

Tmax ゲートウェイガイド (TCP/IP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 (TCP/IP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
1.2. サービスのタイプ	2
1.2.1. Tmaxからのサービス要求	3
1.2.2. リモートノードからのサービス要求	5
第2章 環境設定	9
2.1. 概要	9
2.2. Tmax環境構成	9
2.2.1. ACK/NAK応答のSVCGW	11
2.2.2. 遅延処理のSVCGW	12
2.3. SVCGWの環境ファイル	12
2.3.1. アドレス情報の環境ファイル	12
第3章 COMライブラリー	15
3.1. 概要	15
3.2. ComOpen	15
3.3. ComDetach	16
3.4. ComClose	16
3.5. ComSend	17
3.6. ComRecv	18
第4章 例	19
4.1. NOREPLYサービス呼び出し	19
4.1.1. 環境ファイル	19
4.1.2. リモートノード	20
4.2. REPLYサービス呼び出し	23
4.2.1. 環境ファイル	23
4.2.2. リモートノード	24
4.3. リモートノード同期型呼び出し	27
4.3.1. 環境ファイル	28
4.3.2. リモートノード	29
4.4. ACK/NAKの通信方式	30
4.4.1. 環境ファイル	31
4.4.2. リモートノード	32
付録 A. ヘッダー・ファイル	35
A.1. COMMHEADファイル	35
索引	37

図目次

[図 1.1]	SVCGWの動作構造	1
[図 1.2]	NOREPLYサービス呼び出しのSVCGWの動作構造	4
[図 1.3]	REPLYサービス呼び出しのSVCGWの動作構造	5
[図 1.4]	リモート同期型呼び出しのSVCGW動作構造	6
[図 1.5]	リモート非同期型呼び出しのSVCGW動作構造	7

このガイドについて

対象読者

本書は、Tmax[®](以下、Tmax)TCP/IPサービス・ゲートウェイを使用して開発するユーザーを対象としています。同書では、Tmaxサーバーと非Tmaxサーバー間のTCP/IP通信時にインターフェースを担当するTmaxが提供するTCP/IPサービス・ゲートウェイについて説明しています。

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

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4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

システムの概要およびサービスのタイプについて説明します。

- 第2章: 環境設定

TCP/IPサービス・ゲートウェイの環境構成および環境ファイルについて説明します。

- 第3章: COMライブラリー

TCP/IPサービス・ゲートウェイで使用するAPI関数について説明します。

- 第4章: 例

TCP/IPサービス・ゲートウェイの各役割に関する例について説明します。

- 付録A: ヘッダー・ファイル

COMMHEADファイル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、システムの概要およびサービスのタイプについて説明します。

1.1. 概要

TCP/IPサービス・ゲートウェイ(以下、SVCGW)は、Tmaxサーバーと非Tmaxサーバー(以下、リモートノード)間のTCP/IP通信時にインターフェースを担当するTmaxが提供するゲートウェイです。

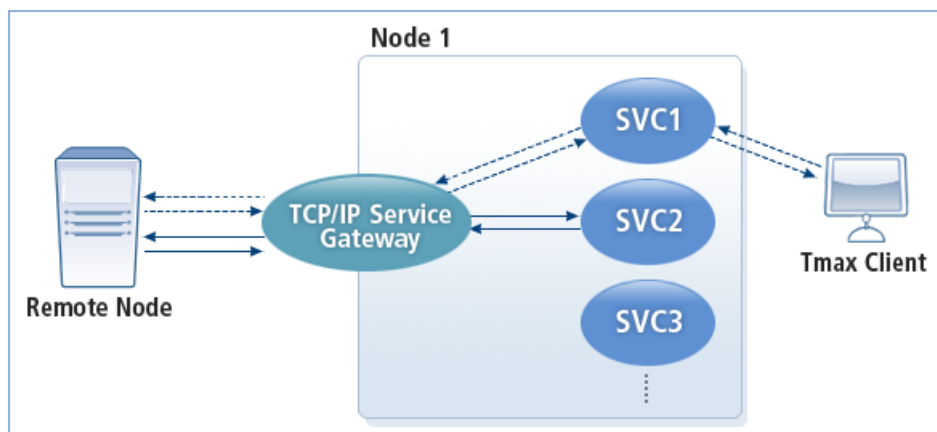
SVCGWは、大きく2つのモジュールで分けられます。1つは、TmaxでリモートノードとTCP/IPで送受信する部分です。もう1つは、リモートノードでTmaxのSVCGWとの通信を担当する、TCP/IP通信ライブラリー(以下、COMライブラリー)です。2つのモジュールを利用して、Tmaxシステムと非TmaxシステムがTCP/IPで通信できるようにするのが、TCP/IPサービス・ゲートウェイです。

SVCGWのユーザーがTCP/IP通信に関連するプログラムを作成しなくても、SVCGWとTCP/IPで通信できるようにCOMライブラリーを提供します。ライブラリーを利用して、容易にSVCGWからサービスを受けたり、サービスを要求したりすることができます。SVCGWは、リモートノードからのメッセージを受け、ユーザーが要求したサービスにtpacallし、サービス結果は最初に要求したリモートノードに転送します。一方、TmaxサービスでSVCGWにtpcallまたは他の方式でサービスを要求すると、SVCGWはリモートノードに要求メッセージを転送し、応答が受信されたら自身を呼び出したサービスにtpreturnします。ソケットを開き、メッセージを送受信するなど、他のシステムとTCP/IPで接続するために必要な複雑な作業は、すべてTCP/IPサービス・ゲートウェイとCOMライブラリーで処理するため、開発者は業務ロジックのみ作成します。

SVCGWの動作方式は、Tmaxのサービスまたはクライアントからリモートノードにサービスを要求する方式と、リモートノードでTmaxのサービスを呼び出す方法で分けられます。

下記は、SVCGWの動作構造です。

[図 1.1] SVCGWの動作構造



- Tmaxからのサービス要求

[図 1.1]の点線で表示されている部分です。Tmaxクライアントまたはサービスからサービス要求を受け、リモートノードにサービスを要求します。このようなサービスを**インバウンド・サービス**といいます。インバウンドの場合は、Tmaxのクライアントまたはサービスで、SVCGWに登録されているサービス名を利用してリモートノードにサービスを要求することができます。

SVCGWは、Tmaxからリモートノードに要求する方式は非同期型方式で処理します。すなわち、サービスまたはクライアントがリモートにサービスをtpcallで要求しても、SVCGWはリモートにサービス要求を転送し、サービスまたはクライアントに直ちに正常応答を返します。その後、リモートノードでは、サービス要求に対する応答であっても新しいサービス要求と見なします。したがって、Tmaxで要求するすべてのサービスは非同期型方式で動作します。

- リモートノードからのサービス要求

[図 1.1]の実線で表示されている部分です。TCPGWはリモートノードからサービス要求を受け、処理することができます。このようなサービスを**アウトバウンド・サービス**といいます。アウトバウンドの場合は、リモートノードでTmaxのサービス名を利用してサービスを呼び出します。

リモートノードでTmaxのサービスを要求する場合は、非同期型、同期型の両方とも使用できます。非同期型は、COMライブラリーを利用してSVCGWにサービスを要求する際、flagsにCOMMNOREPLYを設定すると、SVCGWはTmaxにユーザーが要求したサービスを呼び出すのみで、応答は受けません。同期型方式は、COMライブラリーを利用してSVCGWにサービスを要求する際、flagsにCOMMNOREPLYを設定せずに要求すると、SVCGWはTmaxにユーザーが要求したサービスを呼び出し、応答が受信されたらサービスを要求したリモートノードに転送します。

なお、リモートノードからTmaxにサービスを要求する際、遅延処理方式をサポートします。flagsにCOMMDelayを設定すると、SVCGWは以後ユーザーがRQのデータを一括処理できるようにRQに要求データを保存します。遅延処理は、常に非同期型方式で処理します。

TmaxにインストールされているSVCGWは、Tmaxサーバーの一種です。SVCGWを使用するには、Tmax環境ファイルにサーバーとして登録する必要があります。一般的には、TCSまたはUCS用サーバー・ライブラリーを利用してサーバーを作成しますが、SVCGWはユーザーが直接使用できるように実行可能なファイルで提供します。そのため、ユーザーはSVCGWをTmax環境ファイルにサーバーとして登録する作業のみ行います。Tmax環境設定についての詳細内容は、「[第2章 環境設定](#)」を参照してください。

1.2. サービスのタイプ

SVCGWはリモートノードとTCP/IPで通信します。Tmaxクライアントの要求をリモートノードに転送するか、リモートノードの要求をTmaxサービスで処理できるようにする役割をします。

この節では、Tmaxのインストールによって提供されるサービスのタイプについて説明します。リモートノードで使用するCOMライブラリーについては、「[第3章 COMライブラリー](#)」を参照してください。

SVCGWの動作方式は、Tmaxのサービスまたはクライアントからリモートノードにサービスを要求する方式と、リモートノードからTmaxのサービスを呼び出す方式で分けられます。SVCGWは、常にサーバー・モードで動作しており、リモートノードはクライアント・モードで動作する必要があります。リモートノードと接続されたら、どちらからでも先にサービスを要求することができます。

1.2.1. Tmaxからのサービス要求

Tmaxのサービスまたはクライアントからリモートノードにサービスを要求する場合は、リモートノードにサービス要求を転送し応答は受けない方式と、サービス要求を転送し当該応答を新しい要求として処理する方式で分けられます。

SVCGWは、指定されたリモートノードへのみサービスが要求できるため、ユーザーが事前にサービスを要求しようとするリモートノードを把握しておく必要があります。そのため、Tmax環境ファイルにサービス名を登録し、当該サービス名をSVCGW環境ファイルのリモートノード別情報に登録します。SVCGWはサービス要求を受けると、要求したサービス名を使ってリモートノードを検索し、該当のリモートノードにデータを転送します。

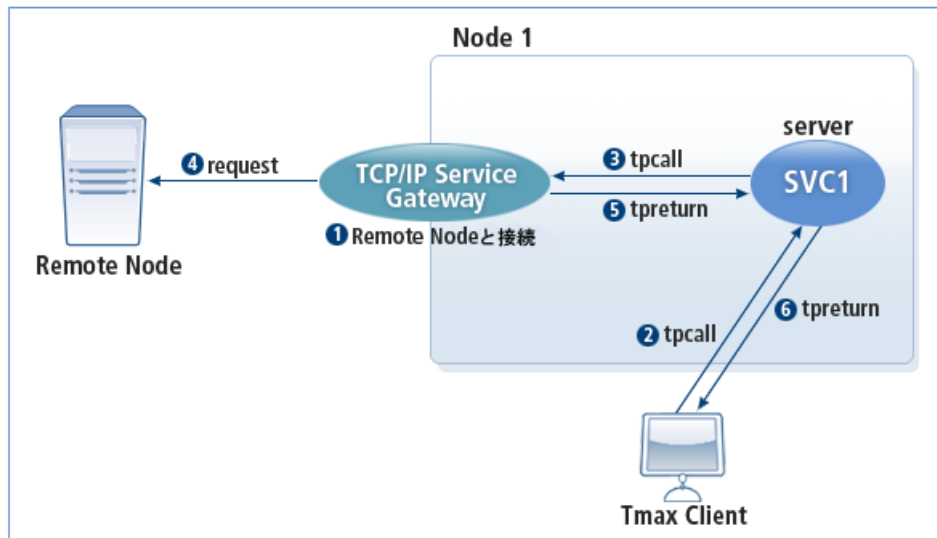
NOREPLYサービス呼び出し

NOREPLYサービス呼び出し方式は、TmaxのサービスまたはクライアントからSVCGWにtpcallでサービスを要求し、結果を問わず処理する方式です。

Tmaxクライアントの要求を受けたサービスからSVCGWにtpcallすると、SVCGWはリモートノードにサービスを要求し、リモートノードの結果を問わず直ちに返します。このような方式でSVCGWを動作させると、Tmaxクライアントで呼び出したTmaxサービスはSVCGWを呼び出した後、結果を問わず継続して処理することになります。

下記は、NOREPLY方式でサービスを要求する場合のSVCGWの動作構造です。

【図 1.2】 NOREPLYサービス呼び出しのSVCGWの動作構造



1. SVCGWとリモートノードが接続されています。
2. TmaxクライアントはTmaxサービスをtpcallします。
3. Tmaxサービスではクライアントの要求を受け、SVCGWにサービスをtpcallします。

SVCGWには多数のサービスが登録されているため、ユーザーが要求しようとするリモートノードと接続されているサービス名を指定する必要があります。

4. SVCGWは、ユーザーが指定したサービスと接続されているリモートノードにサービスを要求します。
5. SVCGWは、サービスを要求したサービスに応答します。
6. SVCGWは、サービスを要求したTmaxクライアントに正常応答をtpreturnします。

REPLYサービス呼び出し

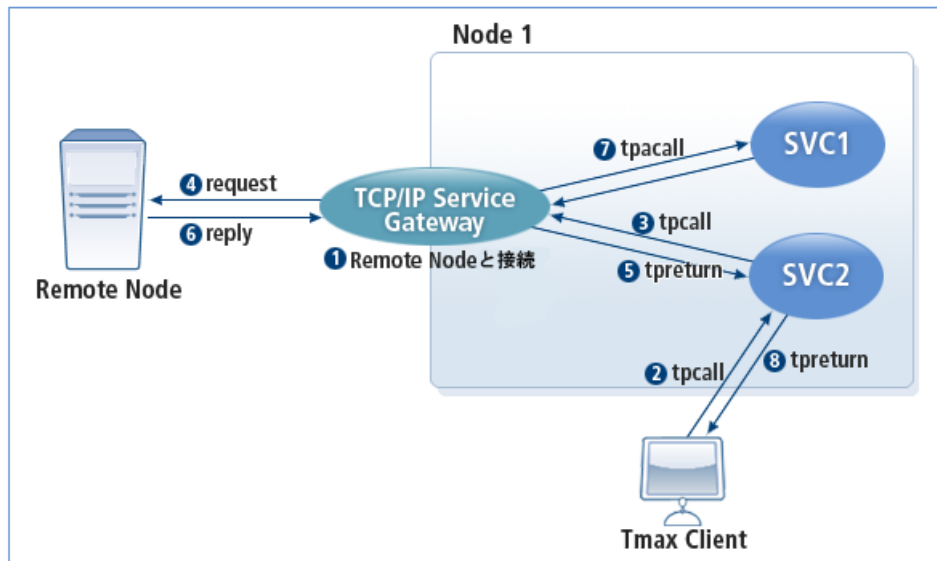
REPLYサービス呼び出しは、最初にサービスを要求するTmaxのサービスまたはクライアントがSVCGWにtpcallでサービスを要求し、結果を問わず処理します。リモートノードが要求サービス进行处理し、その結果でTmaxの新しいサービス呼び出す方式です。

Tmaxクライアントの要求を受けたサービスからSVCGWにtpcallすると、SVCGWはリモートノードにサービスを要求し、リモートノードの結果を問わず直ちに返します。リモートノードは処理結果をもってTmaxの新しいサービス呼び出します。新しいサービス呼び出す場合は、ComSend関数のflagsに設定する値によって、Tmaxのサービス処理結果を受けることも受けないこともできます。

REPLYサービス呼び出し方式は、最初にサービス呼び出すサービスと、結果を処理するサービスを区別するために使用します。

下記は、REPLY方式でサービスを呼び出す場合のSVCGWの動作構造です。

[図 1.3] REPLYサービス呼び出しのSVCGWの動作構造



1. SVCGWとリモートノードが接続されています。

2. TmaxクライアントはTmaxサービスをtpcallします。

3. Tmaxサービスではクライアントの要求を受け、SVCGWにサービスをtpcallします。

SVCGWには多数のサービスが登録されているため、ユーザーが要求しようとするリモートノードと接続されているサービス名を指定する必要があります。

4. SVCGWは、ユーザーが指定したサービスと接続されているリモートノードにサービスを要求します。

5. SVCGWは、サービスを要求したサービスに正常応答をtpreturnします。

6. リモートノードから結果に対する新しいサービス要求を受けます。

7. SVCGWは、ユーザーがTmaxに指定したサービスを呼び出します。

8. Tmaxサービスは、サービスを要求したTmaxクライアントに正常応答をtpreturnします。

1.2.2. リモートノードからのサービス要求

リモートノードでTmaxのサービスを呼び出します。この方式は、同期型方式と非同期型方式で分けられます。同期型は、サービスを要求して応答を待つ方式であり、非同期型は、サービスを要求するが応答は待た

ない方式です。同期型方式を利用する際、サービスを要求して応答を受ける前に、他のサービス要求が当該リモートノードに転送されることもあるため、これに対する処理は適切に行う必要があります。

リモート同期型呼び出し方式

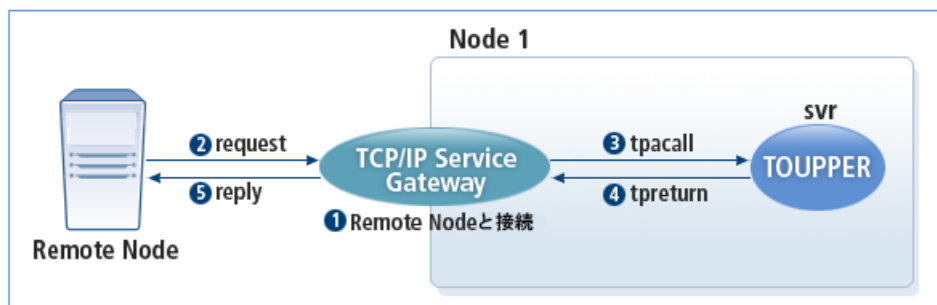
リモート同期型呼び出しは、リモートノードからSVCGWに先にサービスを要求し、応答を待つ方式です。SVCGWはリモートノードから要求したサービスを呼び出し、その結果を受けて当該サービスを要求したチャンネルに転送します。

リモートノードはSVCGWに、Tmax環境ファイルに定義されているMAXSACALL数を超過して呼び出すことができません。リモートノードのユーザーは、COMライブラリーを利用してSVCGWにサービスを呼び出し、応答を受けます。さらに、ComSend時はflagsにいかなる設定もしてはいけません。

このような動作方式は、リモートノードでTmaxのサービスを呼び出す最も一般的な方式です。SVCGWは、リモートノードのチャンネル情報を保存しておき、サービスから結果が受信されたら保有中のチャンネルから該当のチャンネルを見つけ結果を転送します。このとき、該当のチャンネルに結果を返す前に他のサービスを要求することができます。すなわち、リモートノードで要求したチャンネルはブロックされないため、要求に対する応答がリモートノードに転送される前に他のサービス要求がリモートノードに転送される場合があるので、SVCGWはこれも処理しなければなりません。

下記は、リモート同期型方式でサービスを呼び出す場合のSVCGWの動作構造です。

[図 1.4] リモート同期型呼び出しのSVCGW動作構造



1. SVCGWとリモートノードが接続されています。
2. リモートノードは、SVCGWと接続されているチャンネルにメッセージを転送します。
3. SVCGWは、tpacallでTmaxサービスを呼び出します。
4. SVCGWはサービス処理結果を受け、メッセージを要求したチャンネルを検索します。
5. 該当のチャンネルが正常に接続されているなら結果を転送します。

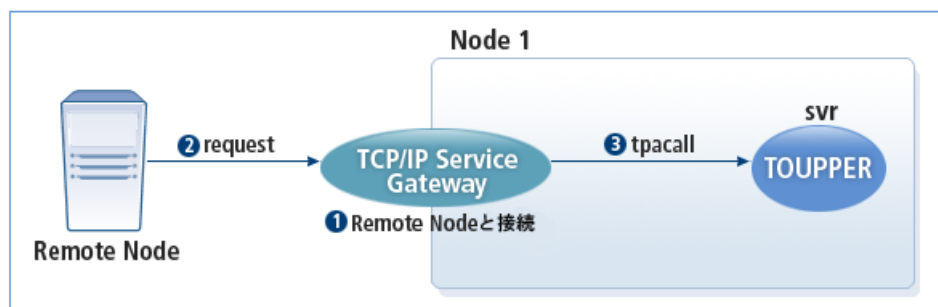
リモート非同期型呼び出し方式

リモート非同期型呼び出しは、リモートノードからSVCGWに先にサービスを要求するが、応答は待たない方式です。SVCGWはリモートノードで要求したサービスを呼び出し、その結果は受けません。

この方式で処理するためにリモートノードでは、ComSend時にflagsにCOMMNOREPLYを設定する必要があります。COMMNOREPLYが設定されていると、SVCGWはTmaxにサービスを要求するが応答は受けないように処理します。

下記は、リモート非同期型方式でサービスを呼び出す場合のSVCGWの動作構造です。

[図 1.5] リモート非同期型呼び出しのSVCGW動作構造



1. SVCGWとリモートノードが接続されています。
2. リモートノードは、SVCGWと接続されているチャンネルにメッセージを転送します。
3. SVCGWは、tpacallでTmaxサービスを呼び出します。

第2章 環境設定

本章では、TCP/IPサービス・ゲートウェイの環境構成および環境ファイルについて説明します。

2.1. 概要

SVCGWサーバーを構成するには、下記のようなファイルが必要となります。

下記は、UNIXで使用される環境ファイルです。

ディレクトリー	ファイル名
Lib	tmaxcom.a、tmaxcom.so
Lib64	ttmaxcom.a、tmaxcom.so
Appbin	tcpsvcgw

下記は、Windowsで使用される環境ファイルです。

ディレクトリー	ファイル名
Lib	tmaxcom.lib、tmaxcom.dll
bin	tcpsvcgw.exe

2.2. Tmax環境構成

SVCGWを使用するには、Tmax環境ファイルにSVCGWをサーバーとして登録する必要があります。Tmaxサーバーのうち、UCS型サーバーと登録方法が類似していますが、SVRTYPEがUCSでCUSTOM_GATEWAYという点が異なります。SVCGWを使用するためにTmax環境ファイルを修正するときは、SERVER、SERVICEセクションのみ設定します。

下記は、Tmax環境ファイルの例です。

```
*DOMAIN
tmax    SHMKEY = 88000,
        MINCLH = 1,
        MAXCLH=1,
        TPORTNO=8800

*NODE
tmax1   TMAXDIR="/home/tmax",
        APPDIR="/home/tmax/appbin"
```

```
*SVRGROUP
svg1      NODENAME=tmax1

*SERVER
tcpsvcgw  SVGNAME=svg1,
          MIN=1,
          MAX=1,
          CPC=10,
          SVRTYPE=CUSTOM_GATEWAY,
          CLOPT="-- -P 8090 -F /home/tmax/config/tcpsvcgw.cfg -N 20"

*SERVICE
TCPSVC1   SVRNAME= tcpsvcgwgv
TCPSVC2   SVRNAME= tcpsvcgw
```

項目	説明
MIN	SVCGWのプロセス数を指定する項目です。値は常に1に指定します
CPC	TmaxエンジンとSVCGW間のチャンネル数を指定します。Tmaxのクライアントまたはサービスからリモートノードに要求する場合は、同時に要求する数の分だけCPC数を指定します。逆の場合、SVCGWはTmaxエンジンにtpacallで要求するため、多くのチャンネルを使用する必要はありません
CLOPT	CLOPT項目(SVCGWオプション) の内容を参照してください
SVCTIME	SVCGWは、リモートノードに要求したサービスに対してサービス・タイムアウトを適用しません。その理由は、リモートノードにはサービスを要求するのみで、応答は別途のサービスで処理するためです。

CLOPT項目(SVCGWオプション)

SVCGWはTmax設定ファイルに登録できる項目が制限されているため、CLOPT項目にいくつかのオプションを設定する必要があります。下記のオプションによってSVCGWの動作方式が異なるため、以下の説明を十分に理解する必要があります。

オプション	説明
[-F]	<p>VCGWで使用するリモートノードの情報およびリモートノードを見つけるサービス名が登録されているファイルを指定します。SVCGWはこのオプションに指定したファイルの情報をロードしてリモートノードの情報を保存しておき、Tmaxのサービスまたはクライアントからリモートノードにサービスを要求する際、要求したサービス名でリモートノードを検索する場合に使用します。</p> <p>ファイルの登録方法については、「2.3. SVCGWの環境ファイル」を参照してください</p>

オプション	説明
[-P]	SVCGWは常にサーバー・モードでのみ動作します。リモートノードは、SVCGWが実行されるサーバーのアドレスとオプションに指定するポート番号を利用して接続します。SVCGWは複数のポート番号が同時にListenできるため、オプションにリモートノードで接続を希望するポートをすべて登録します(例: “8880,8881,8882”)
[-q]	リモートノードでTmaxのサービスを要求するとき、遅延処理方式で 사용할 수 있습니다。すなわち、Tmaxが提供するRQに一時データを保存しておき、以後一括処理する方式でSVCGWが使用できますが、このとき使用するRQ名を登録するオプションです
[-A]	SVCGWとリモートノード間の送受信データに対して、ACK/NAKメッセージの送受信有無を指定するオプションです。このオプションを指定すると、送受信データに対してACK/NAKをやり取りします。TmaxのSVCGWは、内部的にACK/NAKメッセージを処理しますが、リモートノードのCOMライブラリーではACK/NAKを処理しないため、ユーザー側のプログラムでこれが処理できる部分が必要となります(デフォルト値: ACK/NAKメッセージを送受信しない)
[-N]	[-F]オプションにリモートノードの情報を登録するか、ComOpenする場合にリモートノードを登録することができます。この場合、環境ファイルへの登録は不要です。しかし、ComOpenする際、登録するリモートノードの総数が把握できない問題が発生します。それを解決するために、このオプションにリモートノードの総数を指定します。 環境ファイルに登録されたリモートノード数とオプションに登録されたオプション数を比較して多いほうを選択します

2.2.1. ACK/NAK応答のSVCGW

ACK/NAKメッセージを受ける場合のSVCGW環境設定は下記のとおりです。

```
*DOMAIN
...
*NODE
...
*SVRGROUP
...
*SERVER
tcpsvcgw  SVGNAME=svg1,
          MIN=1,
          MAX=1,
          CPC=10,
          SVRTYPE=CUSTOM_GATEWAY,
          CLOPT="-- -P 8090 -F /home/tmax/config/tcpsvcgw.cfg -N 20 -A "
```

```
*SERVICE
TCPSVC1      SVRNAME= tcpsvcgw
```

上記のように環境ファイルを作成した場合、tcpsvcgwという名前のSVCGWの1つがTmaxのブート時に起動されます。tcpsvcgwのサービス名は、TCPSVC1です。SVCGWはリモートノードとのデータ送受信時にACK/NAKメッセージをやり取りするため、リモートノード側のプログラムの場合は、ACK/NAKに関する部分まで考慮する必要があります。

2.2.2. 遅延処理のSVCGW

RQを利用して遅延処理方式でSVCGWを使用する場合の環境設定は下記のとおりです。

```
*DOMAIN
...
*NODE
...
*SVRGROUP
...
*SERVER
tcpsvcgw  SVGNAME=svg1,
          MIN=1,
          MAX=1,
          CPC=10,
          SVRTYPE=CUSTOM_GATEWAY,
          CLOPT="-- -P 8090 -F /home/tmax/config/tcpsvcgw.cfg -N 20 -q testq "

*SERVICE
TCPSVC1    SVRNAME= tcpsvcgw
```

上記のように環境ファイルを作成し、リモートノードでCOMライブラリーを利用して(flagsにCOMMDELAYを設定)Tmaxにサービスを要求すると、SVCGWは要求データをtestqに保存します。SVCGWはリモートノードから受信したデータをRQに保存するのみで、いかなる処理もしないため、ユーザーはRQに保存されているデータを処理する必要があります。

2.3. SVCGWの環境ファイル

Tmax環境ファイルのCLOPTセクションに[-F]オプションで指定したファイルは、下記で説明するフォーマットで登録する必要があります。このファイルはリモートノードのアドレスとサービス名を登録するファイルです。

2.3.1. アドレス情報の環境ファイル

SVCGWは環境ファイルに登録された数と、[-N]オプションで指定した数のうち、大きい値を基準にして内部的にバッファーを割り当て、多重化(multiplexing)方式で処理します。

下記は、アドレス情報の環境ファイルと項目についての説明です。

```
#####  
#          TCP/IP Service Gateway Config          #  
#####  
# gwno  ipaddr  Tmaxsvc  
# 0      1.1.1.1  SVC  
#####  
# line start with "#" is comment line  
# gwno must start at 0 and be increased by 1  
# ipaddr: remote node ipaddr  
# svc:   tmax service name  
#####  
0      1.1.1.1  SVC1
```

項目	説明
gwno(ゲートウェイ番号)	Tmax環境ファイルのMIN値が2以上の場合に使用します。MIN値は常に1なので、この項目はゼロに指定する必要があります
ipaddr	リモートノードのアドレスを登録します。SVCGWは登録されたアドレスを利用してリモートノードから接続する際に関連したサービス名を検索します
Tmaxsvc(サービス名)	<p>Tmaxのクライアントまたはサービスで特定のリモートノードにサービスを要求するためには、SVCGWに送信しようとするリモートノードを知らせる必要があります。このような場合に使用する項目で、指定されたサービス名でSVCGWを呼び出すと、当該サービス名を使ってリモートノードを見つけ、該当のリモートノードにデータを転送します。</p> <p>項目に指定したサービス名は、Tmax環境ファイルのSERVICEセクションに登録されているサービス名である必要があります。SVCGWがリモートノードを見つけるには、登録したサービス名を利用する方法と、以後リモートノードからSVCGWに接続を設定する際にリモートノードの名前を登録する方法があります。このように登録するリモート名は、Tmax環境ファイルのSERVICEセクションに登録されている名前と一致する必要があります</p>

第3章 COMライブラリー

本章では、COMライブラリーが提供するAPIの使用方法について説明します。COMライブラリーのプロトタイプは、comhead.hヘッダー・ファイルに定義されています。

3.1. 概要

COMライブラリーは、リモートノードがTmaxのSVCGWとTCP/IPで通信できるように、Tmaxが提供するライブラリーです。SVCGWとの通信を希望するリモートノードは、必ずCOMライブラリーを使用して通信します。リモートノードのユーザーは、COMライブラリーを利用してTmaxとTCP/IPでサービスを要求するか、またはサービス要求を受けることができます。なお、Tmaxと通信する際、通信関連部分はCOMライブラリーで処理するため、簡単にTmaxのサービスを利用することができます。

リモートノードがTmaxとTCP/IPで通信する場合、常にTmaxがサーバー・モードになり、リモートノードはクライアント・モードになります。

下記は、COMライブラリーが提供するAPIの一覧です。

API	説明
ComOpen	TmaxのSVCGWとTCP/IPで接続を確立し、リモートノードをSVCGWに登録する関数です
ComDetach	TmaxのSVCGWとの登録を解除するために使用する関数です
ComClose	TmaxのSVCGWとの登録を解除し、接続を切断する関数です
ComSend	リモートノードからTmaxのSVCGWにデータを転送する関数です
ComRecv	TmaxのSVCGWからデータを受信する関数です

3.2. ComOpen

TmaxのSVCGWとTCP/IPで接続を確立し、リモートノードをSVCGWに登録する関数です。TmaxのSVCGW環境ファイルのサービス名項目に、この関数を使用してサービス名を登録する必要があります。サービス名が登録されていない場合、リモートノードからTmaxにサービスを要求することが可能ですが、Tmaxからリモートノードにサービスを要求することはできません。ユーザーは他の関数を使用する前に、必ずこの関数を使用してTmaxのSVCGWを接続します。

● プロトタイプ

```
int ComOpen(int type, char *id, char *port)
```

- パラメータ

パラメータ	説明
type	Tmaxと通信が使用するプロトタイプ(TCPIP/SERIAL)です。各プロトタイプは、comhead.hファイルに登録されています
id	リモートノードを識別するコードです。このコードは上述したSVCGW環境ファイルと関連しています。環境ファイルにサービス名を登録するか、またはコードを指定してリモートノードを識別するコードが登録できます。登録するコードは、Tmax環境ファイルのSERVICEセクションに登録されているサービス名である必要があります
port	TmaxのSVCGWと接続する際に使用するサーバーのアドレスとポートを指定します。SVCGWが実行されているサーバーのアドレスとSVCGW CLOPTに、[-P]オプションで登録したポートのうち1つを指定します(例:"1.1.1.1:8090")

- 戻り値

戻り値	説明
0	関数呼び出しに成功した場合です
-1	関数呼び出しに失敗した場合です

3.3. ComDetach

TmaxのSVCGWとの登録を解除するために使用する関数です。登録のみ解除して実際の接続は切断しません。この関数を使用すると、TmaxのSVCGWとデータを送受信することができません。

- プロトタイプ

```
int ComDetach(void)
```

- 戻り値

戻り値	説明
0	関数呼び出しに成功した場合です
-1	関数呼び出しに失敗した場合です

3.4. ComClose

TmaxのSVCGWとの登録を解除し、接続を切断する関数です。TmaxのSVCGWとの通信を切断する場合は、この関数を呼び出します。

- プロトタイプ

```
int ComClose(void)
```

- 戻り値

戻り値	説明
0	関数呼び出しに成功した場合です
-1	関数呼び出しに失敗した場合です

3.5. ComSend

リモートノードからTmaxのSVCGWにデータを転送する関数です。この関数を利用してTmaxにサービスを要求するか、応答を転送することができます。Tmaxのサービス要求に対する応答を転送する場合も、必ずTmaxの新しいサービス名を指定します。

- プロトタイプ

```
int ComSend(char *svc, int msgtype, char *data, long len, long flags)
```

- パラメータ

パラメータ	説明
svc	Tmaxに要求するサービス名を指定します
msgtype	TmaxのSVCGWに転送するデータ型です。各メッセージのタイプについては、commhead.hファイルを参照してください
data	TmaxのSVCGWに転送するデータが保存されているバッファのポインターです
len	転送するデータ長です
flags	サービスを要求する際、SVCGWが当該要求をどのように処理するかを示します。flagsにCOMMDELAYが設定されている場合、SVCGWは当該データを遅延処理として認識してRQに保存します。一方、COMMNOREPLYが設定された場合は、サービスを要求するのみで、応答は受けないという意味で認識されます。現在、flagsに指定できる値は2つです。commhead.hファイルを参照してください

- 戻り値

戻り値	説明
0	関数呼び出しに成功した場合です
-1	関数呼び出しに失敗した場合です

3.6. ComRecv

TmaxのSVCGWからデータを受信する関数です。データはサービス要求または応答データです。

- プロトタイプ

```
int ComRecv(char *svc, int *msgtype, char *data, long *len, long flags)
```

- パラメータ

パラメータ	説明
svc	Tmaxに要求したサービス名が保存される項目です
msgtype	TmaxのSVCGWが転送するデータ型を示します。各メッセージのタイプについては、commhead.hファイルを参照してください
data	TmaxのSVCGWから受信したデータが保存されるバッファのポインターです。このバッファは受信データの中で、最も大きいデータのバッファ・サイズを有する必要があります
len	受信したデータ長です
flags	使用しません

- 戻り値

戻り値	説明
0	関数呼び出しに成功した場合です
-1	関数呼び出しに失敗した場合です

第4章 例

本章では、SVCGWのサービス・タイプ別の例について説明します。

4.1. NOREPLYサービス呼び出し

TmaxのサービスまたはクライアントからSVCGWにtpacallでサービスを要求し、その結果を問わず処理する例です。

プログラムの構成は下記のとおりです。

区分	ファイル名
環境ファイル	tmax.m、tcpsvcgw.cfg
リモートノード	remote.c

4.1.1. 環境ファイル

< tmax.m >

```
*DOMAIN
res      SHMKEY = 88000,
         MINCLH = 1,
         MAXCLH = 1,
         TPORTNO = 8888

*NODE
node1    TMAXDIR="/home/tmax",
         APPDIR="/home/tmax/appbin"

*SVRGROUP
svg1     NODENAME = node1

*SERVER
tcpsvcgw  SVGNAME = svg1, MIN=1, MAX=1,
          CPC=10, SVRTYPE=CUSTOM_GATEWAY,
          CLOPT="-- -P 8090 -F /home/tmax/config/tcpsvcgw.cfg -N 20"

svr      SVGNAME = svg1, MIN=1, MAX=1
```

```
*SERVICE
TCPSVC1      SVRNAME = tcpsvcgw
TOUPPER      SVRNAME = svr
```

< tcpsvcgw.cfg >

```
#####
#                TCP/IP Service Gateway Config                #
#####
# gwno  ipaddr  Tmaxsvc
# 0      1.1.1.1  SVC
#####
# line start with "#" is comment line
# gwno must start at 0 and be increased by 1
# ipaddr: remote node ipaddr
# svc:   tmax service name
#####
0        1.1.1.1   TCPSVC1
```

4.1.2. リモートノード

< toupper.c >

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <usrinc/atmi.h>

main(int argc, char *argv[])
{
    char    *sndbuf, *rcvbuf;
    long    rcvlen;

    if (argc != 2) {
        printf("Usage: toupper string\n");
        exit(1);
    }

    if (tpstart((TPSTART_T *)NULL) == -1){
        printf("Tpstart failed\n");
        exit(1);
    }

    if ((sndbuf = (char *)tpalloc("STRING", NULL, 0)) == NULL) {
```

```

        printf("Sndbuf alloc failed !\n");
        tpend();
        exit(1);
    }

    if ((rcvbuf = (char *)tpalloc("STRING", NULL, 0)) == NULL) {
        printf("Rcvbuf alloc failed !\n");
        tpfree((char *)sndbuf);
        tpend();
        exit(1);
    }

    strcpy(sndbuf, argv[1]);

    if(tpcall("TOUPPER", sndbuf, 0, &rcvbuf, &rcvlen, 0)==-1){
        printf("Can't send request to service TOUPPER\n");
        tpfree((char *)sndbuf);
        tpfree((char *)rcvbuf);
        tpend();
        exit(1);
    }
    printf("Sndbuf data: %s\n", sndbuf);
    printf("Rcvbuf data: %s\n", rcvbuf);

    tpfree((char *)sndbuf);
    tpfree((char *)rcvbuf);
    tpend();
}

```

< SVR.C >

```

#include <stdio.h>
#include <usrinc/atmi.h>

TOUPPER(TPSVCINFO *msg)
{
    char    *sndbuf;
    long    sndlen;

    if ((sndbuf = (char *)tpalloc("STRING", NULL, 0)) == NULL) {
        printf("sndbuf alloc failed !\n");
        tpreturn(TPFAIL, -1, NULL, 0, 0);
    }

    strcpy(sndbuf, msg->data);
    sndlen = msg->len;
}

```

```

/* リモートTCPSVC1サービス呼び出し */
if(tpacall("TCPSVC1", sndbuf, sndlen, TPNOREPLY)==-1){
    printf("Can't send request to service TCPSVC1\n");
    tpfree((char *)sndbuf);
    tpreturn(TPFAIL, -1, NULL, 0, 0 );
}

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

```

< remote.c >

```

#include <stdio.h>
#include <winsock2.h>
#include <usrinc/commhead.h>
#define SVR_PORT    "1.1.1.1:8090"
int main(int argc, char *argv[])
{
    int    n, msgtype;
    long   ilen, olen;
    char   buffer[1024];
    char   svcname[24];

    /* ゲートウェイにリモートノードを登録します。 */
    printf("Remote Service Start...\n\n");
    n = ComOpen(TCPIP, NULL, SVR_PORT );
    if (n < 0) {
        perror("Gateway register error:");
        return -1;
    }

    while(1) {
        memset(svcname, 0x00, sizeof(svcname));
        memset(buffer, 0x00, sizeof(buffer));

        /* データを受信します。 */
        n = ComRecv(svcname, &msgtype, buffer, &olen, COMMNOFLAG);
        if (n < 0) {
            perror("Data Receive error:");
            ComClose();
            return -1;
        }
        printf("REMOTE RECV : svcname = [%s]\n", svcname);
        printf("REMOTE RECV : len = %d\n", olen);
        printf("REMOTE RECV : data= [%s]\n\n", buffer);
    }
}

```



```

/* ゲートウェイとの接続を解除します。 */
ComClose( );
}

```

4.2. REPLYサービス呼び出し

TmaxのサービスまたはクライアントからSVCGWにtpacallでサービスを要求し、結果を問わず処理します。リモートノードで、サービス要求に対する処理結果をTmaxの別のサービスで処理できるようにTOLOWERを呼び出す例です。

プログラムの構成は下記のとおりです。

区分	ファイル名
環境ファイル	tmax.m、tcpsvcgw.cfg
リモートノード	remote.c

4.2.1. 環境ファイル

< tmax.m >

```

*DOMAIN
Res      SHMKEY = 88000,
         MINCLH = 1,
         MAXCLH = 1,
         TPORTNO = 8888

*NODE
node1    TMAXDIR="/home/tmax",
         APPDIR="/home/tmax/appbin"

*SVRGROUP
svg1     NODENAME = node1

*SERVER
tcpsvcgw  SVGNAME = svg1,
         MIN=1,
         MAX=1,
         CPC=10,
         SVRTYPE=CUSTOM_GATEWAY,
         CLOPT="-- -P 8090 -F /home/tmax/config/tcpsvcgw.cfg -N 20"

svr      SVGNAME = svg1,

```

```

        MIN=1,
        MAX=1,

*SERVICE
TCPSVC1      SVRNAME = tcpsvcgw
TOUPPER      SVRNAME = svr
TOLOWER      SVRNAME = svr

```

< tcpsvcgw.cfg >

```

#####
#                TCP/IP Service Gateway Config                #
#####
# gwno  ipaddr  Tmaxsvc
# 0      1.1.1.1  SVC
#####
# line start with "#" is comment line
# gwno must start at 0 and be increased by 1
# ipaddr: remote node ipaddr
# svc:  tmax service name
#####
0        1.1.1.1    TCPSVC1

```

4.2.2. リモートノード

< toupper.c >

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <usrinc/atmi.h>

main(int argc, char *argv[])
{
    char    *sndbuf, *rcvbuf;
    long    rcvlen;

    if (argc != 2) {
        printf("Usage: toupper string\n");
        exit(1);
    }

    if (tpstart((TPSTART_T *)NULL) == -1){
        printf("Tpstart failed\n");
    }
}

```

```

        exit(1);
    }

    if ((sndbuf = (char *)tpalloc("STRING", NULL, 0)) == NULL) {
        printf("Sendbuf alloc failed !\n");
        tpend();
        exit(1);
    }

    if ((rcvbuf = (char *)tpalloc("STRING", NULL, 0)) == NULL) {
        printf("Recvbuf alloc failed !\n");
        tpfree((char *)sndbuf);
        tpend();
        exit(1);
    }

    strcpy(sndbuf, argv[1]);

    if(tpcall("TOUPPER", sndbuf, 0, &rcvbuf, &rcvlen, 0)==-1){
        printf("Can't send request to service TOUPPER\n");
        tpfree((char *)sndbuf);
        tpfree((char *)rcvbuf);
        tpend();
        exit(1);
    }

    printf("Sendbuf data: %s\n", sndbuf);
    printf("Recvbuf data: %s\n", rcvbuf);

    tpfree((char *)sndbuf);
    tpfree((char *)rcvbuf);
    tpend();
}

```

< SVR.C >

```

#include <stdio.h>
#include <usrinc/atmi.h>

TOUPPER(TPSVCINFO *msg)
{
    char    *sndbuf;
    long    sndlen;

    if ((sndbuf = (char *)tpalloc("STRING", NULL, 0)) == NULL) {
        printf("sendbuf alloc failed !\n");
        tpreturn(TPFAIL, -1, NULL, 0, 0);
    }
}

```

```

    }

    strcpy(sndbuf, msg->data);
    sndlen = msg->len;

    if(tpacall("TCPSVC1", sndbuf, sndlen, TPNOREPLY)==-1){
        printf("Can't send request to service TCPSVC1\n");
        tpfree((char *)sndbuf);
        tpreturn(TPFAIL, -1, NULL, 0, 0 );
    }

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

TOLOWER(TPSVCINFO *msg)
{
    int      i;

    printf("TOLOWER service is started!\n");
    printf("INPUT : data=%s\n", msg->data);

    for (i = 0; i < msg->len; i++)
        msg->data[i] = tolower(msg->data[i]);

    printf("OUTPUT: data=%s\n\n", msg->data);

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

```

< remote.c >

```

#include <stdio.h>
#include <winsock2.h>
#include <usrinc/commhead.h>

#define SVR_PORT    "1.1.1.1:8090"

int main(int argc, char *argv[])
{
    int    n, msgtype;
    long   ilen, olen;
    char   buffer[1024];
    char   svcname[24];

    /* ゲートウェイにリモートノードを登録します。 */
    printf("Remote Service Start...\n\n");

```

```

n = ComOpen(TCPIP, "TCPSVC1", SVR_PORT );
if (n < 0) {
    perror("Gateway register error:");
    return -1;
}

while(1) {
    memset(svcname, 0x00, sizeof(svcname));
    memset(buffer, 0x00, sizeof(buffer));

    /* データを受信します。 */
    n = ComRecv(svcname, &msgtype, buffer, &olen, COMMNOFLAG);
    if (n < 0) {
        perror("Data Receive error:");
        ComClose();
        return -1;
    }
    printf("REMOTE RECV : svcname = [%s]\n", svcname);
    printf("REMOTE RECV : len = %d\n", olen);
    printf("REMOTE RECV : data= [%s]\n\n", buffer);

    /* 応答データを転送します。 */
    strcpy(buffer, "I'M REMOTE SERVICE.");
    ilen = strlen(buffer) + 1;
    n = ComSend("TOLOWER", COMMDATA, buffer, ilen, COMMNOREPLY);
    if (n < 0) {
        perror("Data send error:");
        ComClose();
        return -1;
    }
}

/* ゲートウェイとの接続を解除します。 */
ComClose( );
}

```

4.3. リモートノード同期型呼び出し

Tmaxのブート時にSVCGWが起動されており、リモートノードの要求が受信されたら、ユーザーが指定したサービス呼び出した後、再びリモートノードに処理結果を転送する例です。

プログラムの構成は下記のとおりです。

区分	ファイル名
環境ファイル	tmax.m、tcpsvcgw.cfg

区分	ファイル名
リモートノード	remote.c

4.3.1. 環境ファイル

< tmax.m >

```
*DOMAIN
Res      SHMKEY = 88000,
         MINCLH = 1,
         MAXCLH = 1,
         TPORTNO = 8888

*NODE
node1    TMAXDIR="/home/tmax",
         APPDIR="/home/tmax/appbin"

*SVRGROUP
svg1     NODENAME = node1

*SERVER
tcpsvcgw SVGNAME = svg1,
         MIN=1,
         MAX=1,
         CPC=10,
         SVRTYPE=CUSTOM_GATEWAY,
         CLOPT="-- -P 8090 -F /home/tmax/config/tcpsvcgw.cfg -N 20"

svr      SVGNAME = svg1, MIN=1, MAX=1

*SERVICE
TCPSVC1  SVRNAME = tcpsvcgw
TOUPPER  SVRNAME = svr
```

< tcpsvcgw.cfg >

```
#####
#          TCP/IP Service Gateway Config          #
#####
# gwno  ipaddr  Tmaxsvc
# 0      1.1.1.1  SVC
#####
# line start with "#" is comment line
# gwno must start at 0 and be increased by 1
```

```
# ipaddr: remote node ipaddr
# svc:  tmax service name
#####
0      1.1.1.1    TCPSVC1
```

4.3.2. リモートノード

< SVR.C >

```
#include <stdio.h>
#include <usrinc/atmi.h>

TOUPPER(TPSVCINFO *msg)
{
    int    i;
    char    *sndbuf;

    printf("TOUPPER service is started!\n");
    printf("INPUT : data=%s, len = %d\n", msg->data, msg->len);

    for (i = 0; i < msg->len; i++)
        msg->data[i] = toupper(msg->data[i]);

    printf("OUTPUT: data=%s, len = %d\n", msg->data, msg->len);

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

< remote.c >

```
#include <stdio.h>
#include <winsock2.h>
#include <usrinc/commhead.h>

#define SVR_PORT    "1.1.1.1:8090"

int main(int argc, char *argv[])
{
    int    n, msgtype;
    long   ilen, olen;
    char   buffer[1024];
    char   svcname[20];
    char   *ptr;
```

```

/* ゲートウェイにリモートノードを登録します。 */
printf("Remote Service Start...\n\n");
n = ComOpen(TCPIP,"TCPSVC1", SVR_PORT);
if (n < 0) {
    perror("Gateway register error:");
    return -1;
}

ptr = argv[1];
ilen = strlen(argv[1]) + 1;
printf("Send Data = [%s], len = %d\n",ptr,ilen);

n = ComSend("TOUPPER", COMMDATA, ptr, ilen, COMMNOFLAG);
if (n < 0) {
    perror("Data send error:");
    ComClose();
    return -1;
}

memset(svcname, 0x00, sizeof(svcname));
memset(buffer, 0x00, sizeof(buffer));

/* 応答データを受信します。 */
n = ComRecv(svcname, &msgtype, buffer, &olen, COMMNOFLAG);
if (n < 0) {
    perror("Data Receive error:");
    ComClose();
    return -1;
}

printf("REMOTE RECV : svcname = [%s]\n", svcname);
printf("REMOTE RECV : len = %d\n", olen);
printf("REMOTE RECV : data= [%s]\n\n", buffer);

/* ゲートウェイとの接続を解除します。 */
ComClose( );
}

```

4.4. ACK/NAKの通信方式

Tmaxのブート時にSVCGWが起動されており、リモートノードからの要求が受信されたら、ユーザーが指定したサービス呼び出した後、再びリモートノードに処理結果を転送します。このとき、送受信データに対してACK/NAKを処理する例です。

プログラムの構成は下記のとおりです。

区分	ファイル名
環境ファイル	tmax.m、tcpsvcgw.cfg
リモートノード	remote.c

4.4.1. 環境ファイル

< tmax.m >

```
*DOMAIN
res      SHMKEY = 88000,
         MINCLH = 1,
         MAXCLH = 1,
         TPORTNO = 8888

*NODE
node1     TMAXDIR="/home/tmax",
         APPDIR="/home/tmax/appbin"

*SVRGROUP
svg1      NODENAME = node1

*SERVER
tcpsvcgw  SVGNAME = svg1,
         MIN=1,
         MAX=1,
         CPC=10,
         SVRTYPE=CUSTOM_GATEWAY,
         CLOPT="-- -P 8090 -F /home/tmax/config/tcpsvcgw.cfg -N 20 -A"

svr       SVGNAME = svg1, MIN=1, MAX=1

*SERVICE
COM1      SVRNAME = tcpsvcgw
TOUPPER   SVRNAME = svr
```

< tcpsvcgw.cfg >

```
#####
#          TCP/IP Service Gateway Config          #
#####
# gwno  ipaddr  Tmaxsvc
# 0     1.1.1.1  SVC
```

```
#####
# line start with "#" is comment line
# gwno must start at 0 and be increased by 1
# ipaddr: remote node ipaddr
# svc: tmax service name
#####
0      1.1.1.1    TCPSVC1
```

4.4.2. リモートノード

< SVR.C >

```
#include <stdio.h>
#include <usrinc/atmi.h>

TOUPPER(TPSVCINFO *msg)
{
    int    i;
    char    *sdbuf;

    printf("TOUPPER service is started!\n");
    printf("INPUT : data=%s, len = %d\n", msg->data, msg->len);

    for (i = 0; i < msg->len; i++)
        msg->data[i] = toupper(msg->data[i]);

    printf("OUTPUT: data=%s, len = %d\n", msg->data, msg->len);

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

< remote.c >

```
#include <stdio.h>
#include <winsock2.h>
#include <usrinc/commhead.h>

#define SVR_PORT    "1.1.1.1:8090"

int main(int argc, char *argv[])
{

    int    n, msgtype;
    long   ilen, olen;
```

```

char  buffer[1024];
char  svcname[20];
char  *ptr;

/* ゲートウェイにリモートノードを登録します。 */
printf("Remote Service Start...\n\n");
n = ComOpen(TCPIP,"TCPSVC1", SVR_PORT);
if (n < 0) {
    perror("Gateway register error:");
    return -1;
}

while(1) {
    ptr = argv[1];
    ilen = strlen(argv[1]) + 1;
    printf("Send Data = [%s], len = %d\n",ptr,ilen);

    n = ComSend("TOUPPER", COMMDATA, ptr, ilen, COMMNOFLAG);
    if (n < 0) {
        perror("Data send error:");
        ComClose();
        return -1;
    }

    memset(svcname, 0x00, sizeof(svcname));
    memset(buffer, 0x00, sizeof(buffer));

    /* ACK/NAK受信 */
    n = ComRecv(svcname, &msgtype, buffer, &olen, COMMNOFLAG);
    if (n < 0) {
        perror("Data Receive error:");
        ComClose();
        return -1;
    }
    if (msgtype == COMMACK)
        break;
}
printf("REMOTE RECV : msgtype = %d\n\n",msgtype);

memset(svcname, 0x00, sizeof(svcname));
memset(buffer, 0x00, sizeof(buffer));

/* 応答データを受信します。 */
n = ComRecv(svcname, &msgtype, buffer, &olen, COMMNOFLAG);
if (n < 0) {
    perror("Data Receive error:");
    ComClose();
}

```

```

        return -1;
    }

    /* ACK/NAK送信 */
    n = ComSend("TOUPPER", COMMACK, NULL, 0, COMMNOFLAG);
    if (n < 0) {
        perror("Data send error:");
        ComClose();
        return -1;
    }
    printf("REMOTE RECV : svcname = [%s]\n", svcname);
    printf("REMOTE RECV : len = %d\n", olen);
    printf("REMOTE RECV : data= [%s]\n\n", buffer);

    /* ゲートウェイとの接続を解除します。 */
    ComClose( );
}

```

付録 A. ヘッダー・ファイル

A.1. COMMHEADファイル

```
#ifndef _COMM_H_
#define _COMM_H_

#ifndef _WIN32
#define __cdecl
#endif

#ifndef _LOBYTE
#if (defined(_SOCK1) || defined(_SOCK11))
#define _LOBYTE 1
#define _HIBYTE 1
#else
#define _LOBYTE 2
#define _HIBYTE 0
#endif
#endif

/* type */
#define TCPIP 1
#define SERIAL 2

/* flags */
#define COMMNOFLAG 0x00
#define COMMDELAY 0x01
#define COMMNOREPLY 0x10

/* msgtype */
#define COMMDATA 1
#define COMMACK 2
#define COMMNAK 3
#define COMMREGISTER 4
#define COMMUNREGISTER 5

define COMM_MSG_SIZE 1024
#define COMM_DATAP(cp) ((char *)((struct comm_header_t *)cp+1))
#define COMM_HEADER_SIZE sizeof(struct comm_header_t)
#define COMM_LEN_SIZE 4
#define COMM_UID_SIZE 5
```

```

#define COMM_SVC_SIZE          16
#define COMM_ERRCODE_SIZE     4

struct comm_header_t {
    char  len[COMM_LEN_SIZE]; /* data length: データ長 */
    char  msgtype;             /* Message type */
    char  uid[COMM_UID_SIZE]; /* Unique Value: ACK,NAKで使用*/
    char  svc[COMM_SVC_SIZE]; /* Destination Id. */
    char  errcode[COMM_ERRCODE_SIZE]; /* 未使用:次回使用する予定 */
    char  flags;               /* 上記で指定したflagのうち1つ */
    unsigned char crc;         /* Check Byte */
};

typedef struct comm_header_t  COMM_HEADER;

#ifdef __cplusplus
extern "C" {
#endif

/* ----- function prototypes ----- */
int __cdecl ComOpen(int type, char *id, char *port);
int __cdecl ComDetach(void);
int __cdecl ComClose(void);
int __cdecl ComSend(char *svc, int msgtype, char *data, long len, long flags);
int __cdecl ComRecv(char *svc, int *msgtype, char *data, long *len, long flags);

#ifdef __cplusplus
}
#endif
#endif /* _COMM_H_ */

```

索引

A

ACK/NAKの通信方式の例, 30
ACK/NAK応答のSVCGW, 11

C

COMライブラリー, 15
 ComClose(), 16
 ComDetach(), 16
 ComOpen(), 15
 ComRecv(), 18
 ComSend(), 17

N

NOREPLYサービス呼び出しの例, 19

R

REPLYサービス呼び出しの例, 23

S

SVCGW, 1
SVCGWの動作構造, 1
SVCGWの環境ファイル, 12
 アドレス情報の環境ファイル, 12

T

TCP/IPサービス・ゲートウェイ, 1
Tmaxからのサービス呼び出し
 NOREPLYサービス呼び出し, 3
 REPLYサービス呼び出し, 4
Tmaxからのサービス要求, 3
Tmax環境ファイル, 9
Tmax環境構成
 ACK/NAK応答のSVCGW, 11
 CLOPT項目(SVCGWオプション), 10
 遅延処理のSVCGW環境ファイル, 12

Tmax環境構成(CLOPT項目)

[-A], 11
[-F], 10
[-N], 11
[-P], 11
[-q], 11

あ

アウトバウンド・サービス, 2
アドレス情報の環境ファイル, 12
インバウンド・サービス, 2

た

遅延処理のSVCGW環境ファイル, 12

ら

例

 ACK/NAKの通信方式, 30
 NOREPLYサービス呼び出し, 19
 REPLYサービス呼び出し, 23
 リモートノード同期型呼び出し, 27
リモートノードからのサービス要求, 5
 リモート同期型呼び出し方式, 6
 リモート非同期型呼び出し方式, 7
リモートノード同期型呼び出しの例, 27

