドラインで規定する CSW のプロトコルバージョンは、「2.0.0」とする。

5. 関数仕様作成に関する取り決め

5-1. 関数仕様作成に関する取り決め

アプリケーションプログラムを開発する場合、そのアプリケーションプログラムで使用する関数は、本ガイドラインの「標準インタフェース関数仕様」で定める関数とする。

アプリケーションプログラムで使用する関数が、本ガイドラインの「標準インタフェース関数仕様」で定める関数に該当しない場合には、当該アプリケーションプログラムの開発者は、使用する関数について関数仕様を作成する。

関数仕様は、以下の2種類を作成する。

- (1) 使用する関数一覧
- (2) 各関数の説明文書

【解説】

アプリケーションプログラムの開発者は、当該アプリケーションプログラムで使用する関数が本ガイドラインの「標準インタフェース関数仕様」で定める関数に該当しない場合に、関数の概要を説明する「使用する関数一覧」および関数の詳細を説明する「各関数の説明文書」の2種類を作成しなくてはならない。

(1) 使用する関数の一覧

アプリケーションプログラムが使用する関数の一覧を作成する。一覧表には、使用する各関数について「機能分類」「機能」「内容」「関数名」を記載しなくてはならない(表 5-1)。記載例を表 5-2 に示す。

表 5-1 使用する関数一覧の記載項目

項目	定義	データ型	要求度/条件	最大記述回数
関数	アプリケーションプログラムで使用する関数	クラス	必須	N
機能分類	機能について、それをいくつかのまとまりでとりまとめたもの	文字列	任意	1
機能	関数が行う処理を簡潔に説明したもの	文字列	必須	1
内容	関数が行う処理内容の概要	文字列	必須	1
関数名	関数の名前	文字列	必須	1

表 5-2 使用する関数一覧の記載例

機能分類	機能	内容	関数名
初期設定	サービスメタデータ取得	利用可能な操作の情報を取得する	GetCapabilities
...

(2) 各関数の説明文書

一覧として作成したアプリケーションプログラムが使用する各関数について、以下の項目を記載した説明文書を作成する。

- 関数名
- 説明
- 引数
- 戻り値
- 関連項目

記載事項の詳細を表 5-3 に示す。関数の引数および戻り値の詳細は、表 5-4 および表 5-5 のとおりである。記載例を表 5-6 に示す。

表 5-3 各関数の説明文書記載事項

項目	定義	データ型	要求度/条件	最大記述回数
関数説明文書	アプリケーションプログラムで使用する各関数についての説明	クラス	必須	N
関数名	関数の名称	文字列	必須	1
説明	その関数が行う処理内容の概要	文字列	必須	1
引数	関数を呼び出すときに相手に渡す値。呼び出された側では、この引数に応じて処理を行う	クラス 表 5-4	必須	N
戻り値	関数が処理を終了し、処理の結果として返す値	クラス 表 5-5	必須	1
関連項目	当該関数で参照する他の関数	文字列	任意(引数と戻り値で、構造体を用いる場合)	N

表 5-4 各関数の説明文書記載事項（引数）

項目	定義	データ型	要求度/条件	最大記述回数
引数	関数を呼び出すときに相手に渡す値。呼び出された側では、この引数に応じて処理を行う	クラス	必須	N
引数の型	引数のデータ型	文字列	必須	1
引数の名前	引数の名称	文字列	必須	1
引数の説明	引数が意味する内容、引数の定義域	文字列	必須	1
引数の要求度/条件	当該関数に対し、当該引数が必須であるか任意であるかの区別。また、その場合の条件。	文字列	必須	1

表 5-5 各関数の説明文書記載事項（戻り値）

項目	定義	データ型	要求度/条件	最大記述回数
戻り値	関数が処理を終了し、処理の結果として返す値	クラス	必須	1
戻り値の型	戻り値の型	文字列	必須	1
戻り値の説明	戻り値が意味する内容	文字列	必須	1

表 5-6 関数の説明文書の記載例

名称	GetCapabilities		
機能	サービスメタデータを取得する		
引数	型	名前	説明
	String	version	準拠する標準インタフェースのバージョン番号 該当するバージョンのサービスメタデータのみを取得する 指定なしの場合は、最新バージョン
	String	service	サービスタイプ 要求するサービスの識別子 例) 地理情報 “WMS” 地名辞典(空間参照系管理) “SRS” 地名辞典(地名辞典管理) “Gazetteer” メタデータ検索 “CSW”
	String	request	要求名 ”GetCapabilities”でなければならない
	String	format	サービスメタデータの出カフォーマット 応答をどのようなフォーマット(MIME Type)で返すかを指定する 省略された場合は、”text/xml”とみなされる
	String	updateSequence	サービスメタデータの連番 サービス呼び出し側がこの連番に該当するサービスメタデータを キャッシュしている場合に指定する
戻り値	ServiceMetadata	サービスメタデータ	
関連項目	ServiceMetadata 構造体 (※引数”service”において指定したサービスタイプにより、構造体の内容は異なる)		

5-2. 関数と引数の命名規則

関数仕様を作成する場合の関数と引数の命名規則を、以下のとおり取り決める。

関数の命名規則

- 識別子の名前は英語を基本とする。
- 動詞または動詞句とする。
- 識別子の最初の文字と、後に続いて連結されている各単語の最初の文字を大文字にする。

引数の命名規則

- 識別子の名前は英語を基本とする。
- 識別子の最初の文字は小文字にし、後に続いて連結されている各単語の最初の文字を大文字とする(Camel 形式)。
- 原則として、引数の型を説明する名前ではなく、引数の意味を説明する名前を使用する。

【解説】

アプリケーションプログラムの開発者は、当該アプリケーションプログラムで使用する関数が本ガイドラインの「標準インタフェース関数仕様」で定める関数(第3章参照)に該当しない場合に、使用する関数について関数仕様を作成する(5-1節参照)。作成する関数仕様は、本ガイドラインで定める関数仕様と同様に、他のアプリケーション開発者が利用する。したがってアプリケーション開発者が混乱しないよう、その名称を、同じ規則で命名する必要がある。従って、関数と引数についての命名規則を以下のとおり取り決める。

関数、引数とも、命名は英語を基本とする。

関数については、動詞または動詞句とする。特に、処理内容が相対的な関係にある関数の命名にあたっては、以下の英語の相対性に留意する。

- Add / Remove
- Insert / Delete
- Get / Set
- Start / Stop
- Begin / End
- Send / Receive
- First / Last
- Get / Release
- Put / Get

- **Up / Down**
- **Show / Hide**
- **Source / Target**
- **Open / Close**
- **Source / Destination**
- **Increment / Decrement**
- **Lock / Unlock**
- **Old / New**
- **Next / Previous**

6. 実装仕様作成に関する取り決め

6-1. 実装仕様作成に関する取り決め

アプリケーションプログラムで使用する関数が、本ガイドラインの「標準インタフェース関数仕様」で定める関数に該当しない場合には、当該アプリケーションプログラムの開発者は、「標準インタフェース関数仕様作成に関する取り決め」に従い作成した関数仕様に対応する実装仕様を作成する。

この場合の実装は、HTTP による SOAP と WSDL を利用した実装とする。

実装仕様は、以下の 2 種類を作成する。

- (1) 実装仕様の概要
- (2) WSDL による記述文書

【解説】

アプリケーションプログラムの開発者は、当該アプリケーションプログラムで使用する関数が、本ガイドラインの「標準インタフェース関数仕様」で定める関数（第 3 章参照）に該当しない場合には、「標準インタフェース関数仕様作成に関する取り決め」に従い関数仕様を作成する（5 章参照）。当該アプリケーションプログラムの開発者は、さらに作成した関数仕様に対応する実装仕様を作成しなければならない。

実装仕様は、人間にとって可読性の高い「実装仕様の概要」および機械にとって可読性の高い「WSDL による説明文書」の 2 種類からなる。作成する実装仕様の詳細を次に示す。

(1) 実装仕様の概要

機能の説明や処理の内容について、文書形式により作成する。作成する実装仕様は、本ガイドラインで示す実装仕様と同様の書式とする。実装仕様の概要として記載しなければならない項目とその内容を、表 6-1 に示す。

表 6-1 実装仕様の概要の記載項目

項目	定義	データ型	要求度/条件	最大記述回数
関数名	実装仕様を記述する関数の名前	文字列	必須	1
機能説明	関数の処理内容の詳細説明	文字列	必須	1
利用条件	関数を利用する場合に必要な条件	文字列	必須	1
リクエストの内容	リクエストに使用するパラメータの説明	文字列	必須	1
レスポンスの内容	レスポンスで受け渡される情報の説明	文字列	必須	1
例外処理	無効なリクエストが発行された場合やサービスが実装されていない場合などの例外処理の発生時の説明と、その処理内容	文字列	必須	1

(2) WSDL による記述文書

SOAP でのメッセージの受け渡しを考慮し、WSDL の仕様に基づき WSDL 文書を作成する。

WSDL 文書の作成単位は、アプリケーションプログラム単位とする。

7. 標準インタフェースの運用に関する取り決め

7-1. インタフェース公開・管理方法

アプリケーションプログラムの開発者は、開発するアプリケーションプログラムが他のアプリケーションプログラムから接続できるよう、インタフェースを公開する。公開するインタフェースは、本ガイドラインに基づき作成する。

アプリケーションプログラムの開発者は、開発するアプリケーションプログラムについて、アプリケーションプログラムのインタフェースに関する以下の2種類の情報を記載した文書を作成する。

(1) アプリケーションプログラムの内容

- アプリケーションプログラムの目的と、機能およびデータの概要
- 利用にあたっての制限や条件
- 接続方法
- 利用許可・利用申請のための問合せ先

(2) アプリケーションプログラムで利用できる関数等の技術的な仕様

これらの文書は、別途標準インタフェースの管理機関において公開する。

【解説】

本ガイドラインに従って開発するアプリケーションプログラムを有効に利用するために、アプリケーションプログラムの開発者は、開発するアプリケーションプログラムが他のアプリケーションプログラムから接続できるよう、本ガイドラインに基づくインタフェースを実装し、それを公開する。

公開するインタフェースに関する情報は、以下の用途に利用する。

- アプリケーションプログラムを利用したい機関が、利用可能なアプリケーションプログラムの有無を検索する。
- 利用にあたって、アプリケーションプログラムに関する概要ならびに開発に必要な技術情報を入手する。

アプリケーションプログラムの開発者は、公開するインタフェースを他者が利用するために、アプリケーションプログラムのインタフェースに関する情報を文書として、アプリケーションプログラムの概要を説明する「アプリケーションプログラムの内容」およびアプリケーションプログラムの技術的な内容を説明する「アプリケーションプログラムで利用できる関数等の技術的な仕様」の2種類を作成しなければならない。作成した文書は、インタフェースを利用しようとする機関がその情報を入手できるよう公開する。

公開する情報は、表 7-1 に示す情報とする。その仕様は、公開・管理仕様（8章）に示す。

表 7-1 公開する情報

公開する情報		説明
(1) アプリケーションプログラムの内容	アプリケーションプログラムの目的と、機能およびデータの概要	<ul style="list-style-type: none"> ■ アプリケーションプログラム作成の目的 ■ アプリケーションプログラムで提供可能な機能 ■ アプリケーションプログラムで取り扱うデータの種類、データの範囲に関する情報
	利用にあたっての制限や条件	<ul style="list-style-type: none"> ■ プログラムやデータに関する利用上の制限 <ul style="list-style-type: none"> □ 著作権保護に関する情報 □ 使用料に関する情報 ■ アプリケーションプログラムを利用するための物理的な環境に関する情報 <ul style="list-style-type: none"> □ 利用可能な時間に関する情報 □ 利用可能な期間に関する情報 □ 利用可能なデータの範囲に関する情報 □ ネットワーク環境に関する情報
	接続方法	<ul style="list-style-type: none"> ■ 接続ポイントに関する情報(URL など)
	利用許可・利用申請のための問合せ先	<ul style="list-style-type: none"> ■ アプリケーションプログラムを利用するにあたり、利用許可や利用申請を行うために必要となる問合せ先や問合せ方法に関する情報 <ul style="list-style-type: none"> □ アプリケーションプログラムの管理者の組織名称・担当者役職・問合せ方法・問合せ可能時間・電話番号・住所等 □ アプリケーションプログラムの開発者の組織名称・担当者役職・問合せ方法・問合せ可能時間・電話番号・住所等 □ アプリケーションプログラムやデータの著作権者の組織名称・担当者役職・問合せ方法・問合せ可能時間・電話番号・住所等
(2) アプリケーションプログラムで利用できる関数等の技術的な仕様	<ul style="list-style-type: none"> ■ アプリケーションプログラムが対応している関数・引数・戻り値に関する情報(関数仕様) ■ アプリケーションプログラム間でそれらがやり取りされる形式に関する情報(実装仕様) ■ 取得可能なデータの情報 	

7-2. インタフェース接続方式

他のアプリケーションプログラムと接続する場合は、以下の手順で接続する。

- (1) 標準インタフェース登録機関において、利用可能なアプリケーションプログラムを検索する。
- (2) 利用するアプリケーションプログラムについて、そのアプリケーションプログラムの管理者に利用の申請を行う。必要に応じて、利用するための各種条件の処理を行う。
- (3) 利用に必要な関数仕様・実装仕様等の技術資料を入手し、アプリケーションプログラムを利用するためのインタフェースを実装する。
- (4) アプリケーションプログラムに接続し、利用する。

アプリケーションプログラム間の接続は、本ガイドラインで定めるインタフェースを介して行う。

接続に用いる通信プロトコルは、HTTP とし、以下のうちいずれかの方式とする。接続するアプリケーションプログラムの実装仕様に従い、接続を行う。

方式1) POST/GET による方法

方式2) SOAP による方法

【解説】

他のアプリケーションプログラムと接続する場合には、まず既に登録してあるアプリケーションプログラムに関する情報を検索し、利用可能なアプリケーションプログラムが存在するかどうか確認を行う。利用可能なアプリケーションプログラムが見つかった場合には、あわせて登録してあるアプリケーションプログラムの管理者情報を基に、そのアプリケーションプログラムの管理者に対して接続・利用申請を行う。利用にあたっての条件等が解決し、利用に問題がなければ、公開されている技術的な仕様を基に、接続のためのインタフェースを実装する。

なお、アプリケーションプログラム間の接続に用いる通信プロトコルは HTTP とする。アプリケーションプログラムの開発者は、方式1あるいは方式2を採用し、アプリケーションプログラムを接続する必要がある。アプリケーションプログラムの開発者は、以下のいずれかの方式で実装する。

方式1) POST/GET による方法

方式2) SOAP による方法

8. 標準インタフェース公開・管理仕様

標準インタフェース公開・管理仕様として、以下の2種類の文書を定義する。

- 公開・管理のためのアプリケーションプログラムに関する文書
- 公開・管理のためのアプリケーションプログラムで利用できる関数等の技術的文書

【解説】

標準インタフェース公開・管理仕様では、以下の2種類の文書を定義する。アプリケーションプログラムの開発者は、公開するインタフェースを他者が利用するために、これらの2種類を作成しなければならない。

- 公開・管理のためのアプリケーションプログラムに関する文書
- 公開・管理のためのアプリケーションプログラムで利用できる関数等の技術的文書

「公開・管理のためのアプリケーションプログラムに関する文書」は、アプリケーションプログラムの目的、機能、取り扱うデータ、利用時の制限や条件、接続方法、および問合せ先について、その概要を明らかにするための文書である。

「公開・管理のためのアプリケーションプログラムで利用できる関数等の技術的文書」は、アプリケーションプログラムがどのようなインタフェースを公開しているのか、具体的な関数とそれに関する引数等の条件を明らかにするための文書である。

1つのインタフェース公開情報には1つの実装方式のみを記述する。たとえば、HTTP POST/GET と SOAP (セッションあり、セッションなし) をすべて実装するアプリケーションプログラムは、合計3つのインタフェース公開情報を作成する。

8-1. アプリケーションプログラムの内容に関する公開・管理仕様

公開・管理のためのアプリケーションプログラムの内容に関する文書は、以下の内容を含む。

- (1) アプリケーションプログラムの目的と、機能およびデータの概要
- (2) 利用にあたっての制限や条件
- (3) 接続方法
- (4) 利用許可・利用申請のための問合せ先

【解説】

上述 4 項目の詳細を以下の (1) から (4) で示す。この 4 つの項目は公開・管理仕様として原則必須であるが、(4) についてはアプリケーションプログラムの開発者のみ必須である。

(1) アプリケーションプログラムの目的と、機能およびデータの概要

「アプリケーションプログラムの目的と、機能およびデータの概要」として記載する項目は、「名称」「概要」「目的」「データの種類」「データの範囲」である。詳細を表 8-1 に示す。

表 8-1 アプリケーションプログラムの目的と機能およびデータの概要記載項目

項目	定義	データ型	要求度/ 条件	最大記述 回数
アプリケーションプログラムの目的と、機能およびデータの概要	アプリケーションプログラムで実現可能な機能等の概要	クラス	必須	1
名称	アプリケーションプログラムの名称	文字列	必須	1
概要	アプリケーションプログラムの機能の概要	文字列	必須	1
目的	アプリケーションプログラムの目的	文字列	必須	1
データの種類	アプリケーションプログラムで取り扱うデータの種類	文字列	必須	N
データの範囲	アプリケーションプログラムで取り扱うデータの空間範囲	文字列	必須	1

(2) 利用にあたっての制限や条件

「利用にあたっての制限や条件」として記載する項目は、「サービス開始時期」「サービス終了予定時期」「運用時間」「著作権」「費用」である。これらの詳細を、表 8-2 に示す。

表 8-2 利用にあたっての制限や条件記載項目

項目	定義	データ型	要求度/ 条件	最大記述 回数
利用にあたっての制限や条件	アプリケーションプログラムを利用するにあたっての制限事項や条件	クラス	必須	1
サービス開始時期	アプリケーションプログラムの利用を開始する時期	文字列	任意	1
サービス終了予定時期	アプリケーションプログラムの利用を終了する時期(予定)	文字列	任意	1
運用時間	アプリケーションプログラムの利用可能な時間	文字列	必須	N
著作権	アプリケーションプログラムとアプリケーションプログラムで取り扱うデータの著作権	文字列	必須	1
費用	アプリケーションプログラムの利用に必要な費用(プログラムやデータの使用料など)	文字列	任意	1

(3) 接続方法

「接続方法」として記載する項目は、「URL」である。詳細を表 8-3 に示す。

表 8-3 接続方法記載項目

項目	定義	データ型	要求度/ 条件	最大記述 回数
接続方法	接続に必要な URL	クラス	必須	1
URL	接続先の URL アドレス	文字列	必須	1

(4) 利用許可・利用申請のための問合せ先

「利用許可・利用申請のための問合せ先」として記載する項目は、「問合せ区分」「問合せ先名称」「住所」「電話番号」「ファックス番号」「電子メールアドレス」および「問合せ条件」である。詳細を表 8-4 に示す。

表 8-4 利用許可・利用申請のための問合せ先記載項目

項目	定義	データ型	要求度/条件	最大記述回数
利用許可・利用申請のための問合せ先	アプリケーションプログラムの管理者等の問合せ先情報	クラス	アプリケーションプログラムの管理者は必須	N
問合せ先区分	問合せ先の区分 以下のいずれかを記入する。 <ul style="list-style-type: none"> ✓ アプリケーションプログラムの管理者 ✓ アプリケーションプログラムの開発者 ✓ 著作権者(著作権者の場合には、何に対して著作権を保持しているか記述する。 	文字列	必須	1
問合せ先名称	管理者や開発者等の組織名、または個人名	文字列	必須	1
住所	問合せ先の住所	文字列	任意	1
電話番号	問合せ先の電話番号	文字列	必須	N
ファックス番号	問合せ先のファックス番号	文字列	任意	N
電子メールアドレス	問合せ先の電子メールアドレス	文字列	任意	N
問合せ条件	問合せに対応可能な時間、曜日	文字列	任意	N

8-2. アプリケーションプログラムで利用できる関数等の技術的な公開・管理仕様

アプリケーションプログラムで利用できる関数等の技術的な公開・管理仕様として、以下の文書を定義する。

- (1) アプリケーションプログラムを開発する際に引用したガイドライン
- (2) 関数仕様
 - 関数一覧
 - 各関数の説明文書
- (3) 実装仕様
 - 実装仕様文書形式
 - WSDL による記述文書
- (4) 取得可能なデータ

【解説】

開発するアプリケーションプログラムで使用するインタフェースには、使用する関数の種類に応じて次のような3つの場合分けがある。

- 1) 本ガイドラインの関数仕様には示されている関数のみを使用する場合
- 2) 本ガイドラインの関数仕様には示されている関数以外の関数を使用する場合（使用する関数すべてが、本ガイドラインの関数仕様には示されている関数に該当しない場合）
- 3) 本ガイドラインの関数仕様には示されている関数に加え、それ以外の関数も使用する場合（使用する関数の一部が、本ガイドラインの関数仕様には示されている関数に該当しない場合）

各場面において作成する関数等の技術的な公開・管理仕様を以下に示す。

- 1) 本ガイドラインの関数仕様には示されている関数のみを使用する場合

開発するアプリケーションプログラムで使用するインタフェースが、本ガイドラインの関数仕様には示されている関数のみの場合、「関数等の技術的な公開・管理仕様」として以下の4文書を作成しなければならない。

1. アプリケーションプログラムを開発する際に引用したガイドライン記載項目を、表 8-5 に示す。
2. 使用した関数の一覧記載項目を、表 8-6 に示す。
3. WSDL による記述文書

アプリケーションプログラム使用する関数についての WSDL 文書を作成する。WSDL 文書は、当該アプリケーションプログラムに対し、1つのファイルとする。

4. 取得可能なデータの一覧

記載項目を、表 8-7 に示す。

「関数等の技術的な公開・管理仕様」の記載例を表 8-8 に示す。

表 8-5 記載項目一覧(1)

項目	定義	データ型	要求度/条件	最大記述回数
ガイドライン	アプリケーションプログラムを開発する際に引用した「標準インタフェースガイドライン」	クラス	必須	1
名称	ガイドラインの名称	文字列	必須	1
版	ガイドラインの版	文字列	必須	1
年月	ガイドラインの版の発行年月	文字列	必須	1

表 8-6 記載項目一覧(2)

項目	定義	データ型	要求度/条件	最大記述回数
関数	アプリケーションプログラムを開発する際に引用した「標準インタフェースガイドライン」の関数仕様に記載されている関数で、当該アプリケーションプログラムで利用できる関数	クラス	必須	N
機能分類	「標準インタフェースガイドライン」の関数仕様に記載されている当該関数の機能分類	文字列	任意	1
機能	「標準インタフェースガイドライン」の関数仕様に記載されている当該関数の機能	文字列	必須	1
内容	「標準インタフェースガイドライン」の関数仕様に記載されている当該関数の内容	文字列	必須	1
関数名	「標準インタフェースガイドライン」の関数仕様に記載されている当該関数の関数名	文字列	必須	1

表 8-7 記載項目一覧 (3)

項目	定義	データ型	要求度/条件	最大記述回数
データ	開発したアプリケーションプログラムで、「標準インタフェースガイドライン」に基づく関数により取得可能なデータ	クラス	必須	N
名称	データの名称	文字列	必須	1
説明	データの定義、取得基準等の説明	文字列	必須	1
内部名称	当該データの、データベース上の名称	文字列	必須	1
属性	各データ項目に対応する属性	クラス	必須	N
名称	属性の名称	文字列	必須	1
内部名称	当該属性の、データベース上の名称	文字列	必須	1

表 8-8 公開・管理仕様作成例

準拠したガイドラインの名称	標準インタフェースガイドライン
準拠したガイドラインの版	第 1.0 版
準拠したガイドラインの年月日	平成 19 年 3 月 XX 日

機能分類	機能	内容	関数名
初期設定	サービスメタデータ取得	利用可能な操作の情報を取得する	GetCapabilities
...

データ	説明	内部名称	属性 1	属性 1 内部名称	属性 2	属性 2 内部名称
メタデータ ID	カタログサービスを利用して得られるメタデータレコードの ID	recordId	ID	metadataId	名称	name
メタデータスキーマ	カタログサービスを利用して得られるメタデータのスキーマ	schema	スキーマ 名称	recordSchema	—	—
メタデータ要素	カタログサービスを利用して得られるメタデータが持つ要素の内容	elementSet	要素 名称	name	要素の 値	value

2) 開発するアプリケーションプログラムで使用するインタフェースが、本ガイドラインの関数仕様に示されていない場合

開発するアプリケーションプログラムで使用するインタフェースが、本ガイドラインの関数仕様に示されていない場合、「関数等の技術的な公開・管理仕様」として以下の 6 文書を作成しなければならない。

1. アプリケーションプログラムを開発する際に引用したガイドライン
記載項目を、表 8-5 に示す。
2. 関数一覧
本ガイドラインの「標準インタフェース関数仕様作成に関する取り決め」(5 章)に従い、関数一覧を作成する。
3. 各関数の説明文書
本ガイドラインの「標準インタフェース関数仕様作成に関する取り決め」(5 章)に従い、各関数の説明文書を作成する。
4. 実装仕様の概要
本ガイドラインの「標準インタフェース実装仕様作成に関する取り決め」(6 章)に従い、各関数について実装仕様の概要を作成する。
5. WSDL による記述文書
アプリケーションプログラム使用する関数についての WSDL 文書を作成する。WSDL 文書は、当該アプリケーションプログラムに対し、1 つのファイルとする。
6. 取得可能なデータの一覧
記載項目を、表 8-7 に示す。

なお、開発するアプリケーションプログラムで使用するインタフェースが、本ガイドラインの関数仕様に示されているかどうかの判断には注意を要する。

- 3) 開発するアプリケーションプログラムで使用するインタフェースが、本ガイドラインの関数仕様に示されている関数と、それ以外の関数の両方存在する場合

開発するアプリケーションプログラムで使用するインタフェースが、本ガイドラインの関数仕様に示されている関数と、それ以外の関数の両方存在する場合、以下を作成する。

1. アプリケーションプログラムを開発する際に引用したガイドライン
記載項目を、表 8-5 に示す。
2. 関数一覧
本ガイドラインの関数仕様に示されている関数と、それ以外の関数とを
区別して、作成する。
本ガイドラインの関数仕様に示されている関数については、表 8-6 に
示す項目とする。
本ガイドラインの関数仕様に示されていない関数については、本ガイド
ラインの「標準インタフェース関数仕様作成に関する取り決め」(5章)に
従い、関数一覧を作成する。
3. 各関数の説明文書
本ガイドラインの関数仕様に示されていない関数についてのみ、本ガイ
ドラインの「標準インタフェース関数仕様作成に関する取り決め」(5章)
に従い、各関数の説明文書を作成する。
本ガイドラインの関数仕様に示されている関数については作成しない。
4. 実装仕様の概要
本ガイドラインの関数仕様に示されていない関数についてのみ、本ガイ
ドラインの「標準インタフェース実装仕様作成に関する取り決め」(6章)
に従い、各関数について実装仕様の概要を作成する。
本ガイドラインの関数仕様に示されている関数については作成しない。
5. WSDL による記述文書
本ガイドラインの関数仕様に示されている関数と、それ以外の関数とを
まとめて、アプリケーションプログラム使用する関数として WSDL 文書
を作成する。WSDL 文書は、当該アプリケーションプログラムに対し、1
つのファイルとする。
6. 取得可能なデータの一覧
記載項目を、表 8-7 に示す。