

Tmax ゲートウェイガイド (Webサービス)

Tmax v6.0



Copyright © 2016 TmaxSoft Co., Ltd. All Rights Reserved.

Copyright Notice

Copyright © 2016 TmaxSoft Co., Ltd. All Rights Reserved.

45, Jeongjail-ro, Bundang-gu, Seongnam-si, Gyeonggi-do, 13613, South Korea

Restricted Rights Legend

All TmaxSoft Software (Tmax®) and documents are protected by copyright laws and international convention. TmaxSoft software and documents are made available under the terms of the TmaxSoft License Agreement and this document may only be distributed or copied in accordance with the terms of this agreement. No part of this document may be transmitted, copied, deployed, or reproduced in any form or by any means, electronic, mechanical, or optical, without the prior written consent of TmaxSoft Co., Ltd.

Nothing in this software document and agreement constitutes a transfer of intellectual property rights regardless of whether or not such rights are registered) or any rights to TmaxSoft trademarks, logos, or any other brand features. This document is for information purposes only. The company assumes no direct or indirect responsibilities for the contents of this document, and does not guarantee that the information contained in this document satisfies certain legal or commercial conditions. The information contained in this document is subject to change without prior notice due to product upgrades or updates. The company assumes no liability for any errors in this document.

このソフトウェア(Tmax®)マニュアルの内容とプログラムは、日本国の著作権法および国際条約によって保護されています。マニュアルの内容とプログラムは、TmaxSoft Co., Ltd.との使用許諾契約書の下でのみ使用することができ、マニュアルは使用許諾契約で許可されている範囲を除いては、配布または複製することができません。TmaxSoftの書面による事前の承諾を得ることなく、このマニュアルの全部または一部を電子的または機械的な方法を問わず、転送、複製、配布したり、または二次的著作物を作成する等の行為を一切禁じます。

このソフトウェアのマニュアルとプログラムの使用許諾契約は、いかなる場合においても、マニュアル及びプログラムと関連する知的財産権(登録の有無を問わず)を譲渡するものと解釈されず、TmaxSoftのブランド、ロゴ、商標等の使用権限を与えるものではありません。マニュアルは、情報を提供する目的でのみ提供しており、これに伴う契約上の直接的ないしは間接的な責任を負わず、マニュアルの内容は法律上もしくは商業的な特定の条件が満たされることを保証しません。マニュアルの内容は、製品のアップグレード及び修正により、その内容が予告なく変更されることがあり、内容上の誤りがないことを保証しません。

Trademarks

Tmax®, Tmax WebtoB® and JEUS® are registered trademark of TmaxSoft Co., Ltd. Other products, titles or services may be registered trademarks of their respective companies.

Tmax®, Tmax WebtoB®, JEUS® は、TmaxSoft Co., Ltd.の登録商標です。その他、記載されている会社名、製品名などは、各社の商標または登録商標です。

Open Source Software Notice

Some modules or files of this product are subject to the terms of the following licenses. : openssl-0.9.7.m, zlib-1.1.4, expat-2.0.0, net-snmp, DCE1.0, pthread, google-diff-match-patch, libevent, getopt

Detailed Information related to the license can be found in the following directory :
\${INSTALL_PATH}/license/oss_licenses

この製品の一部ファイルまたはモジュールは、openssl-0.9.7.m、zlib-1.1.4、expat-2.0.0、net-snmp, DCE1.0、pthread、google-diff-match-patch、libevent、getoptライセンスを遵守します。

詳細情報については、製品ディレクトリーの\${INSTALL_PATH}/license/oss_licensesに記載されている事項を参照してください。

文書情報

文書名: Tmax ゲートウェイガイド (Webサービス)

発行日: 2016年8月5日

ソフトウェアバージョン: Tmax v6.0

ガイドバージョン: v2.1.1

目次

このガイドについて	ix
第1章 紹介	1
1.1. 概要	1
1.1.1. Webサービス	1
1.2. Webサービス・ゲートウェイ	2
1.2.1. Webサービス・サプライヤー機能	3
1.2.2. Webサービス・リクエスター機能	4
第2章 環境設定	5
2.1. サービス情報ファイル	5
2.1.1. バイナリの作成	8
2.1.2. 例	9
2.2. Webサービス・ゲートウェイ設定ファイル	11
2.2.1. ファイル形式	11
2.2.2. 例	13
2.3. Tmax環境設定	13
第3章 ライブラリー	15
3.1. コールバック関数	15
3.1.1. *request_from_tmax	15
3.1.2. *reply_from_tmax	16
3.1.3. *request_from_remote	16
3.1.4. *reply_from_remote	17
3.2. データ変換ライブラリー	18
3.2.1. init_config	18
3.2.2. convert_tmax_to_soap	18
3.2.3. convert_soap_to_tmax	19
第4章 管理	21
4.1. コマンド	21
4.1.1. wsgwinfo	21
4.1.2. wsgwreload	21
4.2. ユーティリティー	22
4.2.1. tmmbfgen	22
4.2.2. untmmbfgen	23
4.2.3. xwsdlgen	24
第5章 例	25
5.1. サービスの作成	25
5.1.1. サーバー・プログラム	25
5.2. Webサービスの設定	27
5.2.1. サービス情報ファイルの作成	27
5.2.2. Webサービス・ゲートウェイ設定ファイルの作成	29

5.3. Tmax環境ファイル	29
5.4. Axis1.4クライアント・プログラムの作成	30
5.4.1. WSDLファイルの作成	30
5.4.2. Stubの作成	30
付録 A. 内部構造体型のサポート	35
索引	41

図目次

[図 1.1] Webサービス・サプライヤーのWSGWの動作構造	3
[図 1.2] Webサービス・リクエスターのWSGWの動作構造	4

このガイドについて

対象読者

本書は、Tmax[®](以下、Tmax)が提供するゲートウェイ・サーバーを使用する開発者を対象としています。本書は、Tmaxサービスを変更せずにWebサービスで使用するために提供されるゲートウェイ・サーバーのWebサービス・ゲートウェイについて説明します。

前提知識

本書は、Tmaxシステムの概要とTmaxシステムが提供する各種機能や特性などを習得するための手引書です。

本書を理解するためには、以下の事項についての知識が必要です。

- ミドルウェアおよびUNIXシステムについての知識
- Tmaxの基本概念
- Java、Cプログラミングについての知識
- Webサービスについての基本知識

制限事項

本書を読む前にTmaxの基本概念を熟知している必要があります。実務上の具体的な使用方法や管理・運用についての内容は、各製品ガイドを参照してください。

参考

Tmaxシステムの開発についての基本的な内容は、『Tmax 運用ガイド』および『Tmax アプリケーション開発ガイド』を、Tmaxが提供するコマンドとC APIについては、『Tmax リファレンスガイド』を参照してください。

本書の構成

本書は、計5章と付録で構成されています。

各章の主な内容は以下のとおりです。

- 第1章: 紹介

Webサービス・ゲートウェイについて紹介し、動作構造について説明します。

- 第2章: 環境設定

Webサービス・ゲートウェイを動作させるために必要なサービス情報ファイル、Webサービス・ゲートウェイ設定ファイル、Tmax環境設定ファイルについて説明します。

- 第3章: ライブラリー

Webサービス・リクエスター機能のため、デフォルトで提供されるコールバック関数とデータ変換ライブラリーについて説明します。

- 第4章: 管理

Webサービス・ゲートウェイのステータス情報の閲覧および再起動するコマンドとユーティリティーについて説明します。

- 第5章: 例

TmaxサービスのECHOSTRING、ECHOSDL、ECHOFDL(インプットとアウトプットが同様なサービス)をWebサービスとして提供する方法について説明します。

- 付録A: 内部構造体型のサポート

内部サポート可能な構造体型の例について説明します。

表記上の規則

表記	意味
<AaBbCc123>	プログラム・ソースコードのファイル名
<Ctrl>+C	CtrlとCを同時に押す
[Button]	GUIのボタン、メニュー名
太字	強調
「」、『』（鍵カッコ）	関連文書、あるいはガイド内の他の章および節の表示
「入力項目」	画面UI上の入力項目
ハイパーリンク	メール・アカウント、Webサイト
>	メニューの実行順
+----	下位ディレクトリー/ファイル有り
----	下位ディレクトリー/ファイル無し
参考	参照/注意事項
[図1.1]	図の名称
[表1.1]	表の名称
AaBbCc123	コマンド、コマンド実行後の画面に出力された結果物、サンプル・コード
[]	オプション・パラメータ値
	選択パラメータ値

システム要件

	要求事項
プラットフォーム	IBM AIX 5.x / 6.1 / 7.1
	HP-UX 11.xx
	SunOS 5.7~5.9 / SunOS 5.10 / SunOS 5.11
ハードウェア	1GB以上のハードディスク空き容量
	512MB以上のメモリー空き容量
データベース	Oracle 9~12
	Tibero 4~5
	DB2
	Informix

関連文書

ガイド	説明
Tmax 運用ガイド	Tmaxを利用するための環境設定ファイルとシステム運用方法について説明しています
Tmax アプリケーション開発ガイド	Tmaxアプリケーション・プログラムの開発で使用するAPIの概念と使用方法および例について説明しています
Tmax リファレンスガイド	Tmaxアプリケーションの開発に使用するコマンドおよびクライアントとサーバーの接続、通信に使用する関数の使用方法と例について説明しています

お問合せ先

Korea

TmaxSoft Co., Ltd.
45, Jeongjail-ro, Bundang-gu,
Seongnam-si, Gyeonggi-do, 13613
South Korea
Tel: +82-31-8018-1000
Fax: +82-31-8018-1115
Email: info@tmax.co.kr
Web (Korean): <http://www.tmaxsoft.com>
TechNet: <http://technet.tmaxsoft.com>

USA

TmaxSoft Inc.
101 North Wacker Drive, Suite 2014,
Chicago, IL 60606
U.S.A
Tel: +1-312-525-8330
Email: info@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/us_en/home

Japan

TmaxSoft Japan Co., Ltd.
5F Sanko Bldg, 3-12-16 Mita, Minato-Ku, Tokyo, 108-0073
Japan
Tel: +81-3-5765-2550
Fax: +81-3-5765-2567
Email: info@tmaxsoft.co.jp
Web (Japanese): <http://www.tmaxsoft.co.jp>

China

Beijing TmaxSoft System Software Co., Ltd.
Room103, No.2 Huizhong Building, Seven Street Shangdi,
Haidian District, Beijing, 100085
P.R.China
Tel: +86-10-6298-8827
Email: info@tmaxsoft.com.cn
Web (Chinese): http://www.tmaxsoft.com/cn_en/home_cn_en

Brazil

Tmax Brasil Sistemas e Serviços Ltda.
Av. Copacabana, 177, sala 32~35 Empresarial 18 do Fortel
Alphaville Barueri, Sao Paulo, 06472-001
Brazil
Tel: +55-11-4191-3100
Fax: +55(11) 4191-3705 (extension#112)
Email: info.bra@tmaxsoft.com
Web (Portuguese): http://www.tmaxsoft.com/br_en/home_br_en

Russia

Tmax Rus L.L.C.
Leninsky prospekt, 113/1 (Park Place Moscow),
Office 318e, Moscow, 117198
Russia
Tel: +7(495)970-01-35
Email: info.rus@tmaxsoft.com
Web (Russian): http://www.tmaxsoft.com/ru_ru/home_ru_ru

Singapore

Tmax Singapore Pte. Ltd.
430 Lorong 6, Toa Payoh #10-02,
OrangeTee Building, 319402
Singapore
Tel: +65-6259-7223
Fax: +65-6258-7112
Email: info.sg@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/sg_en/home_sg_en

United Kingdom

TmaxSoft UK Ltd.
215 Knyvett House, Watermans Business Park,
The Causeway, Staines TW18 3BAB
United Kingdom
Tel: +44-1784-895005
Email: info.uk@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/gb_en/home_gb_en

Canada

TmaxSoft Canada, Inc.
2425 Matheson Blvd East, 8th floor,
Unit 824 Mississauga, ON, L4W 5K4
Canada
Tel: +1-905-361-2888
Email: info.canada@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/ca_en/home_ca_en

Australia

TmaxSoft Proprietary Limited
L32, 101 Miller Street, North Sydney 2060
Australia
Tel: +91-9845-330-704
Email: info.aus@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/au_en/home_au_en

India

TmaxSoft Technologies Private Limited
Sobha Alexander Plaza, 3rd Floor,
16/2 Commissariat Road, Bangalore-560025
India
Tel: +91-9845-330-704
Email: info.india@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/in_en/home_in_en

Turkey

TmaxSoft Co., Ltd. Turkey Liaison Office
Windowist Tower. Eski Buyukdere Cad. No:26,
Maslak 34467 Istanbul
Turkey
Tel: +90-544-553-6045
Email: cslee@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/tr_en/home_tr_en

第1章 紹介

本章では、Webサービス・ゲートウェイについて紹介し、動作構造について説明します。

1.1. 概要

Webサービスとは、アプリケーションがプラットフォームおよびプログラミング言語とは独立された方式で双方向通信ができるようにする、標準化された技術です。なお、標準XMLメッセージングを利用して、ネットワークにアクセス可能なオペレーションを記述するソフトウェア・インターフェースです。Webサービスはインターネットにさえ接続されていれば、サービスの権限を有するユーザーに対してビジネスを公開し、使用できるようにするため、メッセージング・プロトコル、プログラミング標準、サービスが検索できるような便宜環境を定義しています。

1.1.1. Webサービス

Webサービスは、インターネットのURI(Uniform Resource Identifier)でアクセス可能なアプリケーションです。Webサービスのインターフェースとバインディングは、XML文で定義・記述・検索されることができます。1つのWebサービスは、インターネット上に公開されている別のWebサービスとの相互連動が可能で、かつ既存のバックエンド(back-end)アプリケーションとの連動も可能です。

今までのアプリケーション・アーキテクチャーには、2つの種類がありました。1つは、メインフレームを基盤として動作する単一化されたシステムであり、もう1つは、デスクトップで動作するクライアント・サーバー基盤のシステムです。

システムの動作には両方とも問題ありませんが、このようなアーキテクチャー上で動作するプログラムは、そのシステムでのみ動作できるように設計されたプログラムなので制約があります。アーキテクチャー内でのプログラムは非常に閉鎖的であり、アクセスが容易ではないため、Web上に存在する無数のユーザーには無用の長物です。そのため、ソフトウェア・ビジネスは、サービス指向アーキテクチャー(SOA:Service-Oriented Architecture)の方向で進化し、この基盤上でのアプリケーションは、Web上で動的に相互作用できるようになりました。

アプリケーションは、大規模で構成されているソフトウェア・システムを、小規模でモジュール化された複数のサブシステムとして再構成しています。このように小さくモジュール化されたサブソフトウェア・システムは、それぞれ多様な技術で実現できるため、1つのコンピューターに存在する必要はなく、再使用も可能になりました。なお、XMLとHTTPのような標準Webプロトコルを使用しており、インターネット上のいかなるユーザーも容易にアクセスできるようにしました。

このようなサービス指向技術は、すでに数年前からRMI、COM、CORBAのような様々な技術で実現されており、真新しい技術ではありません。しかし、これらの技術はベンダーに從属されているか、實現された技術に從属的というデメリットがあります。Webサービスは、このようなデメリットを克服しています。

Webサービスの特徴

Webサービスには下記のような特徴があります。

- Webサービスは、Webを通じてアクセスできます。
- Webサービスは自ら標榜し、記述します。
 - Webサービスは自らの役割、機能、属性について記述しており、サービスに対するWebサービス・クライアントの理解をサポートします。WSDL(Web Service Description Language)というファイルにサービスを記述し、これを公開することで、別のアプリケーションでサービスが利用できるようにします。
 - リカバリー/ロールバックによる安定性を保証します。
- Webサービスは、HTTPのような標準インターネット・プロトコルによって渡されるXMLメッセージを通じて、Webサービス・クライアントと通信します。Webサービス・クライアントはアプリケーションであったり、別のWebサービスであったりします。
- Webサービスは、要求/応答あるいはone-way方式で動作します。同期または非同期通信で呼び出されることもあります。なお、動作方式や通信方式を問わず、WebサービスとWebサービス・クライアント間でやり取りされる基本的な単位はメッセージです。

Webサービスの長所

Webサービスには下記のような長所があります。

- インターネット公開標準をサポートします。

Webサービスは、SOAP(Simple Object Access Protocol)、WSDL、UDDI(Universal Description, Discovery, and Integration)のような公開標準を定め、これをベースにして相互作用が行われます。そのため、Webサービスをサポートするアプリケーション・プログラムは問題なく相互作用されます。
- プラットフォームおよび言語に対して独立的です。
- HTTPのようなWebプロトコルを使用しており、ファイアウォールのような障害にも影響を受けないため、アプリケーションに対するアクセスが容易です。

1.2. Webサービス・ゲートウェイ

Webサービス・ゲートウェイ(WebService Gateway、以下、WSGW)は、Webサービス・サプライヤー(Supplier)がTmaxのサービスを呼び出すか、Tmaxで外部のWebサービスを呼び出すために提供されるゲートウェイ・サーバーです。

WSGWはサプライヤーとリクエスターの役割をします。Webサービスのサプライヤーとして機能を実行する場合、Tmaxサービスを変更せずにWebサービスで使用することができます。HTTPプロトコルを基盤とするSOAP要求メッセージを処理する、Webサービス・サプライヤーの役割をします。

WSGWを実行するには、Tmax環境ファイルのGATEWAYセクションに登録します。付加的にWebサービス・ゲートウェイ設定ファイル、サービス情報ファイルが必要です。

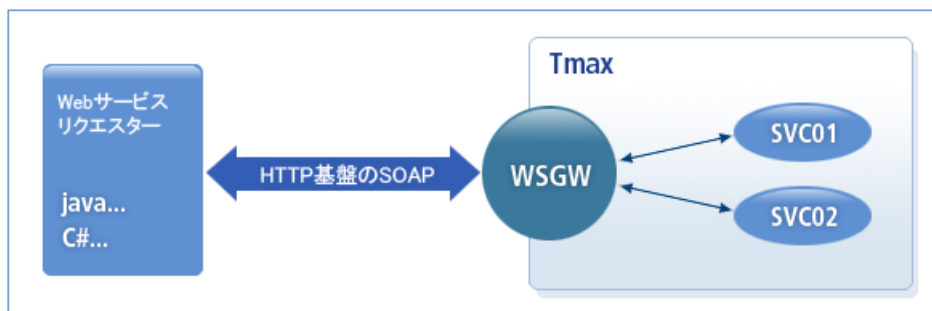
1.2.1. Webサービス・サプライヤー機能

Webサービス・リクエスターが送信したSOAP要求メッセージが受信されたら、Webサービス・ゲートウェイはSOAPメッセージを分析し、該当するTmaxサービスを呼び出します。その後、結果をSOAPメッセージとして作成してリクエスターに転送します。

Webサービス・サプライヤー機能のWSGWが動作するために使用するユーティリティには、tmmbfgen、untmmbfgen、xwsdlgenがあります。サービスの動作のため、Tmax環境設定ファイル、Webサービス・ゲートウェイ設定ファイル、サービス情報ファイル(サービス情報バイナリ・ファイル)、WSDLファイルを設定します。

下記は、Webサービス・サプライヤー機能をするWSGWの動作構造です。

[図 1.1] Webサービス・サプライヤーのWSGWの動作構造



1. サービス情報ファイルを作成します。
2. tmmbfgenでサービス情報バイナリ・ファイルを作成します。
3. Webサービス・ゲートウェイ設定ファイルを作成します。
4. Tmax環境設定ファイルにWebサービス・ゲートウェイを設定します。
5. Tmaxを再起動するか、tmadminでwsgwreloadコマンドを入力して、Webサービス・ゲートウェイを実行させます。
6. xwsdlgenを使ってwsdl文を作成し、サービス・リクエスターに提供します。

7. サービス・リクエスターは、wsdl文を利用してクライアント・アプリケーションを作成します。

1.2.2. Webサービス・リクエスター機能

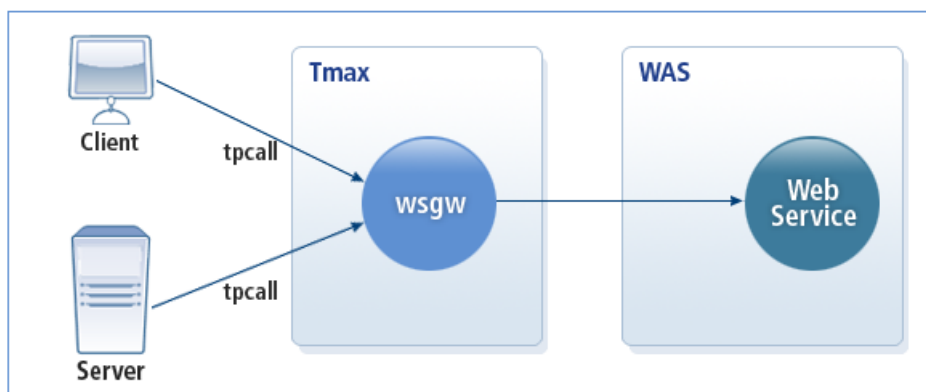
Tmaxで外部Webサービスを提供するサプライヤーにサービスを要求するための機能です。

Tmaxクライアントまたはサーバーで、Webサービス・ゲートウェイに割り当てたサービスを呼び出す際、要求メッセージはWebサービス・ゲートウェイを通じて、サプライヤー(WASと同じ)にSOAPメッセージを渡します。WSGWは要求が発生するたびにサプライヤーと接続し、メッセージを受信するとき要求を終了します。

Webサービス・リクエスター機能のWSGWが動作するために使用するユーティリティには、tmmbfgen、untmmbfgen、xwsdlgenがあります。サービスの動作のため、Tmax環境設定ファイル、Webサービス・ゲートウェイ設定ファイル、サービス情報ファイル(サービス情報バイナリ・ファイル)を設定します。

下記は、リクエスター機能をするWSGWの動作構造です。

[図 1.2] Webサービス・リクエスターのWSGWの動作構造



1. サービス情報ファイルを作成します。
2. tmmbfgenを使ってサービス情報バイナリ・ファイルを作成します。
3. Webサービス・ゲートウェイ設定ファイルを作成します。
4. Tmax環境設定ファイルにWebサービス・ゲートウェイを設定します。
5. Tmaxを再起動するか、tmadminでwsgwreloadコマンドを入力して、Webサービス・ゲートウェイを実行させます。
6. TmaxクライアントとサーバーでWebサービスを呼び出します。

第2章 環境設定

Webサービス・ゲートウェイ(WSGW)を動作させるには、サービス情報ファイル、WSGW設定ファイル、Tmax環境設定ファイルを設定する必要があります。本章では、各環境ファイルの設定方法について説明します。

2.1. サービス情報ファイル

TmaxのサービスをWebサービスで利用するには、外部へ提供するサービスの説明が必要となります。定義されたファイルは、WSGWから受信されるSOAPメッセージをTmaxバッファに変換し、サービス結果をSOAPメッセージとして変換するか、サービス・リクエスターのためにWSDL文を作成する場合に使用します。

サービス定義は、サービス情報(名前、インプット・パラメータ・タイプ、アウトプット・パラメータ・タイプ)、インプット・パラメータ情報、アウトプット・パラメータ情報で構成します。サービスは、SERVICE、REQUEST、RESPONSEセクションを使用して表現します。SERVICEセクションの下位内容は、サービス情報、REQUESTセクションの下位内容は、インプット・パラメータ情報、RESPONSEセクションの下位内容は、アウトプット・パラメータ情報を表します。

要求/応答の動作方式を使用したい場合は、REQUEST、RESPONSEセクションを両方とも定義します。一方、one-way方式を使用する場合は、REQUESTセクションのみ定義します。1つのファイルに複数のサービスが定義可能であり、大小文字を区別する必要があるため、正確に作成します。REQUESTまたはRESPONSEセクションに、フィールドが内部構造体をサポートする場合は、STRUCTセクションに該当するデータ型を記録して利用することができます。各セクションの設定項目に関する詳細内容は、『Tmax運用ガイド』を参照してください。

サービス情報ファイルがそのまま使用されず、**tmmbfgen**ユーティリティに変換されたファイルをWSGWと**xwsdlgen**で使します。

SERVICEセクション

下記は、SERVICEセクションの項目に関する説明です。

- 必須項目

項目	説明
name	外部から提供されるオペレーション名です

- 選択項目

項目	説明
tmaxname	実際にTmaxが提供するサービス名です。この項目が定義されていないと、name項目がtmaxnameと同じであると判断して処理します
intype	インプット・パラメータのタイプ(String、CARRAY、STRUCT、FDL、char、short、int、float、double)、インプット・パラメータが存在しない場合は定義する必要はありません
outtype	アウトプット・パラメータのタイプ(String、CARRAY、STRUCT、FDL、char、short、int、float、double)、アウトプット・パラメータが存在しない場合は定義する必要はありません
instruct	インプット・パラメータ・タイプがSTRUCTかFDLの場合、構造体の名前またはユーザー定義名です <ul style="list-style-type: none"> – STRUCTの場合：SDLに定義された構造体の名前を設定します – FDLの場合：ユーザーが任意で決めます。値を指定しない場合は、自動的に「サービス名RequestFDL」で値を保存します
outstruct	アウトプット・パラメータ・タイプがSTRUCTかFDLの場合、構造体の名前またはユーザー定義名です <ul style="list-style-type: none"> – STRUCTの場合：SDLに定義された構造体の名前を設定します – FDLの場合：ユーザーが任意で決めます。値を指定しない場合は、自動的に「サービス名ResponseFDL」で値を保存します
inalias	Tmaxで使用しているinstructの名前と、Webサービスで使用しているREQUEST名が異なる場合、Webサービスで使用する名前を定義します
outalias	Tmaxで使用しているoutstructの名前と、Webサービスで使用しているRESPONSE名が異なる場合、Webサービスで使用する名前を定義します

REQUEST、RESPONSEセクション

下記は、REQUEST、RESPONSEセクションの項目に関する説明です。

- 必須項目

項目	説明
name	パラメータ名です
type	パラメータのタイプ(char、int、long、float、double、stirng、carray、struct)です

- 選択項目

項目	説明
maxoccurs	該当する全タイプ(intype or outtpye)がSTRUCTかFDLの場合、このフィールドが入力できる最大数を制限することができます
minoccurs	該当する全タイプ(intype or outtpye)がSTRUCTかFDLの場合、このフィールドが入力できる最小数を制限することができます
restriction	wsdlに変更される場合、restriction項目と同様です – length、minLength、maxLength、enumeration、pattern、minExclusive、maxExclusive、minInclusive、maxInclusive、totalDigit、fractionDigit (例: length="3", patter="ab?")
subtype	タイプがSTRUCTの場合、構造体名を定義します

STRUCTセクション

下記は、STRUCTセクションの項目に関する説明です。

- 必須項目

項目	説明
name	構造体名です
TYPE	TYPEセクションに属しているフィールドを定義します

- TYPEセクションの項目

下記は、STRUCT内のTYPEセクションの項目に関する説明です。

- 必須項目

項目	説明
name	パラメータ名です
type	パラメータのタイプ(char、int、long、float、double、stirng、carray、struct)です

- 選択項目

項目	説明
maxoccurs	該当する全タイプ(intype or outtpye)がSTRUCTかFDLの場合、このフィールドが入力できる最大数を制限することができます

項目	説明
minoccurs	該当する全タイプ(intype or outtpye)がSTRUCTかFDLの場合、このフィールドが 入力できる最小数を制限することができます
restriction	wsdlに変更される場合、restriction項目と同様です – length、minLength、maxLength、enumeration、pattern、minExclusive、 maxExclusive、minInclusive、maxInclusive、totalDigit、fractionDigit (例: length="3", patter="ab?")
subtype	タイプがSTRUCTの場合、構造体名を定義します

2.1.1. バイナリの作成

tmmbfgen

ユーザーが作成したサービス情報ファイルは、WSGWとxwsdlgenでそのまま使用することができません。そのため、tmmbfgenを使って新しいファイルを作成する必要があります。tmmbfgenによって作成されたファイルをサービス情報バイナリ・ファイルといいます。tmmbfgenを使用すると、文法とパラメータのタイプをチェックするため、WSGWで参照する前に有効性をチェックすることが可能です。さらに、テキストを複数に分割して管理することができます。

tmmbfgenを下記のような3つの方法で 사용할 ことができます。

- 新しいサービス情報バイナリ・ファイルの作成

下記は、sampleを作成し、sample1.txtとsample2.txtに定義されているサービス情報をsampleに入力する例です。

```
tmmbfgen -r sample1.txt,sample2.txt -o sample
```

- 既存のサービス情報バイナリ・ファイルに追加

下記は、既存のsampleファイルに、sample3.txtに定義されているサービス情報を追加する例です。同じサービスが存在する場合は上書きします。

```
tmmbfgen -i sample3.txt -o sample
```

- 既存のサービス情報バイナリから特定のサービスを削除

下記は、sampleファイルからサービスSVC1、SVC2の情報を削除する例です。

```
tmmbfgen -d SVC1,SVC2 -o sample
```

untmmbfgen

ユーザーがサービス情報バイナリ・ファイルを確認できるように逆変換します。

```
untmmbfgen -i sample -o unsample.txt
```

2.1.2. 例

メッセージのタイプによってサービス情報ファイルを設定する方法について説明します。

文字列(String)型

文字列を取得し、大文字に変換して返すサービスです。

- インプット・パラメータ : char[20]
- アウトプット・パラメータ : char[20]

下記は、TOUPPERサービス情報ファイルの例です。

```
SERVICE
name = TOUPPER
intype = STRING
outtype = STRING

REQUEST
name = req
type = string

RESPONSE
name = result
type = string
```

構造体(STRUCT)型

文字列の長さと言文字列を有する構造体を取得し、大文字に変換して返すサービスです。

- インプット・パラメータ : kstrdata
- アウトプット・パラメータ : kstrdata

下記は、TOUPPERSDLサービス情報ファイルの例です。

```
SERVICE
name = TOUPPERSDL
intype = STRUCT
outtype = STRUCT
instruct = kstrdata
outstruct = kstrdata
```

```
REQUEST
name = len
type = int
name = str
type = string
```

```
RESPONSE
name = len
type = int
name = str
type = string
```

下記は、SDLファイルの例です。

```
struct kstrdata {
    int    len;
    char    sdata[20];
}
```

フィールド(FILED)型

文字列フィールドを取得し、大文字に変換して返すサービスです。

- インプット・パラメータ：フィールドINPUT
- アウトプット・パラメータ：フィールドOUTPUT

下記は、TOUPPERFDLサービス情報ファイルの例です。

```
SERVICE
name = TOUPPERFDL
intype = FDL
outtype = FDL
instruct = toupfdl
outstruct = toupfdl
```

```
REQUEST
name = INPUT
type = string
```

```
RESPONSE
name = OUTPUT
type = string
```

下記は、FDLファイルの例です。

```
INPUT    101  string  -  -
OUTPUT   102  string  -  -
```

2.2. Webサービス・ゲートウェイ設定ファイル

Webサービス・ゲートウェイ(WSGW)が開始するときにWSGW設定ファイルを読み取り、どのようなTmaxサービスを提供するか、HTTPを待ち受け(listen)するIPとポート番号を決めます。WSGW設定ファイルには、提供するサービス一覧とバインディングするHTTPアドレスが含まれており、xwsdlgenでも使用します。

2.2.1. ファイル形式

WSGW設定ファイルは、<Configuration>をルート要素(root element)として有するXMLで構成します。サブ要素(sub element)がWebサービス・サプライヤーの役割をする場合は、<in>の下位に設定し、Webサービス・リクエスター機能を提供する場合は、<out>に設定を定義します。

<in>

<in>のサブ要素は、提供するサービス一覧を説明する<Servicelist>と、バインディングするアドレスを説明する<GWInstance>で表現します。

- <Servicelist>

提供するサービス一覧をサブ要素の<service>を利用して表現します。

```
<Servicelist id="sample">
  <service name="TOUPPER" />
  <service name="TOUPPERSDL" />
</Servicelist>
```

属性	説明
id	サービス一覧の名前です。WSDL文を作成するとき、名前空間で指定します

- <service>

提供する1つのサービス名です。

属性	説明
name	サービス情報ファイルのSERVICEセクションのnameに適用する名前を指定します

属性	説明
wswns	WSDL文を作成する際、ルート要素の名前空間で指定します
typens	WSDL文を作成する際、タイプに関連する要素の名前空間で指定します

- <GWInstance>

```
<GWInstance> <HTTP address="127.0.0.1:10032"/> </GWInstance>
```

- <HTTP>

属性	説明
address	WSGWが待ち受け(listen)するHTTPアドレスとポート番号を定義します

<out>

<out>のサブ要素は、呼び出すWebサービス一覧である<Servicelist>と接続する位置を属性として有します。

- <service>

提供するサービス一覧をサブ要素の<service>を利用して表現します。

属性	説明
uri	実際のサービスを要求するWebサービス・サプライヤーのアドレスです
soap_version	サービスを要求するSOAPバージョン(1.1、1.2)です
encoding_type	rpc、doc
wsns	WSDL文の上位ターゲットの名前空間です
typens	WSDL文のデータ型の名前空間です

- <Servicelist>

呼び出すサービス一覧をサブ要素の<service>を利用して表現します。

```
<Servicelist id="sample">
  <service name="TOUPPER" />
  <service name="TOUPPERSDL" />
</Servicelist>
```

- <service>

提供する1つのサービス名です。

属性	説明
name	サービス情報ファイルのSERVICEセクションのnameに適用する名前を指定します

2.2.2. 例

下記は、TOUPPER、TOUPPERSDL、TOUPPERFDLを提供するゲートウェイ設定ファイルです。

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration xmlns="http://www.tmax.co.kr/twsp/twsgwconfig/">
  <in>
    <Servicelist id="TmaxSampleECHO" >
      <Service name="TOUPPER_TMAX" />
    </Servicelist>
    <GWInstance>
      <HTTP address="192.168.1.43:5001"/>
    </GWInstance>
  </in>
  <out uri="http://192.168.11.240:80/axis2/services/TmaxSampleTouppers"
    soap_version = "1.2" encoding_style="rpc"
    wsns = "http://192.168.11.240:80/axis2/services/TmaxSampleTouppers"
    typens= ">
    <Servicelist>
      <Service name="TOUPPER" />
      <Service name="TOUPPERFDL" />
    </Servicelist>
  </out>
  <out uri="http://192.168.11.240:80/axis2/services/RPCSample"
    soap_version = "1.1" encoding_style="rpc"
    wsns= "http://192.168.11.240:80/axis2/services/RPCSample/RPCSample"
    typens= "http://192.168.11.240:80/axis2/services/RPCSample/RPCSampleType">
    <Servicelist>
      <Service name="TOUPPERSDL" />
    </Servicelist>
  </out>
</Configuration>
```

2.3. Tmax環境設定

Tmax環境設定のGATEWAYセクションの設定に従います。必須属性は、GWTYPE、CLOPT(-c、-m)、NODENAME、DIRECTIONです。

下記は、Tmax環境設定ファイルの例です。

```

*SERVICE
TOUPPER          SVRNAME = WSGWOUT
TOUPPERSDL       SVRNAME = WSGWOUT
TOUPPERFDL       SVRNAME = WSGWOUT

*GATEWAY
WSGWOUT
    GWTYPE=WSGW,
    NODENAME=tmxi4,
    PORTNO=5005,
    RGWADDR="192.168.11.240",
    RGWPORTNO=80,
    DIRECTION=OUT,
    CLOPT="-csample.xml -msample",
    CPC=3

```

下記は、Tmax環境設定の必須設定項目についての説明です。

項目	説明
GWTYPE	必ずWSGWで設定します
CLOPT	使用するWebサービス設定ファイルと、サービス情報バイナリ・ファイルの名前を設定します。このファイルは、\$TMAXDIR/config内に存在する必要があります
NODENAME	WSGWを使用するノード名を入力します
DIRECTION	Webサービス・サプライヤーの役割をする場合はINを、Webサービスのリクエスターの役割をする場合はOUTを入力します
PORTNO	待ち受け(listen)するポート番号を入力します

第3章 ライブラリー

本章では、Webサービス・リクエスター機能のため、基本的に提供されるコールバック関数とデータ変換ライブラリーについて説明します。

3.1. コールバック関数

Webサービス・リクエスター機能のため、基本的にゲートウェイの実行ファイルを提供します。ユーザーが任意でTmaxメッセージをSOAPメッセージに変換するためには、Webサービス・ゲートウェイ(WSGW)ライブラリーと一緒に提供されるコールバック関数を実装、ビルドし、ゲートウェイ実行ファイルを作成して実行することができます。提供するライブラリーは、**libwsgw**です。ユーザーが、ゲートウェイから要求を送信したり、応答を受信したりする際に前処理したい場合は、このライブラリーとコールバック関数を実装して実行ファイルをビルドした後、wsgwを作成して処理します。

下記は、ライブラリーで提供するコールバック関数の一覧です。

関数	説明
*request_from_tmax	Webサービス・ゲートウェイから呼び出されるコールバック関数です。Tmaxから外部への要求が呼び出されます
*reply_from_tmax	外部からTmaxへの要求に対する応答が存在する場合に呼び出されます
*request_from_remote	外部からTmaxへの要求が存在する場合に呼び出されます
*reply_from_remote	Tmaxから外部への要求に対する応答が存在する場合に呼び出されます

3.1.1. *request_from_tmax

Webサービス・ゲートウェイから呼び出されるコールバック関数です。Tmaxから外部への要求が呼び出されます。

- プロトタイプ

```
int (*request_from_tmax)(char *tmaxmsg, int ilen, char **soapmsg, int *olen ,
                        char **service, char **uri, char **soap_version);
```

- パラメータ

パラメータ	説明
tmaxmsg [in]	変換されるTmaxメッセージです

パラメータ	説明
ilen [in]	Tmaxメッセージ(tmaxmsg)の長さです
soapmsg [out]	変換されたSOAPメッセージです。内部でフリーなので、必ずmallocで割り当てます
olen [out]	変換されたSOAPメッセージの長さです
service [out]	実サービス名です。内部でフリーなので、必ずmallocで割り当てます
uri [out]	uriです。内部でフリーなので、必ずmallocで割り当てます
soap_version [out]	SOAPバージョン(1.1 or 1.2)です。内部でフリーなので、必ずmallocで割り当てます

3.1.2. *reply_from_tmax

外部からTmaxへの要求に対する応答が存在する場合に呼び出されるコールバック関数です。

- プロトタイプ

```
int (*reply_from_tmax)(char *tmaxmsg, char **soapmsg, int *olen, char **service);
```

- パラメータ

パラメータ	説明
tmaxmsg [in]	変換されるTmaxメッセージです
soapmsg [out]	変換されたSOAPメッセージです
olen [out]	変換されたSOAPメッセージ(soapmsg)の長さです
service [out]	実サービス名です

3.1.3. *request_from_remote

外部からTmaxへの要求が存在する場合に呼び出されるコールバック関数です。

- プロトタイプ

```
int (*request_from_remote)(char *soapmsg, int soapmsglen, char **tmaxmsg,
                           char **service);
```

- パラメータ

パラメータ	説明
soapmsg [in]	変換されるSOAPメッセージです

パラメータ	説明
soapmsglen [in]	SOAPメッセージ(soapmsg)の長さです
tmaxmsg [out]	変換されたTmaxメッセージです
service [out]	実サービス名です

3.1.4. *reply_from_remote

Tmaxから外部への要求に対する応答が存在する場合に呼び出されるコールバック関数です。

- プロトタイプ

```
int (*reply_from_remote)(char *soapmsg, int soapmsglen, char **tmaxmsg, int *olen);
```

- パラメータ

パラメータ	説明
soapmsg [in]	変換されるSOAPメッセージです
soapmsglen [in]	SOAPメッセージ(soapmsg)の長さです
tmaxmsg [out]	変換されたTmaxメッセージです
olen [out]	Tmaxメッセージ(tmaxmsg)の長さです

例

下記は、使用するコードです。

```
#include <usrinc/wsgw_user.h>

int main(int argc, char *argv[])
{
    request_from_tmax = NULL;
    reply_from_tmax = NULL;
    request_from_remote = NULL;
    reply_from_remote = NULL;

    return wsgw_main(argc,argv);
}
```

3.2. データ変換ライブラリー

TmaxメッセージとSOAPメッセージを変換するために提供されるライブラリーの**libwsgw**です。Webサービス・ゲートウェイのルールに従って、TmaxバッファをSOAPメッセージに自動変換しますが、ユーザーが任意でsvrlibで変換したい場合はこのライブラリーを使用します。

下記は、ライブラリーで提供する関数の一覧です。

関数	説明
init_config	データの変換関数として、環境設定に指定します
convert_tmax_to_soap	Tmaxメッセージ(tmaxmsg)をSOAPメッセージ(soapmsg)に変換する関数です
convert_soap_to_tmax	SOAPメッセージ(soapmsg)をTmaxメッセージ(tmaxmsg)に変換する関数です

3.2.1. init_config

データ変換関数として、環境設定に指定します。

- プロトタイプ

```
int init_config(char *configfile, char *metafile);
```

- パラメータ

パラメータ	説明
configfile [in]	Webサービス環境設定ファイルのパスです
metafile [in]	Webサービス情報ファイルのパスです

3.2.2. convert_tmax_to_soap

Tmaxメッセージ(tmaxmsg)をSOAPメッセージ(soapmsg)に変換する関数です。

- プロトタイプ

```
int convert_tmax_to_soap(char *svcname, char *tmaxmsg, char **soapmsg, int *olen,  
  
                        char *subtype, char *uri, char *soapversion,  
                        char *encodingstyle, char *wsns, char *typens );
```

- パラメータ

パラメータ	説明
svcname [in]	サービス名です
tmaxmsg [in]	変換されるTmaxメッセージです
soapmsg [out]	関数内でメモリーを割り当てます
olen [out]	SOAPメッセージ(soapmsg)の長さです
subtype [in]	Tmaxメッセージ(tmaxmsg)が構成された構造体(struct)名です
uri [in]	Webサービス・サプライヤーのURIです
soapversion [in]	変換されるSOAPメッセージ(soapmsg)のバージョン(1.1 or 1.2)です
encodingstyle [in]	変換されるSOAPメッセージ(soapmsg)のエンコーディング・タイプ(rpc or doc)です
wsns [in]	変換されるSOAPメッセージ(soapmsg)の名前空間です
typens [in]	変換されるSOAPメッセージ(soapmsg)タイプの名前空間です

3.2.3. convert_soap_to_tmax

SOAPメッセージ(soapmsg)をTmaxメッセージ(tmaxmsg)に変換する関数です。

- プロトタイプ

```
int convert_soap_to_tmax(char *soapmsg, int soapmsglen, char **tmaxmsg, int *len);
```

- パラメータ

パラメータ	説明
soapmsg [in]	変換されるSOAPメッセージです
soapmsglen [in]	SOAPメッセージ(soapmsg)の長さです
tmaxmsg [out]	関数内でメモリーを割り当て(memory alloc)ます
len [out]	Tmaxメッセージ(tmaxmsg)の長さです

第4章 管理

本章では、Webサービス・ゲートウェイ(WSGW)のステータス情報を閲覧および再起動するコマンドとユーティリティについて説明します。

4.1. コマンド

コマンドは、ステータス情報を照会するためのコマンドと、動的設定のためのコマンドで分けられます。

4.1.1. wsgwinfo

WSGWのステータス情報は、Tmaxが提供する基本情報と追加されたコマンドで、tmadminで**wsgwinfo**を使って確認することができます。

tmadminでwsgwinfoを入力すると、Webサービス・ゲートウェイの情報が出力されます。その情報は、Webサービス・ゲートウェイが起動された時間(load_time)、現在接続中のクライアント数(current_client)、1つのサービスを処理するためにかかった最長時間(max_attach_time)です。

- 使用方法

使用方法是下記のとおりです。

```
wsgwinfo [-i svrid[ svrid]]
```

オプションを適用しない場合、すべてのWebサービス・ゲートウェイの情報を出力します。

- 例

[-i]オプションを適用する場合、次に入力されるsvridのWebサービス・ゲートウェイの情報のみ出力します。

```
wsgwinfo -i 0 1 2
```

4.1.2. wsgwreload

Tmaxの運用中に、サービス情報設定の変更、またはWebサービス・ゲートウェイ設定の変更が必要な場合があります。この場合、Tmaxシステムはtmadminの**wsgwreload**コマンドを実行して適用します。コマンドが実行されると、Webサービス・ゲートウェイは新たに要求されるサービスには応答せず、すでに処理中のサービスのみ処理した後、設定を適用してからサービスの処理を開始します。

sgwreloadを実行すると全体のWebサービス・ゲートウェイが設定を再ロードします。wsgwreloadに[-i]オプションを使って特定サーバーのID一覧を指定したら、指定されたWebサービス・ゲートウェイのみ変更された設定を適用します。

```
wsgwreload -i 0 1 2
```

4.2. ユーティリティ

下記は、Webサービスに関連して提供されるユーティリティの一覧です。

ユーティリティ	説明
tmmbfgen	サービス情報ファイル(text)をサービス情報バイナリ・ファイルとして作成するコマンドです
untmmbfgen	サービス情報バイナリ・ファイルをサービス情報ファイル(text)として作成するコマンドです
xwsdlgen	Webサービス仕様の中で、明細書の役割をするWSDL文を作成するコマンドです

4.2.1. tmmbfgen

サービス情報ファイル(text)をサービス情報バイナリ・ファイルとして作成するコマンドです。ユーザーが作成したサービス情報ファイルは、Webサービス・ゲートウェイとxwsdlgenでそのまま使用できないため、**tmmbfgen**を使って新規ファイルを作成する必要があります。

tmmbfgenによって作成されたファイルをサービス情報バイナリ・ファイルと称します。tmmbfgenを行うと、文法およびパラメータのタイプをチェックするため、Webサービス・ゲートウェイで参照する前に有効性をチェックすることが可能です。さらに、テキストを複数に分割して管理することができます。

● 使用方法

```
tmmbfgen [ -r text_file , ] [ -i text_file ] [ -d svc ] -o binary_file -r
```

項目	説明
<code>[-i text_file]</code>	サービス情報ファイル(text)を入力します
<code>[-d svc]</code>	削除するサービス名一覧を入力します
<code>-o binary_file</code>	作成する(変更する)サービス情報バイナリ・ファイル(default = sample)です

● 例

– 下記は、新しいサービス情報バイナリ・ファイルを作成する例です。

sampleを作成し、sample1.txtとsample2.txtに定義されているサービス情報をsampleに入力します。


```
tmmbfgen -r sample1.txt,sample2.txt -o sample
```

- 下記は、既存のサービス情報バイナリ・ファイルに追加する例です。

既存のsampleファイルのsample3.txtに定義されたサービス情報を追加します。同じサービスが存在する場合は上書きされます。

```
tmmbfgen -i sample3.txt -o sample
```

- 下記は、既存のサービス情報バイナリから特定のサービスを削除する例です。

sampleファイルからサービスSVC1、SVC2の情報を削除します。

```
tmmbfgen -d SVC1,SVC2 -o sample
```

4.2.2. untmmbfgen

untmmbfgenは、サービス情報バイナリ・ファイルをサービス情報ファイル(text)として作成するコマンドです。

- 使用方法

```
untmmbfgen -i binary_meta_file -o text_file
```

項目	説明
-i <i>binary_meta_file</i>	サービス情報バイナリ・ファイルを指定します
-o <i>text_file</i>	作成されるサービス情報ファイル(text)を指定します

- 例

下記は、untmmbfgenを使ってユーザーがサービス情報バイナリ・ファイルが確認できるように逆変換する例です。

```
untmmbfgen -i sample -o unsample.txt
```

4.2.3. xwsdlgen

Webサービス仕様の中で、明細書の役割をするWSDL文を作成するコマンドです。現在WSDL文は、1.1、2.0が存在しています。xwsdlgenは、Webサービス・ゲートウェイ環境設定ファイルとサービス情報バイナリ・ファイルを取得してWSDL文を作成します。

- 使用方法

```
xwsdlgen [options] -g wsgw_config_file -m binary_meta_file -o wsdl_file
```

- Options

項目	説明
w <i>version</i>	以下の値を設定します – 0 : WSDL1.1 (デフォルト) – 1 : WSDL2.0
b <i>binding_style</i>	以下の値を設定します – 0 : rpc (デフォルト) – 1 : document

- 設定項目

項目	説明
g <i>wsgw_config_file</i>	Webサービス・ゲートウェイ環境設定ファイルを設定します
m <i>binary_meta_file</i>	サービス情報バイナリ・ファイルを設定します
o <i>wsdl_file</i>	作成するWSDLファイルを設定します

第5章 例

TmaxサービスのECHOSTRING、ECHOSDL、ECHOFDL(インプットとアウトプットが同様なサービス)をWebサービスとして提供する方法を説明し、実際にAxis1.4を利用してクライアントを作成して実行します。

5.1. サービスの作成

5.1.1. サーバー・プログラム

ECHOサービスは、入力されたバッファと同一のバッファを返すサービスです。サーバー・プログラムを作成するために必要なファイルは、サーバー・ソース(sample.c)、sdl定義ファイル(demo.s)、fdl定義ファイル(demo.f)です。

<sample.c>

```
#include <stdio.h>
#include <usrinc/atmi.h>
#include <usrinc/fbuf.h>
#include "../sdl/demo_sdl.h"

ECHOSTRING(TPSVCINFO *msg)
{
    printf("client message : %s \n", msg->data );

    tpreturn(TPSUCCESS, 0, (char *)msg->data, msg->len, 0);
}

ECHOSDL(TPSVCINFO *msg)
{
    struct echo_s *rcvbuffer;
    struct echo_s test;
    struct echo_s test_s;

    rcvbuffer = (struct echo_s *)msg->data;

    printf("rcvbuffer\n");
    printf("char : [%d] \n"
           "short : [%d] \n"
           "int : [%d] \n"
           "float: [%f] \n"
```

```

        "double: [%lf] \n"
        "string: [%s] \n",
        rcvbuffer->charvalue,
        rcvbuffer->shortvalue,
        rcvbuffer->intvalue,
        rcvbuffer->floatvalue,
        rcvbuffer->doublevalue,
        rcvbuffer->stringvalue
    );

    tpreturn(TPSUCCESS, 0, (char *)msg->data, msg->len,0);
}

ECHOFDL(TPSVCINFO *msg)
{
    fprintf((FBUF *)msg->data);

    tpreturn(TPSUCCESS,0,(char *)msg->data, msg->len,0);
}

```

<demo.s>

下記は、sdl定義ファイルです。**sdlc**でビルドします。

```

struct echo_s{
    char charvalue;
    short shortvalue;
    int intvalue;
    float floatvalue;
    double doublevalue;
    char stringvalue[20];
};

```

<demo.f>

下記は、fdl定義ファイルです。**fdlc**でビルドします。

CHARVALUE	2100	char
SHORTVALUE	2101	short
INTVALUE	2102	int
FLOATVALUE	2104	float
DOUBLEVALUE	2105	double
STRINGVALUE	2106	string

5.2. Webサービスの設定

5.2.1. サービス情報ファイルの作成

<sample.txt>

```
SERVICE
    name = echostring
    tmaxname = ECHOSTRING
    intype = STRING
    outtype = STRING
REQUEST
    name = req
    type = string
    restriction = maxLength="19"
RESPONSE
    name = result
    type = string
    restriction = maxLength="19"

SERVICE
    name = echosdl
    tmaxname = ECHOSDL
    intype = STRUCT
    outtype = STRUCT
    instruct = echo_s
    outstruct = echo_s
REQUEST
    name = charvalue
    type = char
    name = shortvalue
    type = short
    name = intvalue
    type = int
    name = floatvalue
    type = float
    name = doublevalue
    type = double
    name = stringvalue
    type = string
    restriction = maxLength="19"
RESPONSE
    name = charvalue
    type = char
    name = shortvalue
```

```
type = short
name = intvalue
type = int
name = floatvalue
type = float
name = doublevalue
type = double
name = stringvalue
type = string
restriction = maxLength="19"
```

SERVICE

```
name = echofdl
tmaxname = ECHOFDL
intype = FDL
outtype = FDL
instruct = echofdlRequestFDL
outstruct = echofdlResponseFDL
```

REQUEST

```
name = CHARVALUE
type = char
maxoccurs = 50
name = SHORTVALUE
type = short
maxoccurs = 50
name = INTVALUE
type = int
maxoccurs = 50
name = FLOATVALUE
type = float
maxoccurs = 50
name = DOUBLEVALUE
type = double
maxoccurs = 50
name = STRINGVALUE
type = string
maxoccurs = 50
```

RESPONSE

```
name = CHARVALUE
type = char
maxoccurs = 50
name = SHORTVALUE
type = short
maxoccurs = 50
name = INTVALUE
type = int
maxoccurs = 50
```

```

name = FLOATVALUE
type = float
maxoccurs = 50
name = DOUBLEVALUE
type = double
maxoccurs = 50
name = STRINGVALUE
type = string
maxoccurs = 50

```

<sample.txt>ファイルを\$TMAXDIR/configディレクトリーに格納し、**tmmbfgen**を使ってサービス情報バイナリ・ファイルを作成します。

```
tmmbfgen -r sample.txt -o sample
```

5.2.2. Webサービス・ゲートウェイ設定ファイルの作成

<sample.xml>

\$endpointに、外部に表示されるアドレスを入力します。(例: 192.168.1.2:80)ファイルは、\$TMAXDIR/configディレクトリーに格納します。

```

<?xml version="1.0" encoding="UTF-8"?>
<Configuration xmlns="http://www.tmax.co.kr/twsp/twsgwconfig/">
  <Servicelist id="TmaxSampleECHO" >
    <Service name="echostring" />
    <Service name="echosdl" />
    <Service name="echofdl" />
  </Servicelist>
  <GWInstance>
    <HTTP address="$endpoint" />
  </GWInstance>
</Configuration>

```

5.3. Tmax環境ファイル

下記は、Tmax環境設定ファイルの例です。\$NODENAMEに該当するマシン名を入力し、\$TMAXDIRにTmaxがインストールされているディレクトリー・パスを入力します。

```

*DOMAIN
domain          SHMKEY    = 77521, MINCLH    = 1, MAXCLH = 2,
                  TPORTNO  = 15000, BLOCKTIME = 30, MAXSACALL=1024, MAXCACALL=1024

*NODE

```

```

$NODENAME TMAXDIR = "$TMAXDIR",
          APPDIR  = "$TMAXDIR/appbin",
          PATHDIR = "$TMAXDIR/path",
          TLOGDIR = "$TMAXDIR/log/tlog",
          ULOGDIR = "$TMAXDIR/log/ulog",
          SLOGDIR = "$TMAXDIR/log/slog",
          MAXCPC=2000, MAXGWCPC=300

*SVRGROUP
svg1      NODENAME = $NODENAME

*SERVER
sample SVGNAME = svg1

*SERVICE
ECHOSTRING SVRNAME=sample
ECHOSDL SVRNAME=sample
ECHOFDL SVRNAME=sample

*GATEWAY
WSGW1 GWTYPE=WSGW,
CLOPT="-csample.xml -msample",
NODENAME=$NODENAME,
DIRECTION=IN,
PORTNO=80

```

5.4. Axis1.4クライアント・プログラムの作成

5.4.1. WSDLファイルの作成

xwsdlgenユーティリティを利用して、wsdlファイルを作成します。

```
xwsdlgen -m sample -g sample.xml -o sample.wsdl
```

5.4.2. Stubの作成

Stubを作成するには、サーバーで作成した<sample.wsdl>ファイルが必要です。

下記のように、必要なjarファイルを、**AXIS_CLASS_PATH**に設定します。

```
java -cp $AXIS_CLASS_PATH:. org.apache.axis.wsdl.WSDL2Java
sample.wsdl -p echosamplesrpc
```


上記のコマンドを実行すると、echosamplesrpcフォルダーにクライアントの作成時に必要なファイルが生成されます。このパッケージを利用してクライアントを作成します。

参考

wsdl2javaの詳細内容は、[apache site](#)を参照してください。

<echoclient.java>

```
import java.rmi.RemoteException;
import javax.xml.rpc.ServiceException;

public class EchoClient
{
    public static void main(String[] args){
        String url = "HTTP://192.168.1.2:80";
        EchoClientWithRPC client_r = new EchoClientWithRPC(url);
        client_r.test();
    }
}

class EchoClientWithRPC
{
    private echosamplesrpc.TmaxSampleECHOLocator service;
    private echosamplesrpc.TmaxSampleECHOPortType port;

    public EchoClientWithRPC(String url)
    {
        try
        {
            service = new echosamplesrpc.TmaxSampleECHOLocator();
            service.setEndpointAddress(service.getTmaxSampleECHOPortWSDDServiceName()
,
                                url);
            port = service.getTmaxSampleECHOPort();
        }
        catch(ServiceException e)
        {
            e.printStackTrace();
        }
    }

    private void echostring()
    {
        try
        {
```

```

        String req = "abc";
        String response = port.echostring(req);
        System.out.println("echostring result > " + response);
    }catch(RemoteException e)
    {
        System.out.println(e.getMessage());
    }
}

private void echosdl()
{
    try
    {
        echosamplesrpc.Echo_s req = new echosamplesrpc.Echo_s();
        req.setCharvalue((byte)49);
        req.setShortvalue((short)2);
        req.setIntvalue(3);
        req.setFloatvalue(5.1f);
        req.setDoublevalue(6.1);
        req.setStringvalue("abc");
        echosamplesrpc.Echo_s response = port.echosdl(req);
        System.out.println("echosdl result >");
        System.out.println("char value > " + response.getCharvalue() +
            "\nshort value > " + response.getShortvalue() +
            "\nint value > " + response.getIntvalue() +
            "\nfloat value > " + response.getFloatvalue() +
            "\ndouble value > " + response.getDoublevalue() +
            "\nstring value > " + response.getStringvalue());
    }catch(RemoteException e)
    {
        e.printStackTrace();
    }
}

private void echofdl()
{
    try
    {
        echosamplesrpc.EchofdlRequestFDL echofdlRequestFDL =
            new echosamplesrpc.EchofdlRequestFDL();
        echofdlRequestFDL.setCHARVALUE(new byte[]{(byte)49});
        echofdlRequestFDL.setSHORTVALUE(new short[]{(short)2});
        echofdlRequestFDL.setINTVALUE(new int[]{3});
        echofdlRequestFDL.setFLOATVALUE(new float[]{5.1f});
        echofdlRequestFDL.setDOUBLEVALUE(new double[]{6.1});
        echofdlRequestFDL.setSTRINGVALUE(new String[]{"abc"});
    }
}

```

```

        echosamplesrpc.EchofdlResponseFDL response = port.echofdl(echofdlRequestFDL);

        System.out.println("echofdl result >");
        System.out.println("char value > " + response.getCHARVALUE()[0] +
            "\nshort value > " + response.getSHORTVALUE()[0] +
            "\nint value > " + response.getINTVALUE()[0] +
            "\nfloat value > " + response.getFloatVALUE()[0] +
            "\ndouble value > " + response.getDoubleVALUE()[0] +
            "\nstring value > " + response.getStringVALUE()[0]);
    } catch (RemoteException e)
    {
        e.printStackTrace();
    }
}

public void test()
{
    echostring();
    echosdl();
    echofdl();
}
}

```


付録 A. 内部構造体型のサポート

構造体型(struct type)にchar[]で宣言し、サービス情報ファイルは、char[]ではなく別のデータ型の場合でも変換することが可能です。sdlと情報ファイルを下記のように設定する場合、変換することができます。

sdlを下記のように設定します。

```
struct qmc_rq{
    char value1[4];
    /* struct depth1 */
    char value2[4];
    /* struct depth2 */
    char value3[4];
    char value4[4];
    /* end depth2 */
    char value5[4];
    /* end depth1 */
    char value6[4];
};

struct qmc_rps{
    char value1[4];
    /* struct r_depth1 */
    char value2[4];
    char value3[4];
    /* end r_depth1 */
    char value4[4];
};

struct LANG_POLICI01 {
    char Xid_length[2];
    char Xid[255];
    char customerId_length[2];
    char customerId[255];
    char userId_num[4];
    char userId_cont[16];
    char loopback;
    char freeFormAddress_num[4];
    char freeFormAddress_cont[16];
    char policyType_num[4];
    char policyType_cont[16];
};
```

```

struct LANG_POLICO01 {
    char Xid_length[2];
    char Xid[255];
    char customerId_length[2];
    char customerId[255];
    char Notification_num[4];
    char Notification_cont[16];
    char standardisedAddressAnd_num[4];
    char standardisedAddressAnd_cont[16];
}

```

情報ファイルを下記のように設定します。

```

STRUCT
    name = depth1

    TYPE
        name = value1
        type = int
        name = value2
        type = struct
        subtype = depth2
        name = value3
        type = int

STRUCT
    name = depth2

    TYPE
        name = value1
        type = int
        name = value2
        type = int

STRUCT
    name = r_depth1

    TYPE
        name = value1
        type = int
        name = value3
        type = int

SERVICE
    name = TEST_STRUCT
    tmaxname = TEST_STRUCT
    intype = STRUCT
    outtype = STRUCT

```

```
instruct = qmc_rq
outstruct = qmc_rps
```

REQUEST

```
name = value1
type = int
name = value2
type = struct
subtype = depth1
name = value3
type = int
```

RESPONSE

```
name = value1
type = int
name = value2
type = struct
subtype = r_depth1
name = value3
type = int
```

SERVICE

```
name = TOUPPER
tmaxname = TOUPPER
intype = STRING
outtype = STRING
```

REQUEST

```
name = req
type = string
restriction = maxLength="19"
```

RESPONSE

```
name = result
type = string
restriction = maxLength="19"
```

SERVICE

```
name = TOUPPERSDL1
tmaxname = TOUPPERSDL1
intype = STRUCT
outtype = STRUCT
instruct = wsgw_s
outstruct = wsgw_s
```

REQUEST

```
name = req
type = string
name = intvalue
type = int
name = shortvalue
type = short
name = floatvalue
type = float
name = doublevalue
type = double
name = charvalue
type = char
```

RESPONSE

```
name = result
type = string
name = intvalue
type = int
name = shortvalue
type = short
name = floatvalue
type = float
name = doublevalue
type = double
name = charvalue
type = char
```

SERVICE

```
name = TOUPPERSDL
    tmaxname = TOUPPERSDL
    intype = STRUCT
    outtype = STRUCT
    instruct = kstrdata
    outstruct = kstrdata
```

REQUEST

```
name = len
type = int
name = str
type = string
```

RESPONSE

```
name = len
type = int
name = str
type = string
```


STRUCT

```
name = requestHeaderI
TYPE
name = Xid-length
type = short
name = Xid
type = string
name = customerId-length
type = short
name = customerId
type = string
name = userId-num
type = int
name = userId-cont
type = string
name = loopback
type = char
```

STRUCT

```
name = GetStandardisedAddressAndTerI
TYPE
name = requestHeader
type = struct
subtype = requestHeaderI
name = freeFormAddress-num
type = int
name = freeFormAddress-cont
type = string
name = policyType-num
type = int
name = policyType-cont
type = string
```


索引

シンボル

*reply_from_remote(), 17
*reply_from_tmax(), 16
*request_from_remote(), 16
*request_from_tmax(), 15

C

convert_soap_to_tmax(), 19
convert_tmax_to_soap(), 18

G

GATEWAYセクション, 14

I

init_config(), 18

R

REQUESTセクション, 6
RESPONSEセクション, 6

S

SERVICEセクション, 5
STRUCTセクション, 7
Stubの作成例, 30

T

Tmax環境設定, 14
tmmbfgen, 8, 22
TYPEセクション, 7

U

untmmbfgen, 9, 23

W

Webサービスの特徴, 2

Webサービスの設定例, 27
Webサービスの長所, 2
Webサービス・ゲートウェイ, 2
Webサービス・サプライヤー, 3
Webサービス・リクエスター, 4
WSDLファイルの作成, 30
WSGW, 2
wsgwinfo, 21
wsgwreload, 21
WSGW設定ファイル, 11
WSGW設定ファイルの例, 13
WSGW設定ファイル要素
 <in>, 11
 <out>, 12

X

xwsdlgen, 24

か

コールバック関数

*reply_from_remote(), 17
*reply_from_tmax(), 16
*request_from_remote(), 16
*request_from_tmax(), 15

さ

サービスの作成例, 25
サービス情報ファイル, 5
サービス情報ファイルの例, 9

た

データ変換ライブラリー

convert_soap_to_tmax(), 19
convert_tmax_to_soap(), 18
init_config(), 18

