

Tmax Host-linkガイド (SNA LU 0, SNA LU 6.2)

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 Host-linkガイド (SNA LU 0, SNA LU 6.2)

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. Host-linkの構造	2
第2章 セッション管理	5
2.1. セッション	5
2.2. インバウンド・セッション	5
2.3. アウトバウンド・セッション	6
2.4. セッションの問題解決	7
2.4.1. セッション	7
2.4.2. リンク・セッション	7
2.4.3. セッション・プール	8
第3章 環境設定	9
3.1. Tmax環境設定ファイル	9
3.1.1. SERVERセクション	9
3.1.2. GATEWAYセクション	10
3.1.3. SERVICEセクション	12
3.2. Host-link環境設定ファイル	12
3.2.1. HOSTLINKセクション	13
3.2.2. SERVERセクション	13
3.2.3. SERVICEセクション	20
3.2.4. LOGGINGセクション	25
3.2.5. CONVERSIONセクション	27
3.2.6. DUMMYDATAセクション	31
3.2.7. USERHEADERセクション	32
3.2.8. LUINFOセクション	37
3.3. サービス別の環境設定	39
3.3.1. インバウンド・サービス	40
3.3.2. アウトバウンド・サービス	45
3.3.3. BIDおよびROP	46
3.4. 環境設定ファイルのコンパイル	48
3.4.1. hkcf1	48
第4章 状態モニタリングAPIと構造体	49
4.1. API	49
4.1.1. tpgethlinksvr	49
4.1.2. tpgethlinkproc	50
4.1.3. tpgethlinkluinfo	50
4.1.4. tpgethlinkssinfo	51
4.2. 構造体	51

4.2.1. HLPROCINFO	52
4.2.2. HLLUINFO	53
4.2.3. HLSESSINFO	53
4.3. エラーコード	54
付録 A. 環境設定の例	57
A.1. SNA LU 0	57
A.2. SNA LU 6.2	59
付録 B. ユーザー関数	63
B.1. SNA LU 0	63
B.2. SNA LU 6.2	64
索引	69

図目次

[図 1.1]	Host-linkの概要	1
[図 1.2]	Host-link内部の構造	3
[図 3.1]	tpcallサービス	40
[図 3.2]	tpacall - Tmaxシステムに応答するサービス	41
[図 3.3]	tpacall - Tmaxシステムに応答しないサービス	41
[図 3.4]	tpforward - TPNOREPLYを設定した場合	43
[図 3.5]	tpforward - TPNOREPLYを設定していない場合	44
[図 3.6]	アウトバウンド・サービスの処理手順	45
[図 3.7]	BIDおよびROPサービスの処理手順	47

このガイドについて

対象読者

本書は、Tmax[®](以下、Tmax)とホストを接続するHost-linkの開発者を対象としています。

前提知識

本書は、Tmaxシステムについての全般的な理解と、Tmaxシステムが提供する各種機能および特性を習得するための基本ガイドです。

本書を理解するためには、以下の事項について熟知している必要があります。

- ミドルウェアおよびUNIXシステムについて
- Tmaxの基本概念について
- Java、Cプログラミングについて
- SNA(System Network Architecture)について

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。実務での具体的な使用方法や管理および運用に関する事項は、各製品のガイドを参照してください。

参考

Tmaxシステムの開発に関する基本的な内容は『Tmax 運用ガイド』や『Tmax アプリケーション開発ガイド』を参照してください。Tmaxで提供するコマンドとC APIについての説明は『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4つの章と付録から構成されています。

各章の主な内容は以下のとおりです。

- 第1章: 紹介

Host-linkの概念、構造、機能について説明します。

- 第2章: セッション管理

インバウンド・セッションとアウトバウンド・セッションの管理について説明します。

- 第3章: 環境設定

TmaxとHost-linkの環境設定ファイルについて説明します。

- 第4章: 状態モニタリングAPIと構造体

Host-linkの状態をモニタリングするAPIと構造体について説明します。

- 付録A: 環境設定の例

SNA LU 0とSNA LU 6.2の例を説明します。

- 付録B: ユーザー関数

SNA LU 0とSNA LU 6.2のユーザー関数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、ディレクトリー
<Ctrl>+C	CtrlキーとCキーを同時に押す
[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
<div>参考</div>	参照/注意事項
[図 1.1]	図の名称
[表 1.1]	表の名称
<div>AaBbCc123</div>	コマンド、コマンド実行結果の画面出力、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

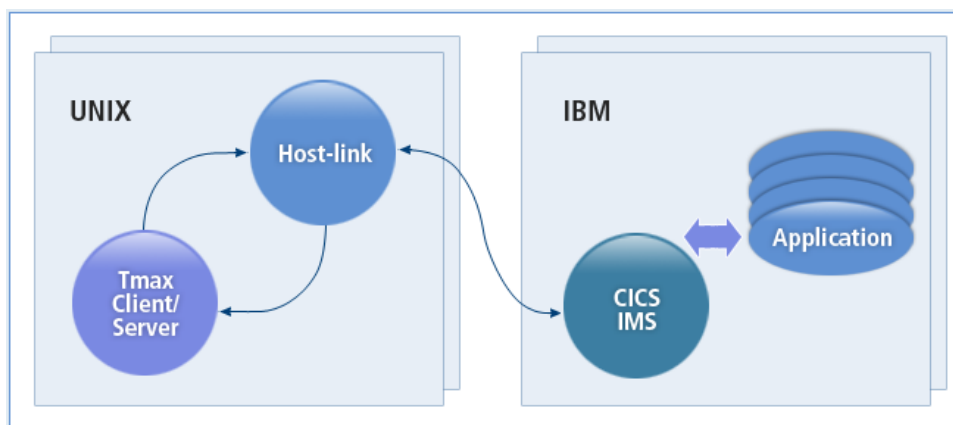
本章では、Host-linkの概念と構造について説明します。

1.1. 概要

Host-linkは、オープン環境でもIBMホストのような基幹系システムに存在するアプリケーションサービスに同様に接続するようにするゲートウェイです。

Host-linkを使用する場合、アプリケーションのプログラム・コードを修正する必要がなく、基幹系システムのリソースをそのまま活用できます。また、標準通信プロトコルのSNA、TCP/IPを使用して、基幹系システムと最適の状態で連動できます。基幹系システムと同時にオープン環境の安価で多様なWebリソースを活用できます。

[図 1.1] Host-linkの概要



Host-linkは基幹系システムとオープン環境システム間の通信プロトコルによって、SNA(System Network Architecture)とTCP/IPに分類されます。

- SNA
 - CICSのDPL(Distributed Program Link)とDTP(Distributed Transaction Processing)、IMSのAPPCとCPI-Cインターフェースを提供します。要求/応答モデル、会話型モデルをサポートします。
 - SNAの場合、Host-linkを使用するために、サーバーの要求に合わせて以下のSNAゲートウェイをインストールします。

プロトコル	説明
Solaris	SunLinkTM

プロトコル	説明
	<ul style="list-style-type: none"> – HP-UX : SNAPPlus2 – AIX : IBM Communications Server

- SNAはタイプによってLU 0とLU 6.2に分類されます。LU(Logical Unit)は、IBM SNAで使用するパッケージの種類です。

- SNA LU 0

1つのプロセスが多様な業務を処理します。

- SNA LU 6.2

各プロセスは内部的にスレッドを作成し、サービスを処理します。スレッドはカーネル・レベルのスレッドとして最大50個まで作成が可能で、各スレッドは1つのLUセッションを担当します。50個以上のセッション接続を行う場合は、別途のプロセスを起動して処理します。

オープン環境システムからホストにサービスを要求する場合はインバウンド・サービス、ホストからオープン環境システムにサービスを要求する場合はアウトバウンド・サービスといいます。

SNA LU 6.2では以下の2つの方法でサービスを処理します。

サービス	説明
DPL方式	ホストが提供するミラーを使用してサービスを渡します。インバウンド・サービスでのみサポートします
DTP方式	ホスト・サービスと直接接続します。インバウンド・サービスとアウトバウンド・サービスの両方をサポートし、Host-link環境設定ファイルに設定されている内容に従って選択的に運用することも可能です

- TCP/IP

OS/390のTPモニターであるCICS、IMSと連動します。要求/応答モデル、会話型モデルをサポートします。

Host-linkのカスタム・ゲートウェイの特性によって、Tmaxシステムに接続されて自動的に開始および終了します。したがって、Host-linkの起動および終了は別途実行する必要がなく、Tmaxの起動および終了によって自動的に処理されます。

1.2. Host-linkの構造

Host-linkの構造は内部構造とシステム全体に分けられます。

全体構造

巨視的な観点で、Host-linkはTmaxのカスタム・ゲートウェイ方式で管理されます。したがって、Host-linkを実行するためには、Tmax環境設定ファイルにカスタム・ゲートウェイ・タイプで登録します。また、Host-linkはオープン環境システムに存在するSNAゲートウェイと接続し、オープン・システムの要求をホストに渡します。

以下は、カスタム・ゲートウェイ方式の長所です。

- 多重チャネルの使用

Host-linkは、Tmaxシステムと多重チャネルを接続することで同時に複数の業務を処理できます。Host-linkとオープン・システムのSNAゲートウェイを多重セッションで接続することで可能です。

- セッション数の制御

Host-linkとSNAゲートウェイのセッション数より多くのサービスが要求された場合、超過サービスを送信プロセス内のキューに保存します。使用可能なセッションが発生した場合、FIFO方式でサービスを要求します。

- 一般サーバープロセス形式と同一管理

一般プロセスはプロセス単位で起動と終了が可能です。Host-linkも同様にtmddown/tmbootコマンドを使用して起動と終了が可能です。また、システム管理ツールであるtmdadminを使用して処理件数および平均応答時間などを確認できます。

- サービス形式で呼び出し

Host-linkは環境設定ファイルに1つのサービスを持つように規定されています。したがって、オープン環境システムでHost-linkを使用するためには、Tmaxシステム内で他の一般サービスの呼び出し方式と同様にサービス名で呼び出します。

内部構造

Host-link内部は、Tmaxサービスのモジュール、セッション・マネージャー・モジュール、ホスト・サービス・モジュールで構成されています。

[図 1.2] Host-link内部の構造



- Tmaxサービス・モジュール

Tmaxサービス・モジュールは、Tmaxシステムとホスト・システム間のセッションを接続し、監視します。また、Tmaxシステムで要求したサービスをホスト・サービス・モジュールに渡したり、ホスト・サービス・モジュールで要求したサービスをTmaxシステムに要求したりする役割をします。

- セッション・マネージャー・モジュール

セッション・マネージャーは、Tmaxシステムとホスト・システムをSNA LU 0またはSNA LU 6.2方式で接続し、これを監視して2つのシステム間の円滑なサービスを実行するようにサポートします。

- ホスト・サービス・モジュール

ホスト・サービス・モジュールは、ホストに存在するプロセスでTmaxサービスを要求した場合、Tmaxサービス・モジュールに該当業務を渡し、Tmaxシステムで要求したサービスをホストに渡すモジュールです。

第2章 セッション管理

本章では、インバウンド・セッションとアウトバウンド・セッションの管理について説明します。

2.1. セッション

Host-linkシステムは、複数のプロセスが存在し、セッションを管理および維持します。Host-linkシステムに存在するプロセス数は、Tmaxの環境設定ファイルのSERVERセクションのMIN、MAX項目で調節できます。プロセスは1つのリンク・セッションで提供する最大セッション数の分だけサービスを処理します。セッションが不足する場合、自動的に処理が可能な他のHost-linkプロセスにサービスを移して処理します。

Host-linkシステムで管理するセッションは、インバウンド、アウトバウンドに分類されます。インとアウトはホストを基準にします。

2.2. インバウンド・セッション

インバウンド・セッションは、オープン環境システムでホスト・システムにサービスを要求するセッションで、プロセス間のセッション共有の対象です。

セッションを使用する方式に従って、Pool方式とDedicate方式に分けられます。

- Pool方式

Pool方式は、現在使用可能なセッションでサービスを要求する方式です。TmaxシステムのロードバランスによってHost-linkプロセスでサービスが要求された場合、Host-linkプロセスは自身が保有している使用可能なセッションを検索してホストにサービスを要求します。使用可能なセッションがない場合、処理が可能な他のHost-linkプロセスにサービスを渡します。

Pool方式でセッションを使用する場合、セッションを業務別に分割し、該当業務で使用するセッションを指定してサービスを要求できます。この方式を使用するには、Host-link環境設定ファイルのユーザー・ヘッダーを指定し、ヘッダーのセッション・タイプを指定してサービスを要求します。

- Dedicate方式

Dedicate方式は、セッションを特定値に指定して、指定した値でセッションを検索してサービスを要求する方式です。Dedicate方式は端末別にセッションを維持する場合に使用します。端末別に一意の番号を付与し、この番号に該当するセッションでサービスを要求します。既存のホスト・システムでは端末別にセッションを維持し、端末で発生した入出金についての時制を管理します。

SNA LU 0

SNA LU 0はPool方式とDedicate方式を両方サポートします。

SNA LU 6.2

SNA LU 6.2はPool方式のみをサポートし、以下の2つの方式を使用できます。

区分	説明
DPL方式	DPLはTmaxの会話型通信モードで使用される方式で、ホストで提供するミラーを使用してサービスを渡します。DPLはインバウンド・サービスでのみサポートします
DTP方式	DTPは、Tmaxの同期型、非同期型通信モードで使用される方式で、ホスト・サービスと直接的に接続する方式です。インバウンド、アウトバウンド・サービスの両方をサポートし、Host-link環境設定ファイルに指定されている内容に従って、選択的な運用も可能です

参考

通信モードについての詳細は、Tmaxガイドの『Tmax 運用ガイド』を参照してください。

2.3. アウトバウンド・セッション

アウトバウンド・セッションは、ホスト・システムでオープン環境システムにサービスを要求するセッションで、プロセス間のセッション共有の対象から除外されます。アウトバウンド・セッションはホストで使用可能なセッションであるため、オープン環境システムのCUSTOM_GATEWAYであるHost-linkはプロセス間でセッションを共有できません。

SNA LU 0

SNA LU 0のアウトバウンド・セッションは、一般セッション、BIDセッション、ROPセッションに分類されます。

区分	説明
一般セッション	Tmaxシステムで提供するサービスを要求するセッションです。このセッションでサービスが要求された場合、Host-linkプロセスはHost-link環境設定ファイルに登録されている方式でTmaxシステムにサービスを要求します
BIDセッション	業務的に緊急のメッセージを特定地域や特定の人に渡すために使用するセッションです。このセッションでサービスが要求された場合、Host-linkプロセスは一般セッションのように環境設定ファイルに登録されている方式でTmaxシステムにサービスを要求します

区分	説明
ROPセッション	BIDセッションで渡すことができない多くのデータを渡し、出力するセッションです。文書を他の地域に渡す場合、以前は郵便を利用していましたが、このセッションを使用して該当地域の特定端末にデータを渡し、即時出力することができます。このセッションも環境設定ファイルに登録されている方式でTmaxシステムにサービスを要求します

SNA LU 6.2

SNA LU 6.2のアウトバウンド・セッションは、Tmaxシステムで提供するサービスを要求するセッションです。このセッションでサービスが要求された場合、Host-linkプロセスは環境設定ファイルに登録されている方式でTmaxシステムにサービスを要求します。

2.4. セッションの問題解決

セッションの問題解決は、Tmaxシステムが提供する問題の解決方法に従います。問題解決方法は次の3つの場合に分類されます。

2.4.1. セッション

Host-linkプロセスは特定セッションが非活性化されたことを検知すると、まずSNAサーバーとセッションを終了し、最初にセッションを接続するプロセスに従ってホストと再度接続を試みます。接続を行う際はタイムアウトで時間を制限するため、接続を繰り返してもシステムに負荷が発生しません。

セッション別の問題解決は以下の順序で行われます。

1. SNAサーバーとセッションを終了します。
2. SNAサーバーにセッションを初期化します。
3. ホストにセッション情報を送信します。
4. ホストからBindを受信した場合は正常にセッションが接続されたことを意味し、そうでない場合は一定時間が経過してから3番の過程から繰り返します。
5. 正常なセッションで登録します。

2.4.2. リンク・セッション

1つのHost-linkプロセスは1つのリンク・セッションのみを管理し、周期的にリンク・セッションを監視します。リンク・ラインが非活性化状態から活性化状態に変化すると、これに属するすべてのセッションをホストに接続

しようします。1つのリンク・セッションには最大256個のセッションを接続できます。リンク・セッションが非活性化されると、該当リンク・セッションに属するすべてのセッションはホストとの接続が終了します。

リンク・セッションの問題解決は以下の順序で行われます。

1. Tmaxシステムとの接続を暫くの間終了します。
2. SNAサーバーとセッションを終了します。
3. リンク・セッションを監視します。
4. リンク・セッションが正常に接続された場合、セッションのうち1つのみ接続を行います。
5. 1つのセッションがホストからBindを受信すると、すべてのセッションをホストと接続します。
6. Tmaxシステムと再度接続し、スケジューリング作業を行います。

2.4.3. セッション・プール

1つのHost-linkプロセスは最大256個のセッションを管理し、複数のHost-linkプロセスを1つのプールという概念で使用できます。この場合、要求したサービスを実行するHost-linkプロセスに使用可能なセッションがない場合、使用可能なセッションを保有する他のHost-linkプロセスにデータを渡します。

Host-linkシステムに属する1つのHost-linkプロセスは最大256個のセッションのみを管理するため、256個以上のサービスが同時に要求された場合、該当Host-linkシステムはサービスを処理できません。このような場合、自身が属するHost-linkシステムで使用可能なセッションを検索し、該当セッションを管理するHost-linkプロセスにデータを渡してサービスが処理されます。

Host-linkシステムは、Tmax環境設定ファイルのSERVERセクションに登録されているサーバーを意味します。1つのHost-linkシステムには複数のHost-linkプロセスを設定でき、プロセスの数はSERVERセクションのMIN項目とMAX項目で設定します。SERVERセクションの設定方法については「[3.1.1. SERVERセクション](#)」を参照してください。

第3章 環境設定

本章では、Host-linkを登録する方法について説明します。

3.1. Tmax環境設定ファイル

Host-linkはTmax環境設定ファイルに別途のセクションを持たず、SERVICEセクションに追加のみすれば使用が可能です。ただし、トランザクション(2PC、Synclevel2)機能を使用するには、SERVERセクションの代わりにGATEWAYセクションを使用します。

現在トランザクション・ゲートウェイはLU 6.2でのみ使用可能で、ゲートウェイ・タイプがSNACICSの場合にのみサポートします。

参考

各セクションの設定項目についての詳細は『Tmax 運用ガイド』を参照してください。

3.1.1. SERVERセクション

以下はSERVERセクションの形式です。実際に登録する内容は、使用例を参照してください。

```
*SERVER
ServerName    SVGNAME = server-group-name ,
               [SVRTYPE = CUSTOM_GATEWAY,]
               [MIN = number,]
               [MAX = number]
....
```

必須項目

- *ServerName* = string
 - Host-linkの共有メモリーを管理するサーバー名で、SERVERセクションに一度だけ設定する必要があります。
 - ユーザーが任意の名前で登録でき、登録後に必ず環境設定ファイルのAPPDIR項目で指定したパスに登録した名前のHost-linkプログラムをコピーします。
- *SVGNAME* = string

- サーバー・グループ名を設定します。

選択項目

- SVRTYPE = CUSTOM_GATEWAY
 - サーバー・タイプを設定します。Host-linkであるため、CUSTOM_GATEWAYに設定します。
- MIN = numeric
 - 範囲: 1 ~ MAX_INT
 - デフォルト値: 1
 - Host-linkプロセスの最小数を設定します。
- MAX = numeric
 - 範囲: 1 ~ MAX_INT
 - デフォルト値: 1
 - Host-linkプロセスの最大数を設定します。

使用例

以下は、SERVERセクションの設定についての例です。

```
*SERVER
hkh          SVGNAME = svg1,
             SVRTYPE = CUSTOM_GATEWAY

snalugw      SVGNAME = svg1,
             SVRTYPE = CUSTOM_GATEWAY,
             MIN  = 2,
             MAX  = 2

...
```

3.1.2. GATEWAYセクション

以下はGATEWAYセクションの形式です。GATEWAYセクションは選択的に必要な場合は関連内容を設定します。実際に登録する内容は使用例を参照してください。

```
*GATEWAY
GatewayName      NODENAME = node-name ,
                  GWTYPE = SNACICS ,
                  [CPC = channel-number]
...

```

必須項目

- *GatewayName* = string
 - ゲートウェイ名で、Host-linkの実行ファイル名を設定します。
 - ユーザーが任意の名前で登録でき、登録後に環境設定ファイルのAPPPDIR項目で指定したパスに登録した名前のHost-linkプログラムを必ずコピーします。
- *NODENAME* = string
 - ノード名を設定します。
- *GWTYPE* = SNACICS
 - ゲートウェイ・タイプを設定します。現在サポートしているSNACICSを設定します。

選択項目

- *CPC* = numeric
 - 範囲 : 1 ~ 128
 - デフォルト値 : 1
 - CPCの数を設定します。CPC(CLH Per Channel)は、Tmaxシステムでノード間のCLHプロセスのチャネル数を意味します。CLHは、サーバーとクライアントを接続するTmaxエンジンです。

使用例

以下は、GATEWAYセクションの設定についての例です。

```
*GATEWAY
snacics_s          GWTYPE = SNACICS,
                   NODENAME = node1,
                   CPC = 10

snacics_r          GWTYPE = SNACICS,

```

```
NODENAME = node1,  
CPC = 10
```

3.1.3. SERVICEセクション

以下はSERVICEセクションの形式です。実際に登録する内容は使用例を参照してください。

```
*SERVICE  
ServiceName SVRNAME = server-process-name  
...
```

必須項目

- *ServiceName* = string
 - サイズ : 63文字以内(ただし、CONVERSIONのMapを使用する場合は8文字以内のstringに設定します。)
 - サーバー・プログラム内の関数名(サービス・ルーチン名)が使用されます。必ずSERVICEセクションで一意の名前である必要があります。マルチ・ドメイン環境ではすべてのドメインのうち一意の名前を推奨します。
- *SVRNAME* = *server-process-name*
 - サービスが属するサーバー名を設定します。

使用例

以下は、SERVICEセクションの設定についての礼です。

```
*SERVICE  
SNALUGW SVRNAME = snalugw  
...
```

3.2. Host-link環境設定ファイル

Host-link環境設定ファイルは、Host-linkシステムを実行するために必要なサービス情報、データ・ロギング、データ変換、セッション・チャネルの使用方式など多様な情報を登録するファイルです。ファイルは8個のセクションから構成されています。HOSTLINKセクション、SERVERセクション、SERVICEセクション、LUINFOセクションは必須であり、それ以外のセクションは必要に応じて選択して設定します。

3.2.1. HOSTLINKセクション

HOSTLINKセクションの形式は以下のとおりです。Host-linkの全体的な環境を設定するセクションです。

```
*HOSTLINK
Host-link Name    TMAX = Y|(N) ,
                  SHMKEY = shared memory segment key
```

必須項目

- Host-link Name = string
 - サイズ : 63文字以内
 - Host-link名を設定します。設定した名前はHOSTLINKセクションで一意である必要があります。
- TMAX = Y|N
 - 以下は各設定値についての説明です。現在は「Y」のみサポートします。

設定値	説明
Y	Tmaxと連動し、Host-linkを運用します
N	Host-linkを独立的に運用します。現在はサポートしていません

- SHMKEY = numeric
 - 範囲 : 32768 ~ 262143
 - Host-linkは内部情報を共有メモリーで管理しますが、共有メモリーのセグメントを表す値です。

3.2.2. SERVERセクション

SERVERセクションの形式は以下のとおりです。Host-linkサーバーの環境を設定するセクションです。

```
* SERVER
ServerName    TMAXSVRNAME = tmaxserver name ,
               TMAXSVRNO = server index ,
               SVRTYPE = {LU0 | LU62S | LU62R | CTG | CICSTCPIP | IMSTCPIP } ,
               [HLINKNAME = Host-Link名 ,]
               [FUNCTION = DPL | DTP ,]
               [HOSTID = host id ,]
               [SESSION_TYPE = IMS_INITSELF|IMS_AUTO|CICS_INITSELF|CICS_AUTO ,]
               [MSGSIZE = 1 ~ MAX_INT ,]
```

```

[TIMEWAIT = WAIT | NOWAIT,]
[LINKDOWN_TIMEOUT = 1 ~ MAX_INT,]
[SESSION_TIMEOUT = 1 ~ MAX_INT,]
[BUFFERING = Y | N,]
[INBOUNDLU = (POOL) | DEDICATE,]
[BIDLU = (POOL) | DEDICATE,]
[LINKNAME = linkname,]
[TRXID = transaction_xid,]
[HOSTADDR = ホストIPアドレスまたはホスト名,]
[HOSTPORT = ホスト・ポート,]
[LISTENPORT = listen port,]
[SESSION = 1 ~ MAX_INT,]
[MAXSVR = 1 ~ MAX_INT,]
[RESEND = Y | N,]
[COMMSIZETYPE = INPUT_EQUAL|INPUT_1024|INPUT_2048|(INPUT_4096)|
                MAPFILE|SPECIFY_SIZE,]
[DELOUTHSIZE = 1 ~ MAX_INT,]
[SVRLIST = サーバー・リスト]

```

必須項目

- *ServerName* = string
 - サイズ：63文字以内
 - サーバー名は1つのHost-linkプロセスを表示する論理名で、SERVERセクションで一意である必要があります。
 - Tmax環境設定ファイルに1つのHost-linkをサーバーに登録後、MIN、MAX項目をそれぞれ2に設定した場合、SERVERセクションに2つのサーバーが設定される必要があります。
- *TMAXSVRNAME* = string
 - サイズ：63文字以内
 - Tmax環境設定ファイルのSERVERセクションで設定したサーバー名と同一に設定します。
- *TMAXSVRNO* = string
 - サイズ：63文字以内
 - Tmax環境設定ファイルのSERVERセクションのMIN、MAX項目値に従ったIndexを設定します。
 - SERVERセクションのMIN、MAXをそれぞれ2に設定した場合、Indexは00から開始し、00、01に設定できます。

- SVRTYPE = {LU0 | LU62S | LU62R | CTG | CICSTCPIP | IMSTCPIP}
 - ホストと通信するプロトコルのタイプを設定します。
 - 現在、**LU0**、**LU62S**、**LU62R**、**CTG**のみサポートします。

選択項目

- HLINKNAME = string
 - サイズ : 63文字以内
 - HOSTLINKセクションのHost-link Nameと同一に設定します。
- FUNCTION = DPL | DTP
 - 関数を呼び出す方式を設定します。
 - SVRTYPE項目がLU62SあるいはLU62Rの場合にのみ設定します。
- HOSTID = string
 - サイズ : 4文字以内
 - セッションを確立するホストのIDを設定します。
 - Host-linkが該当ホストとセッションを接続するためにInitselfデータをホストに送信するのにHOSTIDがInitselfデータに含まれます。
 - SVRTYPE項目をLU0に設定した場合にのみ該当する項目で、SVRTYPEがLU0に設定された場合は設定が必須です。
- SESSION_TYPE = IMS_INITSELF | IMS_AUTO | CICS_INITSELF | CICS_AUTO
 - デフォルト値 : IMS_INITSELF
 - ホストとセッションを接続する際のInitself処理方式を設定します。
 - SVRTYPE項目をLU0に設定した場合にのみ該当する項目です。
- MSGSIZE = numeric
 - デフォルト値 : 1024

– 範囲

区分	説明
LU 0	1~MAX_INT
LU 6.2	1024~MAX_INT (ただし、OpenFrameのCICSを使用する場合は32000に設定します)

● TIMEWAIT = WAIT | NOWAIT

– Tmaxクライアントとサーバーでタイムアウトが発生した際の動作を設定します。

– 以下は各設定値についての説明です。

設定値	説明
WAIT	該当セッションをそのまま使用します
NOWAIT	該当セッションを終了し、再度セッションを開始します

● LINKDOWN_TIMEOUT = numeric

– 範囲：1~MAX_INIT

– Host-linkとホスト間のリンク・ラインが切断された際にリンク・ラインが活性化されているかをチェックする時間の間隔を設定します。

● SESSION_TIMEOUT = numeric

– サイズ：1~MAX_INIT

– セッションが切断された際にセッションが活性化されているかをチェックする時間の間隔を設定します。

● BUFFERING = Y | N

– Pool方式でセッションを使用する際に設定します。

– Host-linkプロセスに使用可能なセッションがない場合の動作を設定します。

– 以下は各設定値についての説明です。

設定値	説明
Y	他のプロセスに要求をせずに、可能なセッションが発生するまでキューに保存します
N	他のプロセスに要求を渡します

- INBOUNDLU = (POOL) | DEDICATE

- Host-linkでセッションを運用できる方式を設定します。
- SVRTYPE項目をLU0に設定した場合にのみ該当する項目です。

- BIDLU = (POOL) | DEDICATE

- BIDデータの処理方式を設定します。Pool方式とDedicate方式の詳細については「[2.2. インバウンド・セッション](#)」を参照してください。
- SVRTYPE項目をLU0に設定した場合にのみ該当する項目です。

- LINKNAME = string

- サイズ : 63文字以内
- ホストと通信する際に物理的に接続する名前を設定します。
名前は以下のコマンドを使用して確認できます。

```
sna -d l
```

- Host-linkは、LINKNAMEが活性化された場合にのみサービスを実行します。活性化状態でない場合、tmadminでHost-linkプロセスを確認するとNOT-READYと現れます。
- SVRTYPE項目をLU0、LU6.2S、LU6.2Rに設定した場合は設定が必須です。

- TRXID = string

- サイズ : 4文字以内
- トランザクションIDを設定します。設定された情報はプログラム名と一緒にホストのサービスを検索するためのIDとして使用されます。

- HOSTADDR = literal

- サイズ : 255文字以内
- ホスト・アドレスを設定します。IPアドレスまたはホスト名を設定できます。
- SVRTYPE項目をCICSTCPIPまたはIMSTCPIP設定した場合にのみ該当する項目で、現在は使用していません。

- HOSTPORT = numeric
 - 範囲：1~MAX_INT
 - ホスト・ポートを設定します。
 - SVRTYPE項目をCICSTCPIPまたはIMSTCPIP設定した場合にのみ該当する項目で、現在は使用していません。
- LISTENPORT = numeric
 - 範囲：1~MAX_INT
 - 待ち受けしているポートを設定します。
 - SVRTYPE項目をCICSTCPIPまたはIMSTCPIP設定した場合にのみ該当する項目で、現在は使用していません。
- SESSION = numeric
 - 範囲：1~MAX_INT
 - SVRTYPE項目をCTGに設定した場合、この値はホストと接続するセッション値を意味します。
 - LU0、LU6.2の場合、該当サーバーLUのセッション数の合計で自動設定されるため、別途設定する必要がありません。
- MAXSVR = numeric
 - 範囲：1~MAX_INT
 - Tmax環境設定ファイルでMIN、MAX項目に設定されている値と同一に設定します。
 - Host-linkプロセスが起動する際に、SESSION項目で設定した数だけのセッションがHost-linkプロセスに接続されます。
 - SVRTYPE項目をCTGに設定した場合にのみ該当する項目です。
- RESEND = Y | N
 - tpacall(NOREPLY)の場合に応答を設定します。
 - 以下は各設定値についての説明です。

設定値	説明
Y	sendtype = AP_SEND_DATA_DEALLOC_FLUSHで応答を転送します
N	sendtype = AP_SEND_DATA_DEALLOC_ABENDで応答を転送します

- COMMSIZETYPE = INPUT_EQUAL | INPUT_1024 | INPUT_2048 | (INPUT_4096) | MAPFILE | SPECIFY_SIZE

- デフォルト値：INPUT_4096
- COMMAREAのサイズを設定します。
- SVRTYPE項目をLU62Sに設定した場合にのみ該当する項目です。
- 以下は各設定値についての説明です。

設定値	説明
INPUT_EQUAL	入力データのサイズと同一に設定します
INPUT_1024	入力データとサイズが同じか、大きい数のうち最も近い1024の倍数に設定されます
INPUT_2048	入力データとサイズが同じか、大きい数のうち最も近い2048の倍数に設定されます
INPUT_4096	入力データとサイズが同じか、大きい数のうち最も近い4096の倍数に設定されます
MAPFILE	MAPファイルの「COMMSIZE」と同一の値が設定されます
SPECIFY_SIZE	「MSGSIZE」と同一の値が設定されます

- DELOUTHSIZE = numeric

- 範囲：1~MAX_INT
- ホストがデータを受信する際に破棄するデータのサイズを設定します。データの先頭から、設定されたサイズの分だけ破棄します。
- SVRTYPE項目をLU0またはLU62Sに設定した場合にのみ該当する項目です。

- SVRLIST = string

- サイズ：256文字以内
- SVRTYPE項目をCTGに設定した場合にのみ該当する項目です。

使用例

以下は、セッションの運用と関連する項目設定についての例です。

- SNA LU 0

INBOUNDLU項目を「POOL」に設定すると、Host-linkは接続されたセッションを通じてラウンドロビン方式でホストにサービスを要求します。BIDLU項目をDEDICATEに設定すると、特定のセッションでBIDデータを受信します。

```
*SERVER
snalugw00      ...
               INBOUNDLU = POOL,
               BIDLU = DEDICATE,
               ...
```

- SNA LU 6.2

FUNCTION項目を「DPL」に設定すると、Tmaxシステムでホストにサービスを要求する際、DPL方式でセッションを使用します。

```
*SERVER
snalusgw00     ...
               FUNCTION = DPL,
               ...
```

3.2.3. SERVICEセクション

SERVICEセクションの形式は以下のとおりです。SERVERのBIDサービス、ROPサービス、OUTサービス、TCLサービスなどを設定するセクションです。

```
*SERVICE
ServerName    [BIDSVCCNAME = BID svc name,]
               [BIDSVCPPOS = 1 ~ MAX_INT,]
               [BIDSVCSIZE = 1 ~ 16,]
               [ROPSVCNAME = ROP svc name,]
               [ROPSVCPOS = 1 ~ MAX_INT,]
               [ROPSVCSIZE = 1 ~ 16,]
               [OUTSVCNAME = outsvc name,]
               [OUTSVCPPOS = 1 ~ MAX_INT,]
               [OUTSVCSIZE = 1 ~ 16,]
               [RLYSVCNAME = relay svc name,]
               [RLYSVCPOS = 1 ~ MAX_INT,]
               [RLYSVCSIZE = 1 ~ 16,]
               [TCLSVCCNAME = tclsvc name,]
               [TCLSVCPPOS = 1 ~ MAX_INT,]
```

```
[TCLSVCSIZE = 1 ~ 16,]  
[CANSVCNAME = cansvc name,]  
[POSTOKSVCNAME = postoksvc name,]  
[POSTFAILSVCNAME = postfailsvc name]
```

必須項目

- *ServerName* = string
 - サイズ : 63文字以内
 - SERVERセクションで設定したサーバー名と同一に設定します。

選択項目

- *BIDSVCNAME* = string
 - サイズ : 63文字以内
 - ホストで強制的に送信したBIDデータを処理するTmaxシステムのサービス名を設定します。
 - 登録したサービス名はTmaxシステムに登録されている名前である必要があります。
- *BIDSVCPPOS* = numeric
 - 範囲 : 1~MAX_INT
 - BIDデータを処理するサービスを、BIDデータのうちのどの位置から取得するのかを設定します。
 - 設定した位置のサービス名はTmaxシステムに登録されている名前である必要があります。設定した場合、*BIDSVCNAME*項目で定義したサービス名は使用しません。
- *BIDSVCSIZE* = numeric
 - 範囲 : 1~16
 - BIDデータを処理するサービスをBIDデータの特定の位置から取得する際、サービス名の長さを設定します。
 - *BIDSVCPPOS*項目を設定した場合、この項目は必ず設定します。
- *ROPSVCNAME* = string

- サイズ : 63文字以内
 - ホストで特定端末に接続したプリンターにデータを送信した場合、このデータを処理するTmaxシステムのサービスを設定します。
 - 設定したサービス名はTmaxシステムに登録されている名前である必要があります。
- ROPSVCPPOS = numeric
 - 範囲 : 1~MAX_INT
 - ROPデータを処理するサービスを、ROPデータのうちのどの位置から取得するのかを設定します。
 - 設定した位置のサービス名はTmaxシステムに登録されている名前である必要があります。設定すると、ROPSVCPOS項目で定義したサービス名は使用しません。
- ROPSVCSIZE = numeric
 - 範囲 : 1~16
 - ROPデータを処理するサービスをROPデータの特定の位置から取得する際、サービス名の長さを設定します。
- OUTSVCCNAME = string
 - サイズ : 63文字以内
 - ホストから要求したデータを処理するTmaxシステムのサービス名を設定します。
 - OUTSVCPPOS項目に登録せずにこの項目のみを登録する場合、ホストから要求したすべてのデータはこの項目で定義したサービスで要求されます。
 - 設定したサービス名はTmaxシステムに登録されている名前である必要があります。
- OUTSVCPPOS = numeric
 - 範囲 : 1~MAX_INT
 - ホストから要求したデータを処理するサービスを、アウトバウンド・データのうちのどの位置から取得するのかを設定します。
 - 設定した位置のサービス名はTmaxシステムに登録されている名前である必要があります。設定すると、OUTSVCCNAME項目で定義したサービス名は使用しません。

- OUTSVCSIZE = numeric

- 範囲 : 1~16
- アウトバウンド・データを処理するサービスをアウトバウンド・データの特定の位置から取得する際、サービス名の長さを設定します。

- RLYSVCNAME = string

- サイズ : 63文字以内
- tpforwardでHost-linkサービスを呼び出す場合、Host-linkシステムは処理結果を常にRelayサービスに連係します。フラグにTPNOREPLY値を設定することとは関係なく、Host-linkシステムはホストにサービスを要求し、応答を受信したデータをこの項目で定義したサービスにtprelayをします。
- RLYSVCPOS項目とRLYSVCSIZE項目が設定されている場合、この項目は無視されます。ユーザー・ヘッダーにRelayするサービスが同時に設定されている場合、この項目が優先されます。
- 設定したサービス名はTmaxシステムに登録されている名前である必要があります。

- RLYSVCPOS = numeric

- 範囲 : 1~MAX_INT
- tpforwardでHost-linkサービスを呼び出し、ホストから受信した応答データのうちtprelayするサービス名の位置を設定します。
- RLYSVCSIZE項目が同時に設定されている場合、この項目が優先されます。設定すると、RLYSVCNAME項目で定義したサービス名は使用しません。

- RLYSVCSIZE = numeric

- 範囲 : 1~16
- ホストから受信した応答データの中からtprelayするサービス名を検索する際のサービス名の長さを設定します。

- TCLSVCNAME = string

- サイズ : 63文字以内

- tpacallのフラグ値をTPNOREPLYと設定し、Host-linkを呼び出した場合、Host-linkはホストにサービスを要求し、応答を受信したデータを処理することができません。この項目を設定すると、Host-linkは定義したサービスでtpacallを実行します。
 - TCLSVCPOS項目とTCLSVCSIZE項目が設定されている場合、この項目は無視されます。
 - ユーザー・ヘッダーにRelayするサービスが同時に設定されている場合、この項目が優先されます。
 - 設定したサービス名はTmaxシステムに登録されている名前である必要があります。
- TCLSVCPOS = numeric
 - 範囲：1~MAX_INT
 - tpacallのフラグ値をTPNOREPLYと設定し、Host-linkを呼び出した場合、ホストから受信した応答データの中からtpacallを実行するサービス名を検索する位置を設定します。
 - 設定すると、TCLSVCSIZE項目で定義したサービス名は使用しません。
 - TCLSVCSIZE項目が同時に設定されている場合、この項目が優先されます。
- TCLSVCSIZE = numeric
 - 範囲：1~16
 - tpacallを実行するサービス名をホストから受信した応答データの中から検索する際のサービス名の長さを設定します。
- CANSVCNAME = string
 - サイズ：63文字以内
 - タイムアウト後にホストから応答があった場合はデータが破棄されますが、この項目を設定すると、Host-linkは定義したサービスでtpacallを実行します。
 - SVRTYPE項目をLU62Sに設定した場合にのみ該当する項目です。
 - 設定したサービス名はTmaxシステムに登録されている名前である必要があります。
- POSTOKSVCNAME = string
 - サイズ：63文字以内

- SVRTYPE項目をLU0またはLU62Rに設定した場合にのみ該当する項目です。
 - ホストにデータを正常に渡した場合、該当データを定義したサービスでtpacallを実行します。
 - 設定したサービス名はTmaxシステムに登録されている名前である必要があります。
- POSTFAILSVCNAME = string
 - サイズ : 63文字以内
 - SVRTYPE項目をLU0またはLU62Rに設定した場合にのみ該当する項目です。
 - ホストへのデータ伝達が失敗した場合、該当データを定義したサービスでtpacallを実行します。
 - 設定したサービス名はTmaxシステムに登録されている名前である必要があります。

3.2.4. LOGGINGセクション

LOGGINGセクションは、ホストと送受信するすべてのデータを指定したファイルに保存したり、指定したサービス呼び出したりできるように設定します。

```
* LOGGING
Log Server Name      [LOGPATH = log path,]
                     [LOGTYPE = (HEX) | TEXT | E2A,]
                     [LOGSVC = log サービス名]
```

必須項目

- Log Server Name = string
 - サイズ : 63文字以内
 - ログを残すサーバーを設定する項目で、SERVERセクションで設定したサーバー名と同一に設定します。
 - 設定すると、該当サーバーはログをファイルに残したり、LOGSVC項目に設定したサービスを呼び出したりできるようになります。

選択項目

- LOGPATH = string

- サイズ：256文字以内
- ログファイルのパスと名前を設定します。
- 絶対パス(「/」で始まるパス)で定義した場合、そのパスをそのまま使用します。相対パス(「/」で始まらないパス)で定義した場合、Tmaxの環境設定ファイルのNODEセクションのULOGDIR項目に定義したディレクトリーの下位パスにログ・ファイルが格納されます。
- LOGSVC項目が設定された場合、この項目は無視されます。
- LOGTYPE項目を同時に定義しなければ、ロギングは実行されません。
- LOGTYPE = (HEX) | TEXT | E2A
 - データをロギングする際の形式を設定します。
 - 以下は、各設定値についての説明です。

設定値	説明
HEX	出力データを16進数の値で出力します(デフォルト値)
TEXT	文字列形式のデータを行単位で出力します
E2A	EBCDICコードをASCIIコードに変換後、HEXに出力します

- LOGPATH項目を同時に定義しなければ、ログを残すサーバー名でログ・ファイルが格納されます。
- LOGSVC = string
 - サイズ：63文字以内
 - 送受信データをロギングする際、Host-linkシステムで提供するファイルにロギングせず、ユーザーが作成したサービスにロギングする際に設定します。
 - Host-linkシステムは特定ヘッダー(LOGHEADER、userinc/hlinkapi.h)を追加し、データと一緒にサービスを呼び出します。
 - 設定したサービス名はTmaxシステムに登録されている名前である必要があります。

使用例

データのロギングの例は以下のとおりです。

```
*LOGGING
snalu0gw00  LOGPATH = aaa,
             LOGTYPE = HEX,
             LOGSVC = LOGSVC
```

Tmax環境設定ファイルのNODEセクションにULOGDIR=/home/tmax/log/ulogと設定すると、ログ・ファイルは「/home/tmax/log/ulog/aaa_サーバー-index.現在の時間」に保存されます。

LOGTYPE項目をHEXに設定したため、以下のようにHEX値でデータを格納します。LOGSVC項目を設定したため、Host-linkシステムが提供するファイルにログインせず、設定したLOGSVCでログインします。

```
HOSTLINK  HOST  Length[70]  time[14:10:58.375]
00000000  e3f1c9d8  c3e5c840  40d3f0f5  f8f3f740  | .....@@.....@|
00000010  40f9f1f0  4040f0f0  f0f04040  40404040  | @...@@.....@@@@@|
00000020  40e5c8e3  f5f0f040  d3404080  80808080  | @.....@.@@.....|
00000030  a040f4f5  f9f9f5f0  f4f0f0f8  f0f8f3f9  | .@.....|
00000040  f0f91ed5  1e0d      | .....|

HOST      HOSTLINK  Length[70]  time[14:10:58.376]
00000000  e3f1c9d8  c3e5c840  40d3f0f5  f8f3f740  | .....@@.....@|
00000010  40f9f1f0  4040f0f0  f0f04040  40404040  | @...@@.....@@@@@|
00000020  40e5c8e3  f5f0f040  d3404080  80808080  | @.....@.@@.....|
00000030  a040f4f5  f9f9f5f0  f4f0f0f8  f0f8f3f9  | .@.....|
00000040  f0f91ed5  1e0d      | .....|
```

3.2.5. CONVERSIONセクション

CONVERSIONセクションの形式は以下のとおりです。該当サーバーのデータを変換するセクションです。

```
* CONVERSION
ServerName      [ INBOUND = Y | (N) ]
                 [ INBOUND_KOR = Y | (N) , ]
                 [ INBOUND_SKIP = 1 ~ MAX_INT , ]
                 [ OUTBOUND = Y | (N) , ]
                 [ OUTBOUND_KOR = Y | (N) , ]
                 [ OUTBOUND_SKIP = 1 ~ MAX_INT , ]
                 [ MAPPATH = map path ]
```

必須項目

- *ServerName* = string
 - サイズ : 63文字以内
 - SERVERセクションで設定したサーバー名と同一に設定します。

選択項目

- INBOUND = Y|(N)
 - デフォルト値 : N
 - 項目をYに設定すると、Host-linkはホストに送信するすべてのデータをASCIIからEBCDICに変換します。
- INBOUND_SKIP = numeric
 - 範囲 : 1~MAX_INT
 - コードを変換する際に除外する特定バイトを設定します。データにHEXデータが含まれている場合はこれを見捨てるために使用します。
 - INBOUND =Nと設定されている場合、INBOUND_SKIPに設定された内容は見捨てられます。
- OUTBOUND = Y|(N)
 - デフォルト値 : N
 - 項目をYに設定するとEBCDICをASCIIに変換します。
- OUTBOUND_SKIP = numeric
 - 範囲 : 1~MAX_INT
 - コードを変換する際に除外する特定バイトを設定します。
 - データにHEXデータが含まれている場合、これを見捨てるために使用します。
 - OUTBOUND =Nと設定されている場合、OUTBOUND_SKIPに設定された内容は見捨てられます。
- MAPPATH = literal
 - サイズ : 255文字以内
 - Host-linkとホストが通信をするには、各サービスで処理するフォーマットでデータを送受信します。各フォーマットを定義するファイルが存在するディレクトリを設定します。
 - ファイル名は以下のとおりです。

区分	説明
LU0	データの前部分8バイト(ユーザー・ヘッダーがある場合はユーザー・ヘッダー以降の8バイト)を使用します
LU62S	TPGWINFO_Tデータ構造体のpgmnameを使用します
LU62R	Tmaxのサービス名を使用します。Mapファイル名が8文字であるため、Tmaxサービスは8文字以内で定義します

使用例

データ変換例は以下のとおりです。

```
*CONVERSION
snalugw00    INBOUND = Y,
              INBOUND_KOR = N,
              INBOUND_SKIP = 50,
              OUTBOUND = N,
              OUTBOUND_KOR = N,
              OUTBOUND_SKIP = 55,
              ...
```

例を実行すると、Host-linkはホストに送信するすべてのデータをASCIIからEBCDICに変換します。INBOUND_SKIP項目で設定した分だけを除いて変換します。INBOUNDとは逆に、OUTBOUND項目はNに設定したため、EBCDICからASCIIへの変換は実行されません。

CONVERSIONセクションでMAPPATH項目を設定する場合、Mapファイルを登録します。MAPファイル名はMAPPATHで定義されたパスに存在する必要があり、<ファイル名.map>ファイルに登録します。ファイル名は8文字以内で定義します。

以下は、Mapファイルの登録例と各項目についての説明です。

```
# COMMSIZEはINPUT/OUTPUTで一緒に使用する必要があるため、INPUT/OUTPUTのうち
# 大きいサイズの値を入力します。
*COMMSIZE
1024

*INPUT
#-----
# 項目名   項目の長さ   データ型
# データ型: CHAR, NUMERIC(UNPACK), BINARY, USER-TYPE
#-----
a           10         CHAR
c           5          NUMERIC
d           10         BINARY
```

*OUTPUT		
a	10	CHAR
c	5	NUMERIC
d	10	BINARY
e	3	USER1
<USER1>		
e1	6	CHAR
e3	8	BINARY
</USER1>		

項目	説明
COMMSIZE	リモート・ノードと送受信するデータの長さを設定する項目です。すべてのサービス・メッセージのうち最も長いメッセージの長さを入力します
INPUT	ゲートウェイからリモート・ノードに送信するメッセージの情報を登録する項目です。 ゲートウェイからリモート・ノードにサービスを要求する場合、リモート・ノードに要求するメッセージのフォーマットを登録します。反対に、リモート・ノードからTmaxにサービスを要求する場合は、Tmaxサービスの処理結果のフォーマットを登録します
OUTPUT	リモート・ノードからゲートウェイに送信されたメッセージの情報を登録する項目です。 ゲートウェイからリモート・ノードにサービスを要求する場合、リモート・ノードで処理した結果を受信したメッセージのフォーマットを登録します。反対に、リモート・ノードからTmaxにサービスを要求する場合は、Tmaxにサービスを要求するメッセージのフォーマットを登録します

INPUT/OUTPUT領域は、以下の項目で構成されます。

- 項目名

ユーザーが参照用にのみ使用する名前です。ゲートウェイは項目名として認識し、内部では使用しません。

- 項目の長さ

送受信データで各項目の実際の長さを示します。各項目の長さのみを表現すると、ゲートウェイ内部でオフセットに変換して使用します。オフセットは常に0から開始します。

- データ型

送受信データで各項目のデータ型を指定します。

データ型	説明
CHAR	英文字、数字を含む場合に使用します。CHAR型を使用する場合、変換後のメッセージのサイズを正しく設定する必要があります

データ型	説明
NUMERIC	<p>データがすべて数字の場合に使用し、コード変換はCHAR型と同一です。</p> <p>NUMERICが整数型を意味することではないことに注意します</p>
BINARY	<p>この型のデータはコード変換が実行されません。データのうちコードを変換しない場合はこの型に設定します</p>
USER-TYPE	<p>配列形式のデータ型を指定する場合に使用します。</p> <p>たとえば、画面に複数の項目が繰り返し出てくる場合、各項目を反復回数の分だけ登録が必要になるという煩わしさがあります。または、反復回数をデータで表現する場合、何回繰り返されるかが分からないため、コード変換が困難です。このためにユーザーはユーザー定義型を指定し、配列形式のデータを簡単に変換できます。</p> <p>上のMapファイルにおいて、USER1はユーザーが指定したユーザー定義型です。型名はユーザーが任意で指定できます。</p> <p>[注意]</p> <p>ユーザー定義型のデータで項目の長さは、配列の反復回数が保存されるバッファのサイズを指定するという点に注意します。つまり、配列の反復回数はメッセージごとに流動的です。反復回数の値は数字で、CHAR形式で保存される必要があります。それ以降、配列の各項目の開始前に<USER-TYPE>、終了後に</USER-TYPE>を挿入します。上記の場合は、e1、e2、e3を配列項目と認識し、メッセージごとのe項目の開始位置で項目の長さである3の分だけ先に読み込み、反復回数を読み込んだ後、その回数分のコードを繰り返して変換します</p>

3.2.6. DUMMYDATAセクション

ホストから受信したデータのサイズが0である場合や、データのうち特定値が設定されている場合、DUMMYデータと見なすように設定するセクションです。DUMMYDATAセクションはサーバー・タイプ(SVRTYPE)がLU0の場合にのみ設定します。

以下は、DUMMYDATAセクションの形式です。

```
* DUMMYDATA
ServerName      [ZEROSIZE = (Y)|N,]
                 [DUMMYPOS = dummy position,]
                 [DUMMYVAL = dummy value]
```

必須項目

- *ServerName* = string
 - サイズ : 63文字以内

- SERVERセクションで設定したサーバー名と同一に設定します。

選択項目

- ZEROSIZE = (Y)|N

- デフォルト値：Y
- ホストから受信したデータの長さが0バイトの場合に使用する項目です。
- 以下は、各設定値についての説明です。

設定値	説明
Y	ホストで0バイトが受信されても応答と見なし、Host-linkシステムのサービスを呼び出すクライアントに応答を送信します
N	0バイトが受信された場合、該当データを無視します

- DUMMYPOS = numeric

- 範囲：1~MAX_INT
- ホストから受信したデータで無視する値がある位置を設定します。
- 設定していない場合、受信したすべてのデータはクライアントに送信されます。

- DUMMYVAL = string

- サイズ：8文字以内
- DUMMYPOS項目で指定した位置の値と比較する値を設定します。設定した値はEBCDICコードに変換して比較します。

3.2.7. USERHEADERセクション

USERHEADERセクションの形式は以下のとおりです。Host-linkとオープン環境システム間でユーザー・ヘッダーを設定するセクションです。

```
* USERHEADER
ServerName      [HSIZE = 1 ~ 512,]
                 [SVCPOS = 1 ~ HSIZE - 1,]
                 [SVCSIZE = 1 ~ 16,]
```

```
[LUTYPEPOS = 1 ~ HSIZE - 1,]  
[LUTYPEPOS = 1 ~ 8,]  
[LUNAMEPOS = 1 ~ HSIZE - 1,]  
[LUNAMESIZE = 1 ~ 8,]  
[WSNAMEPOS = 1 ~ HSIZE - 1,]  
[WSNAMESIZE = 1 ~ 8,]  
[ERRCODEPOS = 1 ~ HSIZE - 1,]  
[ERRCODESIZE = 1 ~ 4,]  
[TIMEWAITPOS = 1 ~ HSIZE - 1]
```

必須項目

- *ServerName* = string
 - サイズ : 63文字以内
 - SERVERセクションで設定したサーバー名と同一に設定します。

選択項目

- *HSIZE* = numeric
 - 範囲 : 1~512
 - ヘッダーの長さを設定します。
 - Host-linkはヘッダー・データを保管し、ホストから応答を受信すると、該当ヘッダー・データを応答の前部分に追加してオープン環境システムに送信します。また、ホストから要求したサービスやBIDデータを受信した場合にもこのヘッダーを受信データの前に追加して、指定したサービスを呼び出します。
 - ヘッダー・データの中からセッションを検索する情報を取得するため、セッションの運用をDedicate方式に設定した場合はこのヘッダーを必ず指定します。
- *SVCPOS* = numeric
 - 範囲 : 1~HSIZE-1
 - ユーザー・ヘッダー・データのうちサービス名の位置を設定します。
 - tpacallにTPNOREPLAYやtpforwardでHost-linkを呼び出し、ホストからの応答を受信すると、Host-linkはこの項目に設定したサービス名でサービスを呼び出します。
- *SVCSIZE* = numeric

- 範囲：1~16
- ユーザー・ヘッダー・データのうちサービス名の長さを設定します。
- LUTYPEPOS = numeric
 - 範囲：1~HSIZE-1
 - データをホストに送信する際、業務別に区分されたセッションの中から希望するセッションの位置を設定します。
 - 1つのHost-linkに属する多くのセッションを業務別に区分して使用する場合に設定する項目です。
 - SVRTYPE項目がLU0の場合にのみ使用する項目です。
- LUTYPESIZE = numeric
 - 範囲：1~8
 - データをホストに送信する際、業務別に区分されたセッションの中から特定セッションの長さを設定します。
 - SVRTYPE項目がLU0の場合にのみ使用する項目です。
- LUNAMEPOS = numeric
 - 範囲：1~HSIZE-1
 - Host-linkシステムがオープン環境システムに要求したサービスを送受信する際に使用するセッションの位置を設定します。
 - SVRTYPE項目がLU0の場合にのみ使用する項目です。
- LUNAMESIZE = numeric
 - 範囲：1~8
 - Host-linkシステムがオープン環境システムに要求したサービスを送受信する際に使用するセッションの長さを設定します。
 - SVRTYPE項目がLU0の場合にのみ使用する項目です。

- WSNAMERPOS = numeric
 - 範囲：1~HSIZE-1
 - セッションの運用をDedicate方式で使用する場合にのみ使用する項目で、特定セッションでデータを送受信するためにセッションの位置を設定します。
 - SVRTYPE項目がLU0の場合にのみ使用する項目です。
- WSNAMESIZE = numeric
 - 範囲：1~8
 - 特定セッションでデータを送受信するためにセッションの長さを設定します。
 - Host-linkシステムがセッションの運用をDedicate方式で使用する場合にのみ使用する項目であるため、SVRTYPE項目がLU0の場合にのみ使用する項目です。
- ERRCODEPOS = numeric
 - 範囲：1~HSIZE-1
 - Host-linkシステムがホストにサービスを要求する際にエラーが発生した場合やタイムアウトにより応答を受信できない場合に、オープン環境システムにエラーコードを渡すために使用する項目で、エラーコードがある位置を設定します。
 - Host-linkはこの項目に設定した位置にERRCODESIZE項目で設定した長さのエラーコードを文字に変換して応答します。
- ERRCODESIZE = numeric
 - 範囲：1~4
 - ERRCODEPOS項目と一緒に使用する項目で、エラーコードの長さを設定します。
- TIMEWAITPOS = numeric
 - 範囲：1~HSIZE-1
 - タイムアウトが発生した場合、セッションを切断して再度開始するのか(NOWAIT)、セッションをそのまま使用するのか(WAIT)を設定した位置を設定します。

ゲートウェイ・ヘッダー

SVRTYPE項目がLU62Sの場合、ユーザー・ヘッダーの前にゲートウェイ・ヘッダーが存在する必要があります。ゲートウェイ・ヘッダーは「usrinc/hlinkapi.h」に以下のように宣言されています。

```
struct tpgwinfo {
    char   svc[XATMI_SERVICE_NAME_LENGTH];
    char   trxid[HOST_TRANS_LENGTH];
    char   pgmname[HOST_PROGRAM_LENGTH];
    char   userid[HOST_USERID_SIZE];
    char   passwd[HOST_PASSWD_SIZE];
    char   resvd[16];
};
typedef struct tpgwinfo TPGWINFO_T;
```

メンバー	説明
char svc[XATMI_SERVICE_NAME_LENGTH]	非ブロッキングモードで使用する場合や非同期方式の場合、応答データを処理するサービス名を設定します
char trxid[HOST_TRANS_LENGTH]	呼び出されるホストのTP名を設定します。この項目は、SVRTYPEがLU62SでfunctionがDPLの場合に記載必須の項目です。サイズは4文字以内で設定します
char pgmname[HOST_PROGRAM_LENGTH]	呼び出されるホストのプログラム名を設定します。SVRTYPE項目がLU62Sに設定されている場合、この項目に設定されている名前でMAPファイルを参照します
char userid[HOST_USERID_SIZE]	ユーザーIDを設定します
char passwd[HOST_PASSWD_SIZE]	パスワードを設定します
char resvd[16]	現在は使用されていません

使用例

ユーザー・ヘッダーの例は以下のとおりです。

```
*USERHEADER
snalu0gw00    HSIZE = 41,          # user total header size
              SVCPOS = 1,         # user defined svc name position
              SVCSIZE = 15,       # user defined svc name length
              LUTYPEPOS = 16,     # inbound lu type position
              LUTYPE SIZE = 7,    # inbound lu type length
              LUNAMEPOS = 23,     # output only, lu name position
              LUNAME SIZE = 7,    # output only, lu name length
              WSNAMEPOS = 30,     # only INBOUND LU is DEDICATE
              WSNAME SIZE = 7,    # only INBOUND LU is DEDICATE
```

```
ERRCODEPOS = 37,      # output only
ERRCODESIZE = 3,      # output only
TIMEWAITPOS = 40      # time wait position
```

3.2.8. LUINFOセクション

LUINFOセクションの形式は以下のとおりです。ホストとHost-linkシステムのセッション情報を設定するセクションです。

```
*LUINFO
LU Name      SVRNAME = servername
              [HOSTID = ホストID,]
              [WSNAME = ws name,]
              [DIRECTION = INBOUND | OUTBOUND,]
              [LUTYPE = lu type,]
              [FQPLUNAME = partner-fully qualified lu name,]
              [TPNAME = ホストとセッションを接続するためのLU名,]
              [MODENAME = セッションの用途,]
              [SYNCMODE = NONE | CONFIRM | SYNCPT,]
              [SESSION = 1 ~ 50,]
              [NETNAME = luネットワーク名]
```

必須項目

- *LU Name* = string
 - サイズ : 8文字以内
 - LUセッション名を設定します。
- *SVRNAME* = string
 - サイズ : 63文字以内
 - LUセッションが接続されるサーバー名を設定します。

選択項目

- *HOSTID* = string
 - サイズ : 4文字以内
 - 該当セッションが属するホストIDを設定します。

- WSNNAME = string
 - サイズ: 16文字以内
 - Host-linkがセッションをDedicate方式で運用する場合に使用する項目です。
 - 設定した値をユーザー・ヘッダー情報に保存し、該当位置をUSERHEADERセクションのWSNAMEPOS項目とWSNAMESIZE項目に設定した場合、該当セッションでデータを送受信できます。
 - SNA LU 0に該当する項目です。

 - DIRECTION = INBOUND | OUTBOUND
 - 該当セッションがインバウンド・セッションであるか、アウトバウンド・セッションであるかを設定します。
 - SNA LU 0に該当する項目です。

 - LUTYPE = string
 - サイズ: 8文字以内
 - 各セッションを用途別に区分するための項目です。インバウンド・セッション(DIRECTION = INBOUND)の場合、ユーザーが値を設定し、Host-link環境設定ファイルのUSERHEADERセクションのうちLUTYPEPOSとLUTYPESIZE項目に設定されている位置に、この項目で設定した型を設定します。
 - アウトバウンド・セッション(DIRECTION = OUTBOUND)の場合は次の3種類に分けられます。
- | 区分 | 説明 |
|----------|----------------------------|
| 一般セッション | ホストで先にサービスを呼び出すセッションです |
| BIDセッション | ホストで強制メッセージを渡すためのセッションです |
| ROPセッション | 特定端末のプリンターにデータを送信するセッションです |
- SNA LU 0に該当する項目です。

 - FQPLUNAME = string
 - サイズ: 17文字以内
 - SNA LU 6.2に該当する項目で、相対LUの全体名(partner-fully qualified lu name)を設定します。

 - TPNAME = string

- サイズ : 64文字以内
- LUのローカルTPNAMEを設定します。
- SNA LU 6.2に該当する項目です。
- MODENAME = string
 - サイズ : 8文字以内
 - 該当セッションのモード名を設定します。使用するモード名についてはホスト管理者にお問い合わせください。
 - SNA LU 6.2に該当する項目です。
- SYNCMODE = NONE | CONFIRM | SYNCPT
 - デフォルト値 : NONE
 - ホストのSyncpointのレベルをサポートします。
 - SNA LU 6.2に該当する項目です。
- SESSION = numeric
 - 範囲 : 1 ~ 50
 - 1つのLUに設定できるセッション数を設定します。
 - SNA LU 0の場合は1つのLUあたり1つのセッションのみ接続できるため、1に設定できます。
- NETNAME = string
 - サイズ : 8文字以内
 - LUのネットワーク名を設定します。完全なLU名 (fully qualified lu name)は「LUネットワーク名.LU名」で構成されます。たとえば、「TESTNET.TESTLU」の場合、「TESTNET」がLUネットワーク名です。

3.3. サービス別の環境設定

本節では、それぞれのサービスについてと環境設定方法について説明します。

3.3.1. インバウンド・サービス

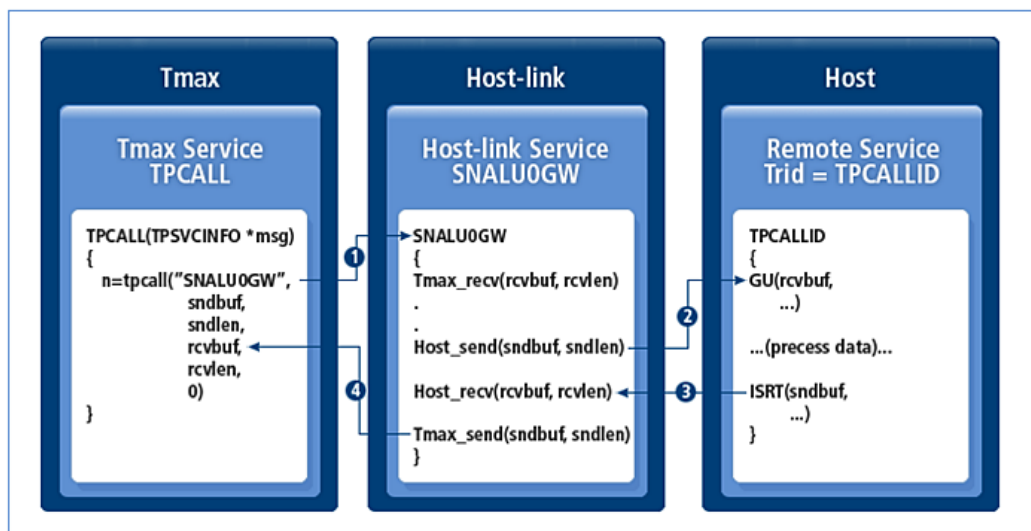
インバウンド・サービスには、**tpcall**、**tpacall**、**tpforward**サービスが含まれます。

tpcall

tpcallサービスは、Tmaxシステムの他のサーバー・プロセスで提供するサービスと同じように使用できるサービスです。Host-linkシステムは一般サーバー・プロセスとは異なり、CPCを使用して同時に複数のサービスを処理できます。たとえば、CPCを5に設定した場合、Host-linkは同時に5件のサービスを処理できます。

サービスの処理手順は以下のとおりです。

[図 3.1] tpcallサービス



1. TmaxシステムでTmax環境設定ファイルに定義したHost-linkサービスを呼び出します。
2. Host-linkは該当ホストで提供するプログラムにデータを送信します。
3. ホストは該当サービス进行处理し、処理結果をHost-linkに送信します。
4. Host-linkはサービスを呼び出したTmaxシステムに応答データを送信します。

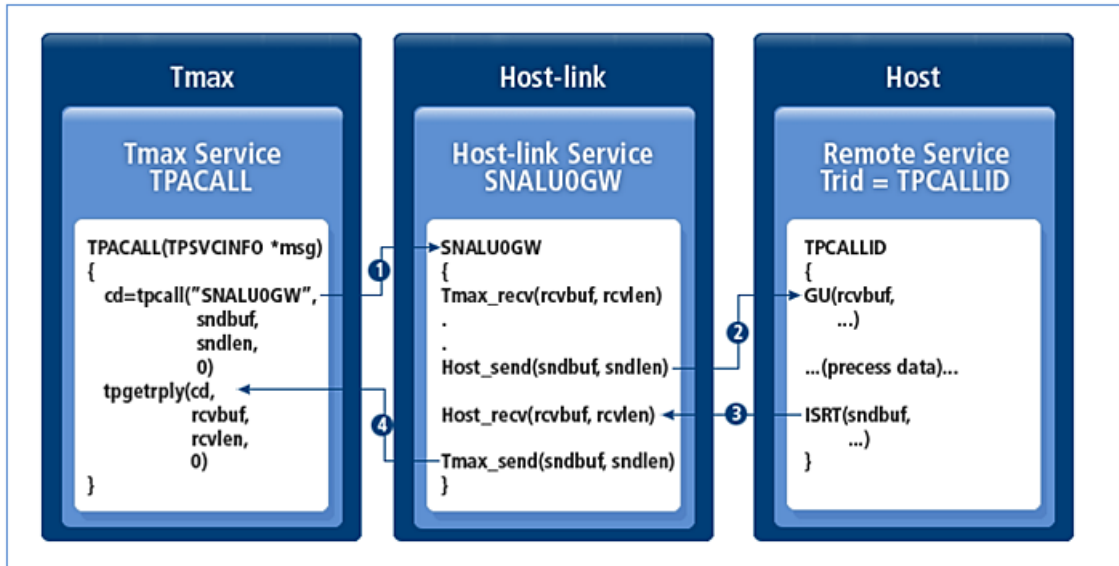
tpacall

tpacallサービスは要求されたサービスを処理後、Tmaxシステムに応答するサービスと応答しないサービスに分けられます。

- Tmaxシステムに応答するサービス

Tmaxシステムに応答するサービスはtpcallサービスと同じように処理されます。

[図 3.2] tpacall - Tmaxシステムに回答するサービス

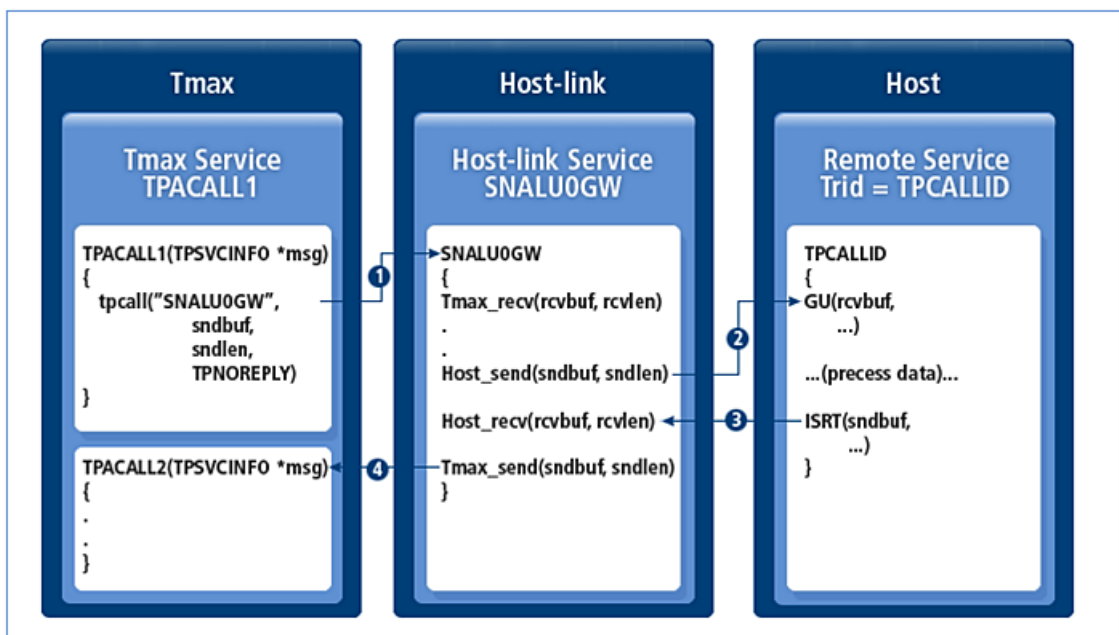


1. TmaxシステムでTmax環境設定ファイルに定義したHost-linkサービスを呼び出します。
2. Host-linkは該当ホストで提供するプログラムにデータを送信します。
3. ホストは該当サービス进行处理し、処理結果をHost-linkに送信します。
4. Host-linkはサービスを呼び出したTmaxシステムに回答データを送信します。

- Tmaxシステムに回答しないサービス

Tmaxシステムに回答しないサービスの処理手順は以下のとおりです。

[図 3.3] tpacall - Tmaxシステムに回答しないサービス



1. TmaxシステムでTmax環境設定ファイルに定義したHost-linkサービスを呼び出します。
2. Host-linkは該当ホストで提供するプログラムにデータを送信します。
3. ホストは該当サービス进行处理し、処理結果をHost-linkに送信します。
4. Host-linkは環境設定ファイルに設定された連係サービスを呼び出します。

Tmaxシステムに応答しないサービスの場合、Host-linkがホストの応答を受信し、他のサービスに連係できます。連係サービス名は以下の方法で認識します。以下の方法をHost-link環境設定ファイルに設定しない場合、ホストで受信した応答データは無視されます。tpacallを実行すると、フラグにTPNOREPLY値を設定してサービスを呼び出し、Host-link環境設定ファイルに以下のような連係サービスを定義します。

– SECVICEセクション

```
*SERVICE
sna lu0gw00    TCLSVCNAME = TPACALL_RECV,      # default svc name
               TCLSVCPOS  = 1,
               TCLSVCSIZE = 10,
               ...
```

ホストで受信したデータのうち1番目から11番目のデータをサービス名として認識し、該当サービスを呼び出します。TCLSVCPOS項目とTCLSVCSIZE項目を定義しない場合、TCLSVCNAMEに定義したサービス名で呼び出します。

– USERHEADERセクション

```
*USERHEADER
sna lu0gw00    HSIZE = 100,      # user total header size
               SVCPOS  = 1,      # user defined svc name position
               SVCSIZE = 10,     # user defined svc name size
               LUNAMEPOS = 11,   # output only
               LUNAMESIZE = 10, # output only
               WSNAMEPOS = 21,   # only INBOUND LU is DEDICATE
               WSNAME SIZE = 10, # only INBOUND LU is DEDICATE
               ERRCODEPOS = 31, # output only
               ERRCODESIZE = 4, # output only
               TIMEWAITPOS = 1   # WAIT or NOWAIT value position
```

SVCPOS項目とSVC SIZE項目でサービス名を認識します。

– TPGWINFO_T

TPGWINFO_TのSVCメンバーで定義したサービス名を認識します。

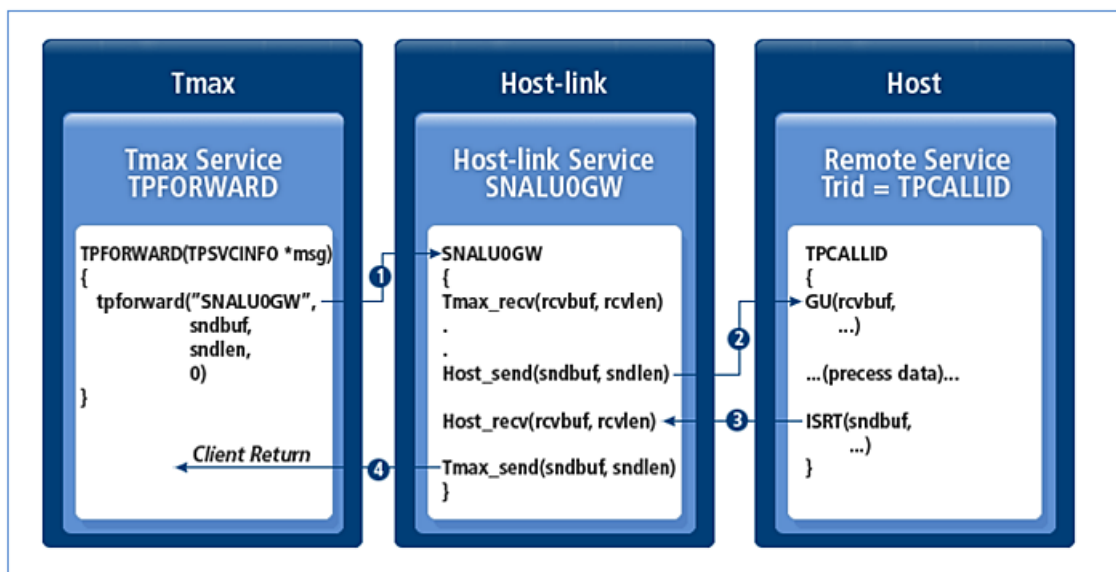
tpforward

tpforwardサービスは、フラグにTPNOREPLY値を設定せずにHost-linkサービスを呼び出すサービスと、TPNOREPLY値を設定してHost-linkサービスを呼び出すサービスに分けられます。

● TPNOREPLY値を設定しない場合

サーバー・プログラムでのみ呼び出すことができるサービスで、Tmax 3.5バージョンでは以下のように動作しますが、3.8バージョンからはフラグ値の設定とは関係なく、常にTPNOREPLY値を設定した場合に動作します。

[図 3.4] tpforward - TPNOREPLYを設定した場合

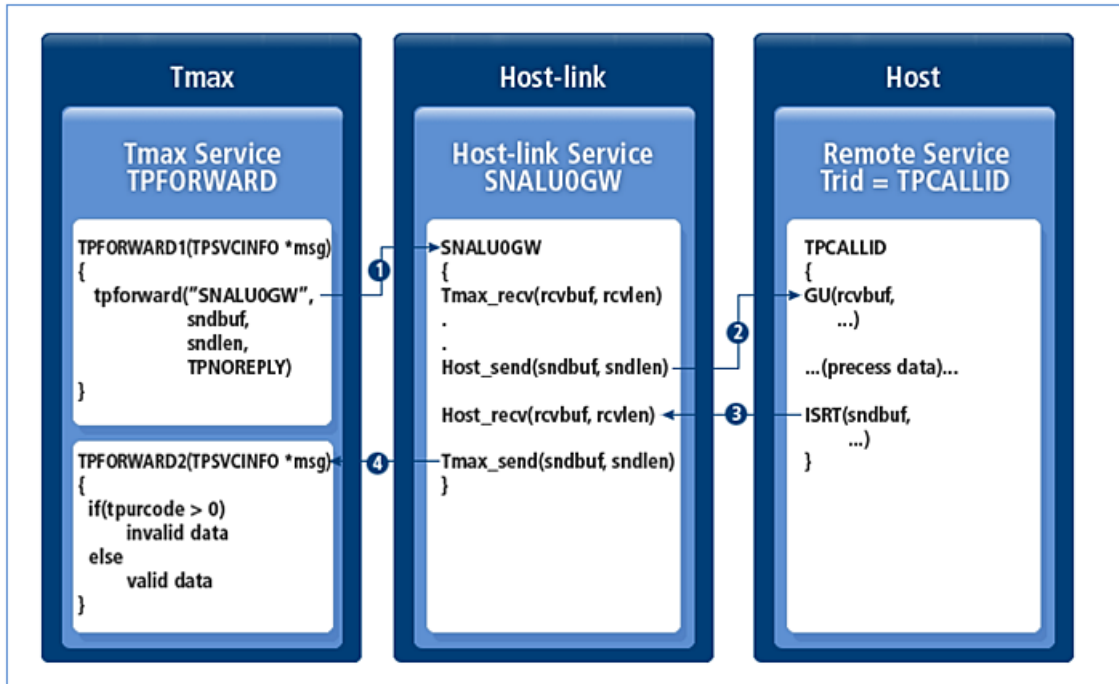


1. TmaxシステムでTmax環境設定ファイルに定義したHost-linkサービスを呼び出します。
2. Host-linkは該当ホストで提供するプログラムにデータを送信します。
3. ホストは該当サービス进行处理し、処理結果をHost-linkに送信します。
4. Host-linkはtpforwardサービスを呼び出したクライアントに応答を送信します。

● TPNOREPLY値を設定した場合

サービスの処理手順は以下のとおりです。

[図 3.5] tpforward - TPNOREPLYを設定していない場合



1. TmaxシステムでTmax環境設定ファイルに定義したHost-linkサービスを呼び出します。
2. Host-linkは該当ホストで提供するプログラムにデータを送信します。
3. ホストは該当サービス进行处理し、処理結果をHost-linkに送信します。
4. Host-linkは環境設定ファイルに定義したサービスで応答を送信します。

サーバー・プロセスでtpcallを使用してHost-linkシステムのサービスを呼び出すと、応答が受信されるまで該当サーバー・プロセスはブロック状態です。このような短所を補完するために、サービスを送信と受信サービスに分離します。

送信サービスはHost-linkシステムにtpforwardを使用してサービスを呼び出し、サービスを終了します。Host-linkシステムはホストで応答を受信すると、受信サービスを呼び出して応答进行处理します。受信サービス名はtpacallと同様にHost-link環境設定ファイルを使用して認識します。

– SERVICEセクション

```

*SERVICE
snalu0gw00    ...
               #RLYSVCNAME = TPACALL_RECV,      # default svc name
               RLYSVCPOS = 1,                    # svcname position
               RLYSVCSIZE = 10,                  # max length 16
               ...
  
```

Host-linkシステムがホストにサービスを要求してエラーが発生した場合、Host-linkはRLYSVCPOS項目とRLYSVCSIZE項目に定義したサービスにエラーを渡します。RLYSVCPOS項目とRLYSVCSIZE

項目で定義したサービスは、サービスの開始前にサービス・データが正常であるか異常であるかを必ず判断します。tpurcodeが0より大きい場合はエラー、そうでない場合は正常なデータと判断します。

RLYSVCPOS項目とRLYSVCSIZE項目が定義されていない場合、RLYSVCNAMEに定義したサービスにエラーを渡します。

– USERHEADERセクション

```
*USERHEADER
snalu0gw00      HSIZE = 100,          # user total header size
                HSVCPPOS = 1,        # user defined svc name position
                HSVCSIZE = 10,       # user defined svc name size
                ...
```

HSVCPPOS項目とHSVCSIZE項目でサービス名を認識します。

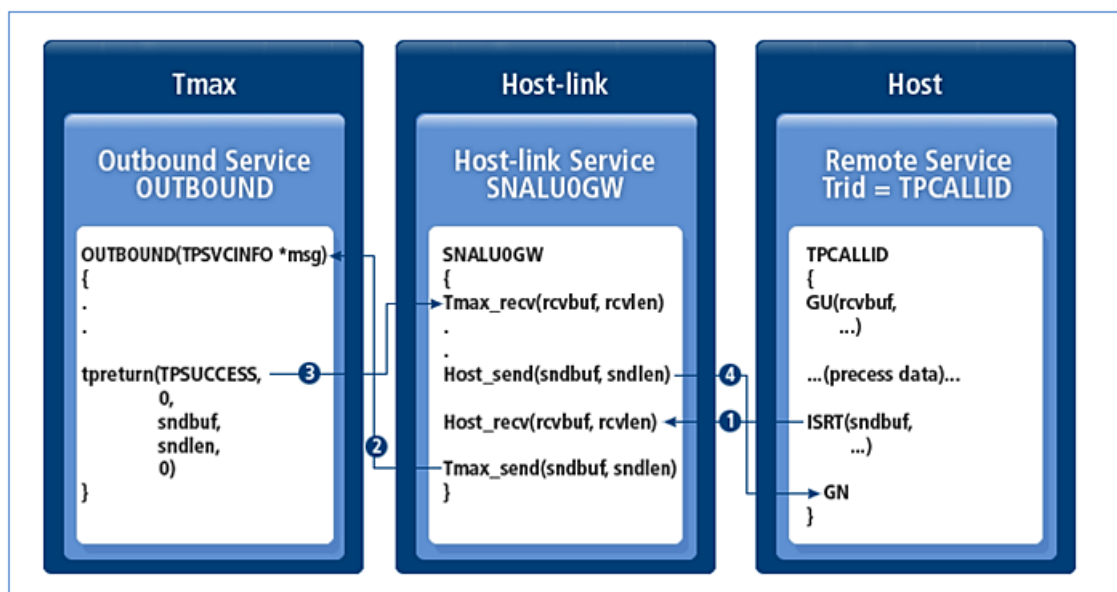
– TPGWINFO_T

TPGWINFO_TのSVCメンバーで定義したサービス名を認識します。

3.3.2. アウトバウンド・サービス

アウトバウンド・サービスは、ホストでTmaxシステムのサービスを呼び出し、処理結果を受信する形式のサービスです。アウトバウンド・サービスの処理プロセスは以下のとおりです。

[図 3.6] アウトバウンド・サービスの処理手順



1. ホストでHost-linkにサービスを要求します。
2. Host-linkは環境設定ファイルに定義されたとおりTmaxシステムのサービスを呼び出します。

3. Tmaxシステムのサービスを処理し、処理結果をHost-linkに送信します。
4. Host-linkは応答データをホストに送信します。

Tmaxシステムで提供するサービス名は以下の2つの方式で認識します。

- SERVICEセクション

```
*SERVICE
snalu0gw00    ...
               #OUTSVCTYPE = OUTBOUND,          # default svc name
               OUTSVCPPOS = 1,                    # outsvc position
               OUTSVCSIZE = 10,                   # max length 16
               ...
```

Host-linkは、OUTSVCPPOS項目とOUTSVCSIZE項目で定義した位置でサービス名を検索してサービス呼び出します。OUTSVCPPOS項目とOUTSVCSIZE項目を定義していない場合、すべてのアウトバウンド・データはOUTSVCTYPE項目で定義したサービス呼び出します。

- USERHEADERセクション

```
*USERHEADER
snalu0gw00    ...
               ERRCODEPOS = 1,                    #response error position
               ERRCODESIZE = 4,                   # max length 4
               ...
```

ERRCODEPOS項目とERRCODESIZE項目は、Tmaxシステムで提供するサービスがない場合や異常終了した場合にエラーを送信するために使用する項目です。エラー・データは、ホストから受信したデータのうちERRCODEPOS項目に定義した位置にERRCODESIZE項目分のサイズで定義されます。このエラー・データをホストに送信します。

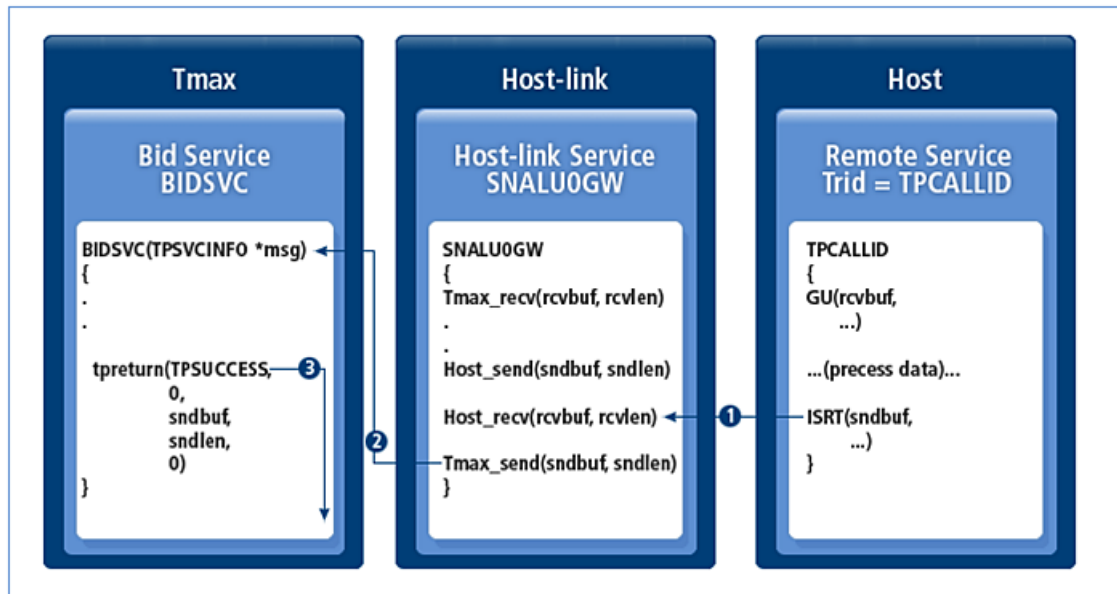
3.3.3. BIDおよびROP

ホストで強制的にHost-linkシステムの特定セッションにメッセージを送信できる機能があり、これをBIDやROPといいます。Host-linkシステムはBIDやROPデータを受信すると、Host-link環境設定ファイルのSERVICEセクションに定義したBIDやROPサービス呼び出します。

アウトバウンド・サービスは処理結果をホストに送信する必要がありますが、BIDデータはホストから受信のみ行い、応答しない形式です。

サービスの処理手順は以下のとおりです。

【図 3.7】 BIDおよびROPサービスの処理手順



1. ホストでBIDまたはROP要求を呼び出します。
2. ゲートウェイで設定のあるTmaxのサービスに当該要求を呼び出します。
3. サービスでは業務を処理した後、tpreturnを実行します。要求そのものがBIDまたはROPなので、応答をホストまで送信はしません。

Tmaxシステムで提供するサービス名は環境設定ファイルのSERVERセクションを通じて認識します。

```
*SERVICE
snalu0gw00    ...
               #BIDSVCNAME = BIDSVC,           # default svc name
               BIDSVCPOS = 1,                   # svcname position
               BIDSVCSize = 10,                  # max length 16
               #ROPSVCNAME = ROPSVC,           # default svc name
               ROPSVCPOS = 11,                  # svcname position
               ROPSVCSize = 10,                  # max length 16
               ...
```

Host-linkは、BIDSVCPOS項目とBIDSVCSize項目で定義したとおりサービス名を検索してサービス呼び出します。(ROPサービスの場合、ROPSVCPOS項目とROPSVCSize項目)

BIDSVCPOS項目とBIDSVCSize項目を定義していない場合、すべてのBIDデータはBIDSVCNAME項目で定義したサービス呼び出します。(ROPサービスの場合、ROPSVCNAME項目)

3.4. 環境設定ファイルのコンパイル

3.4.1. hkcfl

hkcflは、テキスト形式のホストリンク環境設定ファイルをコンパイルして、バイナリ・ホストリンク環境設定ファイル(hlinkcfg)を作成します。コンパイル中にエラーが発生すると、コンパイルを中断します。コンパイルが正常に完了すると、バイナリ環境設定ファイルが生成されます。

- 使用方法

```
hkcfl [-h] [-p] [-o バイナリ・ホストリンク環境設定ファイル名] -i テキスト・ホストリンク環境設定ファイル名  
      -c コメント処理文字
```

項目	説明
[-h]	コマンドのヘルプ・オプションです
[-p]	ホストリンク環境設定ファイルのコンパイル実行内容を画面に出力します
[バイナリ・ホストリンク環境設定ファイル名]	コンパイル結果生成されるバイナリ・ホストリンク環境設定ファイルの名前を指定します。パスと一緒に指定することができます。パスを指定しない場合、\${TMAXDIR}/configに結果ファイルが生成されます(デフォルト値: 'hlinkcfg')
-i テキスト・ホストリンク環境設定ファイル名	テキスト形式のホストリンク環境設定ファイル名を指定します。必須オプションであり、ディレクトリーを指定しない場合、\${TMAXDIR}/configに設定されます。パスと一緒に指定することができます。ソース・ファイルが見つからない場合、警告メッセージが出力されます
-c コメント処理文字	テキスト形式の環境設定ファイルのコメント文字を変更します。デフォルトでは「#」文字がコメント文字として使用されます

第4章 状態モニタリングAPIと構造体

本章では、Host-linkの状態をモニタリングするAPIと構造体について説明します。

4.1. API

Host-linkシステムはHost-linkの状態をモニタリングできるAPIを提供します。APIはHost-linkシステムが実行された後、Tmaxシステムのクライアント、サーバー・プログラムおよびその他の一般プログラムで使用できます。

以下は、Host-linkの状態をモニタリングするためのAPI一覧です。

API	説明
tpgethlinksvr	サーバー・タイプに該当するHost-linkシステムのプロセス数を表示します
tpgethlinkproc	サーバー・タイプに該当するHost-linkプロセスの情報を表示します
tpgethlinkluinfo	Host-linkシステムに属するサーバーのLU情報を表示します
tpgethlinkssinfo	Host-linkシステムに属するLUのセッション情報を表示します

4.1.1. tpgethlinksvr

サーバー・タイプに該当するHost-linkシステムのプロセス数を表示する関数です。関数によって取得したHost-linkの共有メモリー情報を、tpgethlinkproc、tpgethlinkluinfo、tpgethlinkssinfoの3つのAPIで使います。

● プロトタイプ

```
int tpgethlinksvr(int shmkey, int svrtype)
```

● パラメータ

パラメータ	説明
shmkey	Host-linkの共有メモリー・キーの値です
svrtype	サーバー・タイプを設定します – SVRTYPE_LU0, SVRTYPE_LU62S, SVRTYPE_LU62R, SVRTYPE_CTG, SVRTYPE_CSKL, SVRTYPE_IMSLSTN, SVRTYPE_IMSOTMA

- 戻り値

戻り値	説明
Host-linkのプロセス数	関数の呼び出しに成功した場合です
-1	関数の呼び出しに失敗した場合です

4.1.2. tpgethlinkproc

サーバー・タイプに該当するHost-linkプロセスの情報を表示する関数です。

- プロトタイプ

```
int tpgethlinkproc(int svrtype, HLPROCINFO *info)
```

- パラメータ

パラメータ	説明
svrtype	サーバー・タイプで、SVRTYPE_LU0、SVRTYPE_LU62S、SVRTYPE_LU62R、SVRTYPE_CTG、SVRTYPE_CSKL、SVRTYPE_IMSLSTN、SVRTYPE_IMSOTMAの中から1つを設定します
info	サーバー・プロセス情報が格納される構造体テーブルのバッファです。必ずtpgethlinksvr()で受信した数だけの構造体バッファを割り当てます

- 戻り値

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

4.1.3. tpgethlinkluinfo

Host-linkシステムに属するサーバーのLU情報を表示する関数です。この関数は内部でLU数の分だけのメモリーを割り当てているため、この関数の使用後には割り当てられたメモリーを必ず解除します。

- プロトタイプ

```
HLUINFO * tpgethlinkluinfo(char *svrname, int *lunum)
```

- パラメータ

パラメータ	説明
svrname	Host-linkシステムに属するサーバー名です
lunum	返されるLUの数です

- 戻り値

戻り値	説明
構造体テーブル	関数の呼び出しに成功した場合です。Host-linkサーバーのLU情報を格納している構造体テーブルを返します
NULL	関数の呼び出しに失敗した場合です

4.1.4. tpgethlinkssinfo

Host-linkシステムに属するLUのセッション情報を表示する関数です。この関数は内部でセッション数の分だけのメモリーを割り当ててため、この関数の使用後には割り当てられたメモリーを必ず解除します。

- プロトタイプ

```
HLSESSINFO * tpgethlinkssinfo(char *luname, int *ssnum)
```

- パラメータ

パラメータ	説明
luname	Host-linkシステムに属するLU名です
ssnum	返されるセッションの数です

- 戻り値

戻り値	説明
構造体テーブル	関数の呼び出しに成功した場合です。Host-linkLUのセッション情報を格納している構造体テーブルを返します
NULL	関数の呼び出しに失敗した場合です

4.2. 構造体

APIの説明のうちHost-linkで定義した構造体は、HLPROCINFO、HLLUINFO、HLSESSINFOです。

構造体	説明
HLPROCINFO	Host-linkサーバーのプロセス情報を格納している構造体です

構造体	説明
HLLUINFO	Host-linkサーバーのLU情報を格納している構造体です
HLSESSINFO	Host-linkLUのセッション情報を格納している構造体です

4.2.1. HLPROCINFO

HLPROCINFOは、Host-linkサーバーのプロセス情報を格納している構造体です。

```
struct HLPROCINFO
{
    int svrno;
    int msgsize;
    int function;
    int cpc;
    int buffering;
    int innum;
    int outnum;
    int line_status;
    char svrname[20];
    char linkname[8];
    char trxid[4];
    char svrlist[128];
}
```

メンバー	説明
int svrno	SERVERセクションのTMAXSVRNOです
int msgsize	SERVERセクションのMSGSIZEです
int function	SERVERセクションのFUNCIONです(DPL(1), DTP(2))
int cpc	SERVERセクションのCPCです
int buffering	SERVERセクションのBUFFERINGです(YES(1), NO(0))
int innum	SERVERセクションのSESSIONです
int outnum	未使用メンバーです
int line_status	該当サーバーの回線状態です(Active(1), Inactive(0))
char svrname[20]	SERVERセクションのServer Nameです
char linkname[8]	SERVERセクションのLINKNAMEです
char trxid[4]	SERVERセクションのTRXIDです
char svrlist[128]	SERVERセクションのSVRLISTです

4.2.2. HLLUINFO

HLLUINFOは、Host-linkサーバーのLU情報を格納している構造体です。

```
struct HLLUINFO
{
    char luname[8];
    char wsname[8];
    char lutype[12];
    char svcname[16];
    char tpname[64];
    char pluname[16];
    char modename[8];
    int status;
    int send;
    int direction;
    int session;
    int available;
    int count;
}
```

メンバー	説明
char luname[8]	LUINFOセクションのLUNAMEです
char wsname[8]	LU処理中のWSNAMEです
char lutype[12]	LUINFOセクションのLUTYPEです
char svcname[16]	LU処理中のサービス名です
char tpname[64]	LUINFOセクションのTPNAMEです
char pluname[16]	LUINFOセクションのPLUNAMEです
char modename[8]	LUINFOセクションのMODENAMEです
int status	LU状態です(LU-LU(0x50), LU-SSCP(0x51), DOWN(0x52), CSDN(0x53), ACTLU(0x54), INACTLU(0x55), NSPE(0x56))
int send	LU 0ホストへのメッセージ送信可否です(NONE(0), SENT(1), TIMEOUT(2))
int direction	LUINFOセクションのDIRECTIONです(INBOUND(1), OUTBOUND(2))
int session	LUINFOセクションのSESSIONです
int available	LU6.2の使用可能なセクション数です
int count	LU処理件数です

4.2.3. HLSESSINFO

HLSESSINFOは、Host-link LUのセッション情報を格納している構造体です。

```

struct HLSESSINFO
{
    int status;
    char luname[8];
    char local_tp[16];
    unsigned char tp_id[8];
    char remote_tp[16];
    char pgmname[8];
    char wsname[8];
    unsigned int conv_id;
    time_t time;
}

```

メンバー	説明
int status	LU6.2セッションの状態です(READY(1), BUSY(2))
char luname[8]	LU6.2セッション名です
char local_tp[16]	LU6.2セッションのローカルTPNAMEです
unsigned char tp_id[8]	LU6.2セッションのローカルTPIDです
char remote_tp[16]	LU6.2セッションのリモートTPNAMEです
char pgmname[8]	LU6.2セッションのPGMNAMEです
char wsname[8]	LU6.2セッションのWSNAMEです
unsigned int conv_id	LU6.2セッションのConversation IDです
time_t time	LU6.2セッションのTimestampです

4.3. エラーコード

Host-linkのエラーコードは以下のとおりです。

エラーコード	説明
TPEOS	Host-linkで使用するメモリーを割り当てる過程で発生するエラーです
TPENOREADY	Host-linkシステムで管理する多くのセッションのうち使用可能なセッションがない場合に発生するエラーです
TPESVCERR	Host-linkプロセスでホストにデータを送信する際に発生するエラーです
TPESVCFAIL	Host-linkプロセスでホストへのデータを受信する際に発生するエラーです
TPEBADDESC	Host-linkプロセスでホストにデータを送信後にホストからのセンスコードが送信された場合に発生するエラーです
TPETIME	Host-linkプロセスでホストにデータを送信後に指定された時間が超過した場合に発生するエラーです

エラーコード	説明
TPEPROTO	Host-linkプロセスで管理するLinkセッションが有効でない場合に発生するエラーです

付録 A. 環境設定の例

本章では、SNA LU 0とSNA LU 6.2の環境設定の例について説明します。

A.1. SNA LU 0

以下は、SNA LU 0プロセスを実行するのに必要な全体環境設定の例です。

```
*HOSTLINK
aaa          SHMKEY=98765,
              TMAX=Y

*SERVER
snalu0gw00   TMAXSVRNAME=snalu0gw00,
              TMAXSVRNO=0,
              SVRTYPE=LU0,
              HOSTID="X3",
              MSGSIZE=8192,
              TIMEWAIT=WAIT,
              LINKDOWN_TIMEOUT=60,
              SESSION_TIMEOUT=300,
              BUFFERING=YES,
              INBOUNDLU=DEDICATE,
              BIDLU=DEDICATE,
              LINKNAME=AAA

*SERVICE
snalu0gw00   BIDSVCNAME=BIDSVC,
              BIDSVCPOS=1,
              BIDSVCsize=10,
              ROPSVCNAME=BIDSVC,
              ROPSVCPOS=1,
              ROPSVCsize=10,
              OUTSVCNAME=OUTBOUND,
              OUTSVCPOS=1,
              OUTSVCsize=10,
              OUTERRPOS=20,
              OUTERRVAL="0001",
              RLYSVCNAME=BIDSVC,
              RLYSVCPOS=1,
              RLYSVCsize=10,
              TCLSVCNAME=BIDSVC,
```

```

TCLSVCSVCPOS=1,
TCLSVCSVCsize=10

*LOGGING
snalu0gw00      LOGPATH=aaa,
                  LOGTYPE=HEX,
                  LOGSVC=AAA

*CONVERSION
snalu0gw00      INBOUND=YES,
                  INBOUND_SKIP=50,
                  OUTBOUND=YES,
                  OUTBOUND_SKIP=50

*DUMMYDATA
snalu0gw00      ZEROSIZE=YES,
                  DUMMYPOS=1,
                  DUMMYVAL="xxxx"

*USERHEADER
snalu0gw00      HSIZE=20,
                  SVCPOS=1,
                  SVCsize=10,
                  LUTYPEPOS=2,
                  LUTYPEsize=6,
                  LUNAMEPOS=2,
                  LUNAMEsize=8,
                  WSNAMEPOS=10,
                  WSNAMEsize=8,
                  ERRcodePOS=1,
                  ERRcodeSIZE=4,
                  TIMEWAITPOS=1

*LUINFO
TLU015          svrname=snalu0gw00,
                  HOSTID=I9,
                  WSNAME="A000001",
                  LUKIND=INBOUND,
                  LUTYPE=i
TLU016          svrname=snalu0gw00,
                  HOSTID=I9,
                  WSNAME="A000002",
                  LUKIND=OUTBOUND,
                  LUTYPE=BID
TLU017          svrname=snalu0gw00,
                  HOSTID=I9,
                  WSNAME="A000002",

```

```

                LUKIND=OUTBOUND,
                LUTYPE=NORMAL
TLU018          svrname=snalu0gw00,
                HOSTID=I9,
                WSNAME="A000002",
                LUKIND=OUTBOUND,
                LUTYPE=ROP
TLU019          svrname=snalu0gw00,
                HOSTID=I9,
                WSNAME="A000002",
                LUKIND=OUTBOUND,
                LUTYPE=ROP

```

A.2. SNA LU 6.2

以下は、SNA LU 6.2プロセスを実行するのに必要な全体環境設定の例です。

```

*HOSTLINK
aaa              SHMKEY=98765,
                  TMAX=Y

*SERVER
sna62rgw00       TMAXSVRNAME=sna62rgw00,
                  TMAXSVRNO=0,
                  SVRTYPE=LU62R,
                  MSGSIZE=8192,
                  REPSND=Y,
                  LINKNAME=AAA
sna62sgw00       TMAXSVRNAME=sna62sgw00,
                  TMAXSVRNO=0,
                  SVRTYPE=LU62S,
                  MSGSIZE=8192,
                  FUNCTION=DPL,
                  LINKNAME=AAA

*SERVICE
sna62rgw00       OUTSVCNAME=BIDSVC,
                  OUTSVCPOS=1,
                  OUTSVCsize=10
sna62sgw00       RLVSVCNAME=BIDSVC,
                  RLVSVCPOS=1,
                  RLVSVCsize=10,
                  TCLSVCNAME=BIDSVC,
                  TCLSVCPOS=1,
                  TCLSVCsize=10

```

```

*LOGGING
sna62rgw00          LOGPATH=aaa ,
                    LOGTYPE=HEX ,
                    LOGSVC=AAA
sna62sgw00          LOGPATH=aaa ,
                    LOGTYPE=HEX ,
                    LOGSVC=AAA

*CONVERSION
sna62rgw00          INBOUND=YES ,
                    INBOUND_SKIP=50 ,
                    OUTBOUND=YES ,
                    OUTBOUND_SKIP=50
sna62sgw00          INBOUND=YES ,
                    INBOUND_SKIP=50 ,
                    OUTBOUND=YES ,
                    OUTBOUND_SKIP=50

*DUMMYDATA
sna62rgw00          DUMMYPOS=0 ,
                    DUMMYVAL="xxxx"
sna62sgw00          DUMMYPOS=0 ,
                    DUMMYVAL="xxxx"

*USERHEADER
sna62rgw00          HSIZE=20
sna62sgw00          HSIZE=20

*LUINFO
TLU015              svrname=sna62rgw00 ,
                    FQPLUNAME="KMTNET.CICSD1" ,
                    SESSION=20
TLU016              svrname=sna62rgw00 ,
                    FQPLUNAME="KMTNET.CICSD2" ,
                    SESSION=20
TLU017              svrname=sna62rgw01 ,
                    FQPLUNAME="KMTNET.CICSD3" ,
                    SESSION=20
TLU018              svrname=sna62sgw00 ,
                    FQPLUNAME="KMTNET.CICSD4" ,
                    TPNAME=A000001 ,
                    MODENAME=IBMRDB1 ,
                    SYNCMODE=NONE ,
                    SESSION=20
TLU019              svrname=sna62sgw00 ,
                    FQPLUNAME="KMTNET.CICSD5" ,
                    TPNAME=A000002 ,

```

TLU020

```
MODENAME=IBMRDB1,  
SYNCMODE=NONE,  
SESSION=20  
svrname=sna62sgw01,  
FQPLUNAME="KMTNET.CICSD6",  
TPNAME=A000003,  
MODENAME=IBMRDB1,  
SYNCMODE=NONE,  
SESSION=20
```


付録 B. ユーザー関数

本章では、SNA LU 0とSNA LU 6.2で使用するユーザー関数について説明します。

B.1. SNA LU 0

ヘッダー・ファイル

以下は、ユーザー関数を使用するためのヘッダー・ファイルです。

<lu0gw_prototype.h>

```
#ifndef _LU0GW_PROTOTYPE_H_
#define _LU0GW_PROTOTYPE_H_

#ifdef _WIN32
int get_msg_info(char *header, char *snddata, char *rcvdata);
#endif

#endif /* _LU0GW_PROTOTYPE_H_ */
```

ソースコード・ファイル

以下は、ユーザー関数を使用するためのソースコード・ファイルです。

<custom.c>

```
#include <stdio.h>
#include <fcntl.h>
#include <string.h>
#include <sys/types.h>

#include "lu0gw_prototype.h"

/* -----
get_msg_info : ホストから応答データを受信すると呼び出されます。
header : ユーザー・ヘッダー, NULLチェックが必要
```

```

snddata : ホスト送信データ
rcvdata : ホスト受信データ
return : 0より大きい場合、正常にリターン
         0の場合、ホスト受信データを廃棄 (NOREPLY処理)
         -1の場合、ホスト受信データを廃棄 (引き続きメッセージを待機)

----- */
int get_msg_info(char *header, char *snddata, char *rcvdata)
{
    /* 転送データがない場合: すでにタイムアウトになっている */
    if (snddata == NULL)
        return 1;          /* must 1 return */

    return 1;
}

```

B.2. SNA LU 6.2

ヘッダー・ファイル

以下は、ユーザー関数を使用するためのヘッダー・ファイルです。

<lu62sgw_prototype.h>

```

#ifndef _LU62SGW_PROTOTYPE_H_
#define _LU62SGW_PROTOTYPE_H_

#ifdef _WIN32
int get_wsname(char *uhead, char *wsname);
int get_msg_info(char *data, int len);
int put_msg_info(char *data, int len, char *luname);
int get_msg_info2(TPGWINFO_T *gwinfo, char *data, int len);
int put_msg_info2(TPGWINFO_T *gwinfo, char *data, int len, char *luname);
#endif

#endif /* _LU62SGW_PROTOTYPE_H_ */

```

ソースコード・ファイル

以下は、ユーザー関数を使用するためのソースコード・ファイルです。

<custom.c>

```

#include <stdio.h>
#include <fcntl.h>
#include <string.h>
#include <sys/types.h>
#include <usrinc/atmi.h>
#include <usrinc/hlinkapi.h>

#include "lu62sgw_prototype.h"

/* get_msg_info() return value */
#define DATA_INCOMPLETE      0
#define DATA_COMPLETE        1

/* -----
   get_msg_info2/put_msg_info2を使用したい場合、
   _tmax_custom_version = 2に変更
   ----- */
int _tmax_custom_version = 0;

/* -----
   get_wsname : ユーザー・ヘッダーの中からホストにサービスを要求したクライアントの
                 名前を返します。ホストにデータを送信前に呼び出されます。
   uhead: ユーザー・ヘッダーのデータ
   wsname : クライアントの名前が保存されるバッファ
   ----- */
int get_wsname(char *uhead, char *wsname)
{
    if (uhead == NULL)
        return 1;

    return 1;
}

/* -----
   get_msg_info : ホストの受信データをユーザーが処理できる関数です。
                   会話型モードの場合は連続メッセージの終わりであるのかを返す
                   必要があります。ホストの受信データはEBCDICコードです。
                   ホストでDeallocを実行した場合はこの関数で
                   DATA_INCOMPLETEを返したとしてもDATA_COMPLETEと
                   処理します。
   data: ホストの受信データ
   len : ホストの受信データ長
   return : 受信の完了(DATA_COMPLETE), 受信の未完了(DATA_INCOMPLETE)
   ----- */
int get_msg_info(char *data, int len)

```

```

{
    return DATA_COMPLETE;
}

/* -----
put_msg_info : ホストの送信前にユーザーが処理できる関数です。
data: ホストの送信データ
len : ホストの送信データ長
luname : ホスト送信LU名
return : 成功するとホストに送信する実際の長さをリターン,実際の値-1を
返すとホストに送信せず、エラーをクライアントに返す
----- */
int put_msg_info(char *data, int len, char *luname)
{
    return len;
}

/* -----
get_msg_info2 : ホストの受信データをユーザーが処理できる関数です。
会話型モードの場合は連続メッセージの終わりであるのかを返す
必要があります。ホストの受信データはEBCDICコードです。
ホストでDeallocを実行した場合はこの関数で
DATA_INCOMPLETEを返したとしてもDATA_COMPLETEと
処理します。
gwinfo: TPGWINFO_T情報の構造体
data: ホストの受信データ
len : ホストの受信データ長
return : 受信の完了(DATA_COMPLETE),受信の未完了(DATA_INCOMPLETE)
----- */
int get_msg_info2(TPGWINFO_T *gwinfo, char *data, int len)
{
    return DATA_COMPLETE;
}

/* -----
put_msg_info2 : ホストの送信前にユーザーが処理できる関数です。
gwinfo: TPGWINFO_T情報の構造体
data: ホストの送信データ
len : ホストの送信データ長
luname : ホスト送信LU名
return : 成功するとホストに送信する実際の長さをリターン,実際の値-1を
返すとホストに送信せず、エラーをクライアントに返す
----- */
int put_msg_info2(TPGWINFO_T *gwinfo, char *data, int len, char *luname)

```

```
{  
    return len;  
}
```


索引

B

BIDおよびROP, 46
BIDセッション, 6

C

CLH, 11
CONVERSIONセクション
 Mapファイル, 29
CPC, 11

D

Dedicate方式, 5
DPL方式, 2, 6
DTP方式, 2, 6

H

hkcf1, 48
HLLUINFO, 53
HLPROCINFO, 52
HLSESSINFO, 53
Host-linkの定義, 1
Host-link全体の構造, 3
Host-link環境設定ファイル, 12
 CONVERSIONセクション, 27
 DUMMYDATAセクション, 31
 HOSTLINKセクション, 13
 LOGGINGセクション, 25
 LUINFOセクション, 37
 SERVERセクション, 13
 SERVICEセクション, 20
 USERHEADERセクション, 32

P

Pool方式, 5

R

ROPセッション, 7

S

SNA LU 0, 2
SNA LU 6.2, 2
struct tpgwinfo, 36

T

Tmaxサービス・モジュール, 3
tpgethlinkluinfo(), 50
tpgethlinkproc(), 50
tpgethlinkssinfo(), 51
tpgethlinksvr(), 49

あ

一般セッション, 6
アウトバウンド・サービスの処理手順, 45
インバウンド・サービスの処理プロセス
 tpcall, 40
インバウンド・サービスの処理手順
 tpacall, 40
 tpforward, 43
エラーコード
 TPEBADDESC, 54
 TPENOREADY, 54
 TPEOS, 54
 TPEPROTO, 55
 TPESVCERR, 54
 TPESVCFAIL, 54
 TPETIME, 54

か

構造体
 HLLUINFO, 53
 HLPROCINFO, 52
 HLSESSINFO, 53
カスタム・ゲートウェイ, 3
ゲートウェイ・ヘッダー, 36

さ

状態モニタリングAPI

tpgethlinkluinfo(), 50

tpgethlinkproc(), 50

tpgethlinkssinfo(), 51

tpgethlinksvr(), 49

セッションの運用, 20

セッション・マネージャー・モジュール, 4

セッション数の制御, 3

た

多重チャネルの使用, 3

な

内部構造, 3

は

ホスト・サービス・モジュール, 4