

Tmax プログラミングガイド (MultipleRM)

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 プログラミングガイド (MultipleRM)

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. トランザクション処理	1
1.2.1. XAサーバー	1
1.2.2. MultipleRMアプリケーション・サーバー	2
第2章 環境設定	3
2.1. Tmax環境設定	3
2.2. サーバーの使用方法	4
第3章 MultipleRMプログラムの例	5
3.1. TiberoとOracleを使用	5
3.1.1. プログラムの例	5
3.1.2. 環境設定とコンパイル	7
3.2. OracleとOracleを使用	10
3.2.1. プログラムの例	10
3.2.2. 環境設定とコンパイル	13
3.3. TiberoとTiberoを使用	16
3.3.1. プログラムの例	16
3.3.2. 環境設定とコンパイル	19
3.4. OracleとDB2を使用	22
3.4.1. プログラムの例	22
3.4.2. 環境設定とコンパイル	35
索引	39

図目次

[図 1.1]	XAサーバーのトランザクション処理	1
[図 1.2]	MultipleRMアプリケーション・サーバーのトランザクション処理	2

このガイドについて

対象読者

本書は、Tmax[®](以下、Tmax)サーバー・プロセスのうち、Tmax MultipleRMについての概要と機能および特性について説明しています。同書は、Tmax MultipleRMを用いて開発する開発者とシステム管理者を対象としています。

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基本知識
- Tmaxの基本概念
- Cプログラミングについての知識

制限事項

本書では、Multiple RMについてのみ説明しています。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3章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

MultipleRMの概要と関連する設定について説明します。

- 第2章: 環境設定

MultipleRMを使用するための環境設定とサーバーの使用方法について説明します。

- 第3章: MultipleRMプログラムの例

OracleとTiberoを使用するプログラムとOracleとOracleを使用するMultipleRMサーバーの例について説明します。

表記上の規約

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]	図の名前
[表1.1]	表の名前
AaBbCc123	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、MultipleRMの概要と関連する設定について説明します。

1.1. 概要

XAサーバーの場合、1つのサーバー・グループ内のサーバーは、サーバー・グループに設定した1つの同じRMにアクセスしてトランザクションを処理するのがTmaxの基本ポリシーです。そのため、1つのプロセスで複数のRMを使用しなければならない場合、1つのトランザクションとして処理することができませんでした。これらの問題を解消するため、1つのプロセスで複数のRM、あるいは同じRMであるが複数のインスタンスを使用するトランザクションを1つのトランザクションとして処理するよう、サーバー・ライブラリーの機能を拡張しました。MultipleRMをサポートするため、既存のTCS、UCSサーバー・ライブラリーを利用するので、機能をそのまま使用することができます。

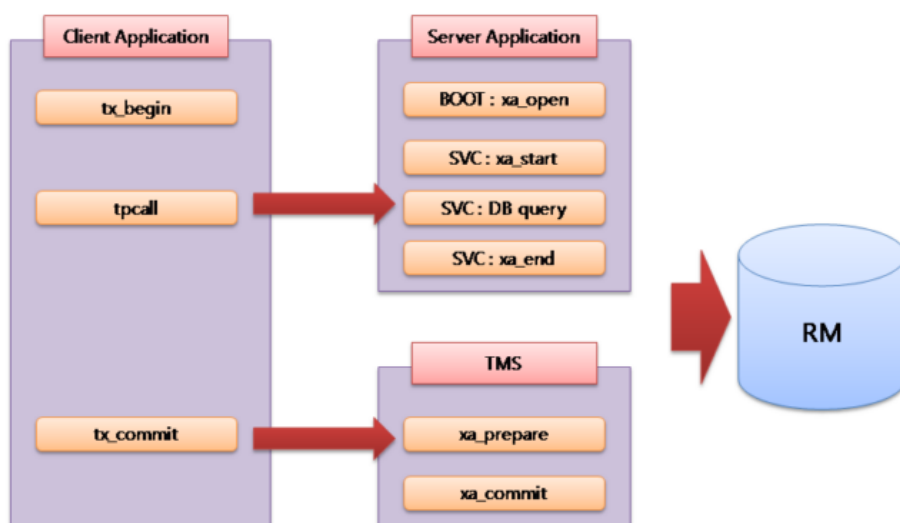
1.2. トランザクション処理

XAサーバーとMultipleRMアプリケーション・サーバーのトランザクション処理方法について説明します。

1.2.1. XAサーバー

以下は、XAサーバーのトランザクション処理プロセスです。

[図 1.1] XAサーバーのトランザクション処理



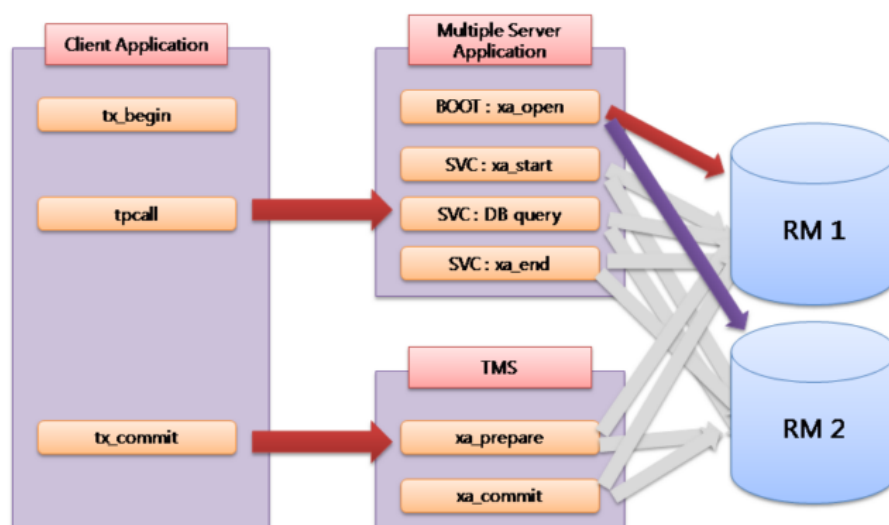
XAサーバーでは、起動時に`xa_open`、サービスを要求する際は、`xa_start` → 「RM作業」 → `xa_end`まで実行して、実際の`xa_prepare`と`xa_commit`は、サーバーが属しているサーバー・グループのTMSが処理しています。

1.2.2. MultipleRMアプリケーション・サーバー

MultipleRM設定を利用して開発したXAサーバーは、複数のRMIに接続してトランザクションを試みます。

以下は、MultipleRMアプリケーション・サーバーのトランザクション処理プロセスです。

【図 1.2】 MultipleRMアプリケーション・サーバーのトランザクション処理



起動時に登録したすべてのRMIに`xa_open`を実行します。サービスを要求するとき、CLHでRMIの数に応じてxidを発行し、サーバーでは発行されたxidを使用してすべてのRMIに`xa_start`を実行します。サービスを終了するときは、すべてのRMIに`xa_end`処理を行います。それ以降の作業は以前と同様、TMSが`xa_prepare`と`xa_commit`を実行します。

MultipleRMの制約

MultipleRMは、以下のような制約があります。

- COUSIN設定をサポートしていません。
- 以下の関数をサポートしていません。
 - `tpsuspendtx`
 - `tpresetmetx`
- 対応するデータベースは、Tibero、Oracle、DB2です。

第2章 環境設定

本章では、MultipleRMを使用するためのTmaxの環境設定とサーバーの使用方法について説明します。

2.1. Tmax環境設定

MultipleRMサーバーは複数のRMをサポートするので、1つのサーバー・グループを1つのRMに対応させる既存の設定では表現できません。MultipleRMサーバーを利用するには、SVGTYPE項目に「MTMAX」と「STMAX」を追加し、SVGLIST項目とRMID項目を追加します。

MTMAXはSTMAXサーバー・グループの集まりの代表的なサーバー・グループです。MTMAXに属しているサーバーはMultipleRMサーバーとなり、SVGLISTに存在するSTMAXのRMに対する処理が可能となります。

たとえば、TiberoとOracleが同じプロセスで動作する必要があり、かつ、一緒にトランザクションに含まれる必要がある場合、まず、STMAXサーバー・グループとしてTiberoとOracleを設定します。その後、MTMAXサーバー・グループを作成してSVGLISTにSTMAXサーバー・グループを設定し、サーバーにはMTMAXサーバー・グループに属しているサーバーを設定すると、MultipleRMサーバーの設定が可能となります。

以下は、前述した環境設定の例です。

```
*SVRGROUP
svg_s1      NODENAME = hahehihol,
            DBNAME = ORACLE,
            OPENINFO="Oracle_Xa+SqlNet=KOR04+Acc=P/scott/tiger+SesTm=60",
            TMSNAME = tms_ora,
            SVGTYPE = STMAX

svg_s2      NODENAME = hahehihol,
            DBNAME = TIBERO,
            OPENINFO = "TIBERO_XA:user=scott,pwd=tiger, sestm=60,db=kor01_tibero",

            TMSNAME = tms_tbr,
            SVGTYPE = STMAX

svgm1      NODENAME = hahehihol,
            SVGLIST = "svg_s1,svg_s2",
            SVGTYPE = MTMAX
```

以下は、STMAX関連の設定項目であり、必須設定項目です。

項目	説明
NODENAME	使用するノード名を設定します
DBNAME	該当するRM名を示します。現在は、「TIBERO」、「ORACLE」、「DB2_STATIC」のみサポートしています
OPENINFO	STMAXはXAサーバー・グループである必要がありますので、該当するOPENINFOを入力してください
TMSNAME	サーバー・グループのデータベースを管理するTMSプロセス名を定義します
SVGTYPE	必ずSTMAXで入力してください
RMID	1つのMTMAXに属しているSTMAXの中で同じRMを使用するものが存在する場合、インスタンスを区別するために異なるRMIDを入力します。同じRMを使用しない場合は、値を入力する必要はありません。 上記の例では、同じRMを使用しないため、RMIDを設定していません。RMIDの設定例については、 3.2.2節「環境設定」 を参照してください

以下は、MTMAX関連の設定項目であり、必須設定項目です。

項目	説明
NODENAME	使用するノード名を設定します
SVGLIST	含まれているSTMAXサーバー・グループの名前をカンマ(,)区切って示します
SVGTYPE	必ず「MTMAX」で設定します

2.2. サーバーの使用方法

TCS、UCSとMultipleRMサーバーの使用方法は同じです。TCS、UCSで使用していたいくつかの関数を除いて、すべてのAPIを使用することが可能であり、プログラムの構造も同様です。

ただし、サーバーに応じて、以下のようにプログラムを構成する要素が変わります。

- TCSとUCS方式のサーバー

TCSとUCSをビルドするためには、次のように構成します。

```
ユーザーコード + libsvr(ucs) + db_stub(nodb) library
```

- MultipleRMサーバー

MultipleRMサーバーの場合は、次のように構成します。

```
ユーザーコード + libsvr(ucs) + libnodb
```

第3章 MultipleRMプログラムの例

本章では、TCSでのMultipleRMプログラムの例と環境設定およびコンパイルのためのMakefileの例について説明します。

3.1. TiberoとOracleを使用

サーバー・ライブラリーを利用して、TCSでTiberoとOracleを使用することができます。本節では、TiberoとOracleを使用するプログラムの例と環境設定およびコンパイルのためのMakefileの例について説明します。

3.1.1. プログラムの例

以下は、TiberoとOracleと一緒に使用するTCSサンプル・プログラムです。埋め込みSQLを使用してサーバーを作成し、各データベース・ベンダーが提供するプリコンパイラを使ってコンパイルして構成することができます。

サービスの実行

以下は、サービスを実行するプログラムのソースです。

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

extern int oracle_call(TPSVCINFO *msg);
extern int tiberocal (TPSVCINFO *msg);

MDB_SVC1(TPSVCINFO *msg)
{
    int ret;
    int fail = 0;

    ret = oracle_call(msg);
    if (ret < 0) {
        fail = 1;
    }

    ret = tiberocal(msg);
    if (ret < 0) {
        fail = 1;
    }
}
```

```

    }

    if ( fail ) {
        tpreturn(TPFAIL,0,(char *)msg->data, msg->len,0);
    }

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

```

Oracleの呼び出し関数

以下は、Oracleを呼び出す関数を使用したプログラムの例です。

```

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

EXEC SQL include sqlca.h;

EXEC SQL begin declare section;
    char h_addr[100];
EXEC SQL end declare section;

int oracle_call(TPSVCINFO *msg)
{
    int    i;

    strcpy(h_addr, msg->data);

    EXEC SQL INSERT
        INTO account( account_id, address, phone )
        VALUES ( tmax_seq.nextval,:h_addr, '007-1234' );

    if ( sqlca.sqlcode != 0 ){
        printf( "insert failed sqlcode = %d\n",sqlca.sqlcode );
        tpreturn( TPFAIL, -1, NULL, 0, 0 );
    }

    printf("ORACLE insert Success\n");

    return 0;
}

```


Tiberoの呼び出し関数

以下は、Tiberoを呼び出す関数を使用したプログラムの例です。

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

EXEC SQL include sqlca.h;

EXEC SQL begin declare section;
    char h_addr[100];
EXEC SQL end declare section;

int tibero_call(TPSVCINFO *msg)
{
    strcpy(h_addr, msg->data);

    EXEC SQL INSERT
        INTO account( account_id, address, phone )
        VALUES ( tmax_seq.nextval, h_addr, '007-1234' );

    if ( sqlca.sqlcode != 0 ){
        printf( "tibero insert failed sqlcode = %d\n", sqlca.sqlcode );
        tpreturn( TPFAIL, -1, NULL, 0, 0 );
    }

    printf("TIBERO insert Success\n");

    return 0;
}
```

3.1.2. 環境設定とコンパイル

TiberoとOracleを使用するMultipleRMの環境設定とコンパイルの例について説明します。

環境設定

以下は、MultipleRMサーバーの環境設定ファイルの例です。

```
*DOMAIN
domain          SHMKEY    = 81522, MAXUSER = 10000,
                  MINCLH   = 1,  MAXCLH   = 1,
                  TPORTNO  = 11500,  BLOCKTIME = 30,
                  MAXSACALL = 1024, MAXCACALL = 1024,
                  MAXMTMAX  = 3,  MAXSTMAX  = 10
```

```

*NODE
node1      TMAXDIR   = "/home/tmax",
           APPDIR    = "/home/tmax/appbin",
           PATHDIR   = "/home/tmax/path",
           TLOGDIR   = "/home/tmax/log/tlog",
           ULOGDIR   = "/home/tmax/log/ulog",
           SLOGDIR   = "/home/tmax/log/slog"

*SVRGROUP
svg_s1     NODENAME = node1,
           DBNAME    = ORACLE,
           OPENINFO  = "Oracle_Xa+SqlNet=TMAX+Acc=P/scott/tiger+SesTm=60",
           TMSNAME   = tms_ora,
           SVGTYPE   = STMAX

svg_s2     NODENAME = node1,
           DBNAME    = TIBERO,
           OPENINFO  = "TIBERO_XA:user=scott,pwd=tiger,sestm=60,db=TMAX",
           TMSNAME   = tms_tbr,
           SVGTYPE   = STMAX

```

Makefile

Makefile内のLIBS(TMAXライブラリー)には、**-lnodb**が含まれている必要があります。

以下は、32ビットLinuxでMultipleRMサーバー・プログラムをコンパイルするためのMakefileの例です。

```

# Oracle Env
ORALIBDIR = /home/OraHome1/lib/
ORALIB = -lclntsh `cat /home/OraHome1/lib/ldflags` `cat
/home/OraHome1/lib/sysliblist` -ldl -lm

# Tiberio Env
TBRLIB = -ltbxa -ltbertl
TBRLIBDIR = $(TB_HOME)/client/lib
TBRLIBDIR = $(TB_HOME)/client/include

$ Server build
TARGET = $(COMP_TARGET)
APOBJS = $(TARGET).o
AP_ORA_OBJS = $(TARGET)_ora.o
AP_TBR_OBJS = $(TARGET)_tbr.o
NSDLOBJ = $(TMAXDIR)/lib/sdl.o

LIBS = -lnsl -lsvr -lnodb

```

```

OBSJ      = $(AP_ORA_OBJS) $(AP_TBR_OBJS) $(SVCTOBJ) $(APOBJS)
SVCTOBJ   = $(TARGET)_svctab.o

CFLAGS    = -O -I$(TMAXDIR)

APPDIR    = $(TMAXDIR)/appbin
SVCTDIR   = $(TMAXDIR)/svct
TMAXLIBDIR = $(TMAXDIR)/lib

#
.SUFFIXES : .c

.c.o:
    echo $(OBSJ)
    $(CC) $(CFLAGS) -c $<

#
# server compile
#
all: $(TARGET)

$(TARGET):$(OBSJ)
    echo $(OBSJ)
    $(CC) $(CFLAGS) -L$(TMAXLIBDIR) -o $(TARGET) -L$(TBRLIBDIR) -L$(ORALIBDIR)
$(ORALIB) $(TBRLIB) $(OBSJ) $(LIBS) $(NSDLOBJ)
    mv $(TARGET) $(APPDIR)/.
    rm -f $(OBSJ)

$(AP_ORA_OBJS):
    echo $(OBSJ)
    proc iname=$(TARGET)_ora.pc include=$(TMAXDIR)
    $(CC) $(CFLAGS) -c $(TARGET)_ora.c

$(AP_TBR_OBJS):
    echo $(OBSJ)
    tbpc iname=$(TARGET)_tbr.tbc include=$(TMAXDIR)
    $(CC) $(CFLAGS) -I$(TBRINCDIR) -c $(TARGET)_tbr.c

$(APOBJS):
    echo $(OBSJ)
    $(CC) $(CFLAGS) -c $(TARGET).c

$(SVCTOBJ):
    echo $(OBSJ)
    cp -f $(SVCTDIR)/$(TARGET)_svctab.c .

```

```
touch ./${TARGET}_svctab.c
${CC} ${CFLAGS} -c ./${TARGET}_svctab.c

#
clean:
    -rm -f *.o core ${TARGET} ${TARGET}.lis
```

参考

OSに応じて、Makefileの内容が異なる場合があります。

3.2. OracleとOracleを使用

サーバー・ライブラリーを利用して、TCSでOracleとOracleを使用することができます。本節では、OracleとOracleを使用するプログラムの例と環境設定およびコンパイルのためのMakefileの例について説明します。

3.2.1. プログラムの例

以下は、2つのRMがすべてOracleであるが、インスタンスが異なる場合に1つのサーバーで処理する例です。

同じOracleクライアント・ライブラリーを使用するので、ユーザー・コードでクエリーを実行する場合は、実行するインスタントをユーザーが指定する必要があります。Oracleでは、xa_openの場合、ユーザー・コードでxaoSvcCtx、xaoEnvなどのOCI関数を使用して、該当するインスタント・コネクションを取得して指定することができます。

サービスの実行

以下は、サービスを実行するプログラムのソースです。

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

extern int oracle_call(TPSVCINFO *msg);

MDB_SVC2(TPSVCINFO *msg)
{
    int ret;
    int fail = 0;

    ret = oracle_call(msg);
    if (ret < 0) {
        fail = 1;
    }
}
```

```

    if ( fail ) {
        tpreturn(TPFAIL,0,(char *)msg->data, msg->len,0);
    }

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

```

Oracleの呼び出し関数

以下は、Oracleを呼び出す関数を使用するプログラムの例です。

```

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

static void checkerr(errhp, status)
OCIError *errhp;
sword status;
{
    text errbuf[512];
    sb4 errcode;

    switch (status)
    {
        case OCI_SUCCESS:
            break;
        case OCI_SUCCESS_WITH_INFO:
            break;
        case OCI_NEED_DATA:
            break;
        case OCI_NO_DATA:
            break;
        case OCI_ERROR: /* get the error back and display on the screen */
            (void) OCIErrorGet ((dvoid *) errhp, (ub4) 1, (text *) NULL, &errcode,

                                errbuf, (ub4) sizeof(errbuf), (ub4)
OCI_HTYPE_ERROR);
            (void) printf("Error - %s\n", errbuf);
            break;
        case OCI_INVALID_HANDLE:
            break;
        case OCI_STILL_EXECUTING:
            break;
        case OCI_CONTINUE:

```

```

        break;
    default:
        break;
    }
}

int oracle_call(TPSVCINFO *msg)
{
    int i;
    OCIEnv *envhp;
    OCIServer *srvhp;
    OCIError *errhp;
    OCISvcCtx *svchp;
    OCISession *usrhp;
    OCISmt *stmthp;
    dvoid *tmp;
    char *sql = {"INSERT "
        "INTO hahehiho_account( account_id, address, phone ) "
        "VALUES ( skt_seq.nextval,'asdf', '007-1234' )"};

    char *sql2 = {"INSERT "
        "INTO hahehiho_account( account_id, address, phone ) "
        "VALUES ( skt_seq.nextval,'fdsa', '700-1234' )"};

    svchp = xaoSvcCtx("A");
    envhp = xaoEnv("A");
    printf("oracle call A octxt[%d]\n", svchp);

    OCIHandleAlloc( (dvoid *) envhp, (dvoid **) &errhp, (ub4) OCI_HTYPE_ERROR,
        52, (dvoid **) &tmp);

    OCIHandleAlloc( (dvoid *) envhp, (dvoid **) &stmthp, (ub4) OCI_HTYPE_STMT,
        50, (dvoid **) &tmp);

    checkerr(errhp, OCISmtPrepare(stmthp, errhp, (text *)sql, (ub4)strlen(sql),
        (ub4) OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));

    checkerr(errhp, OCISmtExecute(svchp, stmthp, errhp, (ub4) 1, (ub4) 0,
        (OCISnapshot *) NULL, (OCISnapshot *) NULL,
        (ub4) OCI_DEFAULT));

    svchp = xaoSvcCtx("B");
    envhp = xaoEnv("B");
    printf("oracle call B octxt[%d]\n", svchp);

    OCIHandleAlloc( (dvoid *) envhp, (dvoid **) &errhp, (ub4) OCI_HTYPE_ERROR,
        52, (dvoid **) &tmp);

```

```

OCIHandleAlloc( (dvoid *) envhp, (dvoid **) &stmthp,
                (ub4) OCI_HTYPE_STMT, 50, (dvoid **) &tmp);

checkerr(errhp, OCISstmtPrepare(stmthp, errhp, (text *)sql2, (ub4)strlen(sql2),

                                (ub4) OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));

checkerr(errhp, OCISstmtExecute(svchp, stmthp, errhp, (ub4) 1, (ub4) 0,
                                (OCISnapshot *) NULL, (OCISnapshot *) NULL,
                                (ub4) OCI_DEFAULT));

return 0;
}

```

3.2.2. 環境設定とコンパイル

OracleとOracleを使用するMultipleRMの環境設定とコンパイルの例について説明します。

環境設定

以下は、MultipleRMサーバーの環境設定ファイルの例です。

```

*DOMAIN
domain      SHMKEY    = 81522, MAXUSER = 10000,
              MINCLH   = 1, MAXCLH   = 1,
              TPORTNO  = 11500, BLOCKTIME = 30,
              MAXSACALL = 1024, MAXCACALL = 1024,
              MAXMTMAX  = 3, MAXSTMAX  = 10

*NODE
node1       TMAXDIR   = "/home/tmax",
              APPDIR   = "/home/tmax/appbin",
              PATHDIR  = "/home/tmax/path",
              TLOGDIR  = "/home/tmax/log/tlog",
              ULOGDIR  = "/home/tmax/log/ulog",
              SLOGDIR  = "/home/tmax/log/slog"

*SVRGROUP
svg_s1      NODENAME  = node1,
              DBNAME   = ORACLE,
              OPENINFO = "Oracle_Xa+SqlNet=TMAX1+Acc=P/scott/tiger+SesTm=60+DB=A",

              TMSNAME  = tms_ora,
              SVGTYPE  = STMAX,
              RMID     = 0

```

```

svg_s2      NODENAME = node1,
            DBNAME = ORACLE,
            OPENINFO = "Oracle_Xa+SqlNet=TMAX2+Acc=P/scott/tiger+SesTm=60+DB=B",

            TMSNAME = tms_ora,
            SVGTYPE = STMAX,
            RMID = 1

svgm1      NODENAME = node1,
            SVGLIST = "svg_s2,svg_s1",
            SVGTYPE = MTMAX

*SERVER
sample SVGNAME = svgm1, LOGLVL = DEBUG4

*SERVICE
MDB_SVC2 SVRNAME = sample

```

Makefile

MakefileのTMAXLIBSには、**-lnodb**が含まれている必要があります。以下は、32ビットLinuxでMultipleRMサーバー・プログラムをコンパイルするためのMakefileの例です。

```

# Oracle Env
ORALIBDIR = /home/OraHome1/lib/
ORALIB = -lclntsh `cat /home/OraHome1/lib/ldflags` `cat
/home/OraHome1/lib/sysliblist` -ldl -lm

LIBS      = -lnsl -lsvrmrm

OBSJS     = $(AP_ORA_OBJS) $(AP_TBR_OBJS) $(SVCTOBJ) $(APOBJS)
SVCTOBJ   = $(TARGET)_svctab.o

CFLAGS    = -O -I$(TMAXDIR)

APPDIR    = $(TMAXDIR)/appbin
SVCTDIR   = $(TMAXDIR)/svct
TARGET    = $(COMP_TARGET)
APOBJS    = $(TARGET).o
AP_ORA_OBJS = $(TARGET)_ora.o
NSDLOBJ   = $(TMAXDIR)/lib/sdl.o

LIBS      = -lnsl -lsvr -lnodb

OBSJS     = $(AP_ORA_OBJS) $(SVCTOBJ) $(APOBJS)
SVCTOBJ   = $(TARGET)_svctab.o

```



```

CFLAGS = -O -I$(TMAXDIR) -I$(ORACLE_HOME)/include -I$(ORACLE_HOME)/rdbms/demo
-I$(ORACLE_HOME)/rdbms/public

APPDIR = $(TMAXDIR)/appbin
SVCTDIR = $(TMAXDIR)/svct
TMAXLIBDIR = $(TMAXDIR)/lib

#
.SUFFIXES : .c

.c.o:
    echo $(OBS)
    $(CC) $(CFLAGS) -c $<

#
# server compile
#
all: $(TARGET)

$(TARGET):$(OBS)
    $(CC) $(CFLAGS) -L$(TMAXLIBDIR) -o $(TARGET) -L$(ORALIBDIR) $(ORALIB) $(OBS)
$(LIBS) $(NSDLOBJ)
    mv $(TARGET) $(APPDIR)/.
    rm -f $(OBS)

$(AP_ORA_OBS):
    $(CC) $(CFLAGS) -c $(TARGET)_ora.c

$(APOBS):
    $(CC) $(CFLAGS) -c $(TARGET).c

$(SVCTOBJ):
    echo $(OBS)
    cp -f $(SVCTDIR)/$(TARGET)_svctab.c .
    touch ./$(TARGET)_svctab.c
    $(CC) $(CFLAGS) -c ./$(TARGET)_svctab.c

#
clean:
    -rm -f *.o core $(TARGET) $(TARGET).lis

```

参考

OSに応じて、Makefileの内容が異なる場合があります。

3.3. TiberοとTiberοを使用

サーバー・ライブラリーを利用して、TCSでTiberοとTiberοを使用することができます。本節では、TiberοとTiberοを使用するプログラムの例と環境設定およびコンパイルのためのMakefileの例について説明します。

3.3.1. プログラムの例

以下は、2つのRMがすべてTiberοであるが、インスタンスが異なる場合に1つのサーバーで処理する例です。

同じTiberοクライアント・ライブラリーを使用するので、ユーザー・コードでクエリーを実行する場合は、実行するインスタントをユーザーが指定する必要があります。Tiberοでは、ESQL文(EXEC SQL XA SET CONNECTION AT :conn_id)を使用して、該当するインスタント・コネクションを取得して指定することができます。

サービスの実行

以下は、サービスを実行するプログラムのソースです。

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

extern int SVC2301TX_1_tiberο(TPSVCINFO *msg);

SVC2301TX_1( TPSVCINFO *msg )
{
    int ret;
    int fail = 0;
    char *rcvbuf;

    rcvbuf=(char *)tpalloc("STRING", NULL, 4096);

    if(rcvbuf==NULL) {
        printf("tpalloc failed (rcvbuf) : %s\n", tpstrerror(tperrno));
        tpreturn(TPFAIL, -1, NULL, 0, 0);
    }

    ret = SVC2301TX_1_tiberο(msg);
    if (ret < 0) {
        fail = 1;
    }

    if ( fail ) {
        printf("SVC2301TX_1 FAIL\n");
    }
}
```

```

        tpreturn(TPFAIL,-1,NULL, 0,0);
    }

    strcpy(rcvbuf, "Multiple RM Insert Success");
    tpreturn( TPSUCCESS, 0, rcvbuf, strlen(rcvbuf), 0 );
}

```

Tiberoの呼び出し関数

以下は、Tiberoを呼び出す関数を使用するプログラムの例です。

```

#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>
#include <string.h>
#include <usrinc/atmi.h>
#include <usrinc/tmaxapi.h>
#include <usrinc/fbuf.h>
#include <usrinc/tx.h>
#include "../sdl/demo_sdl.h"
#include "../fdl/demo_fdl.h"

#define SQLOK          0
#define SQLDUP         1
#define SQLNOTFOUND 1403

EXEC SQL INCLUDE SQLCA.H;
EXEC SQL BEGIN DECLARE SECTION;
    int  h_empno;
    char h_ename[MAXLEN];
    char h_job[MAXLEN];
    char h_date[MAXLEN];
    float h_sal;
    int  h_count;
    char h_xid[100];
    char h_conn_id[MAXLEN];
EXEC SQL END DECLARE SECTION;

SVC2301TX_1_tibero( TPSVCINFO *msg )
{
    str  sndbuf;
    char *rcvbuf;
    char tmp[1024];
    TXINFO info;

    sndbuf = (str)msg->data;

```

```

rcvbuf=(char *)tpalloc("STRING", NULL, 4096);
if(rcvbuf==NULL) {
    printf("tpalloc failed (rcvbuf) : %s\n", tpstrerror(tperrno));
    tpreturn(TPFAIL, -1, NULL, 0, 0);
}

h_empno = h_sal = 0;
memset( h_ename, 0x00, sizeof( h_ename ) );
memset( h_job, 0x00, sizeof( h_job ) );
memset( h_date, 0x00, sizeof( h_date ) );
memset( tmp , 0x00, sizeof( tmp ) );
memset( h_xid , 0x00, sizeof( h_xid ) );
memset( h_conn_id, 0x00, sizeof( h_conn_id ) );

h_empno = sndbuf->empno;
h_sal   = sndbuf->sal;
strcpy( h_ename, sndbuf->ename );
strcpy( h_job , sndbuf->job );
strcpy( h_date , sndbuf->date );

if (tx_info(&info)==1) {
    sprintf(h_xid, "%x %x %x %x - %x %x %x %x",
        (unsigned char)info.xid.data[0],
        (unsigned char)info.xid.data[1],
        (unsigned char)info.xid.data[2],
        (unsigned char)info.xid.data[3],
        (unsigned char)info.xid.data[4],
        (unsigned char)info.xid.data[5],
        (unsigned char)info.xid.data[6],
        (unsigned char)info.xid.data[7]);
    printf("xid=[%s] \n", h_xid);
} else {
    printf("Not in transaction \n");
    tpreturn(TPFAIL, -1, NULL, 0, 0);
}

/* Set Default SQL CTX with conn_id */
strcpy(h_conn_id, "DB1");

EXEC SQL XA SET CONNECTION AT :h_conn_id;
if ( sqlca.sqlcode != SQLOK )
{
    sprintf(tmp, "[%s] emp Insert Fail", msg->name);
    strcpy(rcvbuf, tmp);
    printf("[%s] %d \n", rcvbuf, sqlca.sqlcode);
    tpreturn( TPFAIL, sqlca.sqlcode, rcvbuf, strlen(rcvbuf), 0 );
}

```

```

EXEC SQL INSERT INTO tmax( empno, ename, job, hiredate,sal, xid)
VALUES (:h_empno, :h_ename, :h_job, to_date(:h_date,'yymmdd'), :h_sal, :h_xid
);
if ( sqlca.sqlcode != SQLOK )
{
    sprintf(tmp, "[%s] TIBERO tmax Insert Fail", msg->name);
    strcpy(rcvbuf, tmp);
    printf("[%s] %d \n", rcvbuf, sqlca.sqlcode);
    tpreturn( TPFAIL, sqlca.sqlcode, rcvbuf, strlen(rcvbuf), 0 );
}
sleep(1);

/* Set Default SQL CTX with conn_id */
strcpy(h_conn_id, "DB2");

EXEC SQL XA SET CONNECTION AT :h_conn_id;
if ( sqlca.sqlcode != SQLOK )
{
    sprintf(tmp, "[%s] emp Insert Fail", msg->name);
    strcpy(rcvbuf, tmp);
    printf("[%s] %d \n", rcvbuf, sqlca.sqlcode);
    tpreturn( TPFAIL, sqlca.sqlcode, rcvbuf, strlen(rcvbuf), 0 );
}

EXEC SQL
INSERT INTO tmax( empno, ename, job, hiredate,sal, xid)
VALUES (:h_empno, :h_ename, :h_job, to_date(:h_date,'yymmdd'), :h_sal, :h_xid
);
if ( sqlca.sqlcode != SQLOK )
{
    sprintf(tmp, "[%s] TIBERO tmax Insert Fail", msg->name);
    strcpy(rcvbuf, tmp);
    printf("[%s] %d \n", rcvbuf, sqlca.sqlcode);
    tpreturn( TPFAIL, sqlca.sqlcode, rcvbuf, strlen(rcvbuf), 0 );
}

sprintf(tmp, "[%s] TIBERO tmax Insert Success", msg->name);
strcpy(rcvbuf, tmp);
return 0;
}

```

3.3.2. 環境設定とコンパイル

TiberoとTiberoを使用するMultipleRMの環境設定とコンパイルの例について説明します。

環境設定

以下は、MultipleRMサーバーの環境設定ファイルの例です。

```
*DOMAIN
domain      SHMKEY    = 81522, MAXUSER = 10000,
             MINCLH   = 1, MAXCLH = 1,
             TPORTNO  = 11500, BLOCKTIME = 30,
             MAXSACALL = 1024, MAXCACALL = 1024,
             MAXMTMAX  = 3, MAXSTMAX = 10

*NODE
node1       TMAXDIR   = "/home/tmax1/tmax",
             APPDIR    = "/home/tmax1/tmax/appbin",
             PATHDIR   = "/home/tmax1/tmax/path",
             TLOGDIR   = "/home/tmax1/tmax/log/tlog",
             ULOGDIR   = "/home/tmax1/tmax/log/ulog",
             SLOGDIR   = "/home/tmax1/tmax/log/slog"

node2       TMAXDIR   = "/home/tmax2/tmax",
             APPDIR    = "/home/tmax2/tmax/appbin",
             PATHDIR   = "/home/tmax2/tmax/path",
             TLOGDIR   = "/home/tmax2/tmax/log/tlog",
             ULOGDIR   = "/home/tmax2/tmax/log/ulog",
             SLOGDIR   = "/home/tmax2/tmax/log/slog"

*SVRGROUP
svg_s1      NODENAME  = node1,
             DBNAME    = TIBERO,

OPENINFO="TIBERO_XA:user=tibero,pwd=tmax,db=tibero1,Loose_Coupling=false,sestm=60,conn_id=DB1",

             TMSNAME   = tms_tbr,
             SVGTYPE    = STMAX,
             RMID       = 1
svg_s2      NODENAME  = node2,
             DBNAME    = TIBERO,

OPENINFO="TIBERO_XA:user=tibero,pwd=tmax,db=tibero2,Loose_Coupling=false,sestm=60,conn_id=DB2",

             TMSNAME   = tms_tbr,
             SVGTYPE    = STMAX,
             RMID       = 2

svgm1       NODENAME  = node1,
             SVGLIST    = "svg_s1, svg_s2",
             SVGTYPE    = MTMAX
```

```

*SERVER
svr2301TX      SVGNAME = svgm1, MIN=5, MAX=5, MAXRSTART=-1

*SERVICE
SVC2301TX_1    SVRNAME = svr2301TX, SVCTIME=5

```

Makefile

MakefileのTMAXLIBSには、**-lnodb**が含まれている必要があります。以下は、64ビットLinuxでMultipleRMサーバー・プログラムをコンパイルするためのMakefileの例です。

```

# Tiberio Env
TBINC = $(TB_HOME)/client/include
TBLIBDIR = $(TB_HOME)/client/lib
TBLIB = -ltbxa -ltbertl -ltbcli -lm -lpthread

# Server makefile
TARGET = $(COMP_TARGET)
APOBJS = $(TARGET).o
NSDLOBJ = $(TMAXDIR)/lib64/sdl.o
AP_TBR_OBJS = $(TARGET)_tbr.o

LIBS = -lsvr -lnsl -lnodb
OBJS = $(AP_TBR_OBJS) $(APOBJS) $(SVCTOBJ)
SVCTOBJ = $(TARGET)_svctab.o
CFLAGS = -O -I$(TMAXDIR)

APPDIR = $(TMAXDIR)/appbin
SVCTDIR = $(TMAXDIR)/svct
TMAXLIBDIR = $(TMAXDIR)/lib64

#
.SUFFIXES : .c

.c.o:
    $(CC) $(CFLAGS) -c $<

#
# server compile
#
all: $(TARGET)

$(TARGET): $(OBJS)
    echo $(ORALIBDIR)
    $(CC) $(CFLAGS) -L$(TMAXLIBDIR) -o $(TARGET) -L$(TBLIBDIR) $(TBLIB) $(OBJS)

```

```
$(LIBS) $(NSDLOBJ)
    mv $(TARGET) $(APPDIR)/.
    rm -f $(OBSJ)

$(AP_TBR_OBJS):
    echo $(OBSJ)
    ttpc iname=$(TARGET)_tbr.tbc include=$(TMAXDIR)
    $(CC) $(CFLAGS) -I$(TBINC) -c $(TARGET)_tbr.c

$(APOBJS): $(TARGET).c
    $(CC) $(CFLAGS) -c $(TARGET).c

$(SVCTOBJ):
    cp -f $(SVCTDIR)/$(TARGET)_svctab.c .
    touch ./$(TARGET)_svctab.c
    $(CC) $(CFLAGS) -c ./$(TARGET)_svctab.c

#
clean:
    -rm -f *.o core $(APPDIR)/$(TARGET)
```

参考

OSに応じて、Makefileの内容が異なる場合があります。

3.4. OracleとDB2を使用

サーバー・ライブラリーを利用して、TCSでOracleとDB2を使用することができます。本節では、OracleとDB2を使用するプログラムの例と環境設定およびコンパイルのためのMakefileの例について説明します。

3.4.1. プログラムの例

以下は、OracleとDB2と一緒に使用するTCSサンプル・プログラムです。埋め込みSQLを使用してサーバーを作成し、各データベース・ベンダーが提供するプリコンパイラを使ってコンパイルして構成することができます。

サービスの実行

以下は、サービスを実行するプログラムのソースです。

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



```

#include <usrinc/tx.h>

extern int SVC2301TX_1_oracle(TPSVCINFO *msg);
extern int SVC2301TX_1_db2(TPSVCINFO *msg);
extern int SVC2301TX_2_oracle(TPSVCINFO *msg);
extern int SVC2301TX_2_db2(TPSVCINFO *msg);
extern int SVC2301TX_3_oracle(TPSVCINFO *msg);
extern int SVC2301TX_3_db2(TPSVCINFO *msg);
extern int SVC2301TX_4_oracle(TPSVCINFO *msg);
extern int SVC2301TX_4_db2(TPSVCINFO *msg);

SVC2301TX_1( TPSVCINFO *msg )
{
    int ret;
    int fail = 0;
    char *rcvbuf;

    rcvbuf=(char *)tpalloc("STRING", NULL, 4096);
    if(rcvbuf==NULL) {
        printf("tpalloc failed (rcvbuf) : %s\n", tpstrerror(tperrno));
        tpreturn(TPFAIL, -1, NULL, 0, 0);
    }

    ret = SVC2301TX_1_oracle(msg);
    if (ret < 0) {
        fail = 1;
    }

    ret = SVC2301TX_1_db2(msg);
    if (ret < 0) {
        fail = 1;
    }

    if ( fail ) {
        printf("SVC2301TX_1 FAIL\n");
        tpreturn(TPFAIL,-1,NULL, 0,0);
    }

    strcpy(rcvbuf, "Multiple RM Insert Success");
    tpreturn( TPSUCCESS, 0, rcvbuf, strlen(rcvbuf), 0 );
}

SVC2301TX_2( TPSVCINFO *msg )
{
    int ret;
    int fail = 0;

```

```

char *rcvbuf;

rcvbuf=(char *)tpalloc("STRING", NULL, 4096);
    if(rcvbuf==NULL) {
        printf("tpalloc failed (rcvbuf) : %s\n", tpstrerror(tperrno));
        tpreturn(TPFAIL, -1, NULL, 0, 0);
    }

ret = SVC2301TX_2_oracle(msg);
if (ret < 0) {
    fail = 1;
}

ret = SVC2301TX_2_db2(msg);
if (ret < 0) {
    fail = 1;
}
if ( fail ) {
    printf("SVC2301TX_2  FAIL\n");
    tpreturn(TPFAIL,-1,NULL, 0,0);
}
strcpy(rcvbuf, "Multiple RM Update Success");
tpreturn( TPSUCCESS, 0, rcvbuf, strlen(rcvbuf), 0 );
}

SVC2301TX_3( TPSVCINFO *msg )
{
    int ret;
    int fail = 0;
    char *rcvbuf;

    rcvbuf=(char *)tpalloc("STRING", NULL, 4096);
        if(rcvbuf==NULL) {
            printf("tpalloc failed (rcvbuf) : %s\n", tpstrerror(tperrno));
            tpreturn(TPFAIL, -1, NULL, 0, 0);
        }

    ret = SVC2301TX_3_oracle(msg);
    if (ret < 0) {
        fail = 1;
    }

    ret = SVC2301TX_3_db2(msg);
    if (ret < 0) {
        fail = 1;
    }
}

```

```

        if ( fail ) {
            printf("SVC2301TX_3  FAIL\n");
            tpreturn(TPFAIL,-1,NULL, 0,0);
        }
        strcpy(rcvbuf, "Multiple RM Delete Success");
        tpreturn( TPSUCCESS, 0, rcvbuf, strlen(rcvbuf), 0 );
    }

SVC2301TX_4( TPSVCINFO *msg )
{
    int ret;
    int fail = 0;
    char *rcvbuf;

    rcvbuf=(char *)tpalloc("STRING", NULL, 4096);
    if(rcvbuf==NULL) {
        printf("tpalloc failed (rcvbuf) : %s\n", tpstrerror(tperrno));
        tpreturn(TPFAIL, -1, NULL, 0, 0);
    }

    ret = SVC2301TX_4_oracle(msg);
    if (ret < 0) {
        fail = 1;
    }

    ret = SVC2301TX_4_db2(msg);
    if (ret < 0) {
        fail = 1;
    }

    if ( fail ) {
        printf("SVC2301TX_4  FAIL\n");
        tpreturn(TPFAIL,-1,NULL, 0,0);
    }

    strcpy(rcvbuf, "Multiple RM Select Success");
    tpreturn( TPSUCCESS, 0, rcvbuf, strlen(rcvbuf), 0 );
}

```

Oracleの呼び出し関数

以下は、Oracleを呼び出す関数を使用したプログラムの例です。

```

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

EXEC SQL INCLUDE SQLCA.H;

EXEC SQL BEGIN DECLARE SECTION;
    int h_empno;
    char h_ename[MAXLEN];
    char h_job[MAXLEN];
    char h_date[MAXLEN];
    float h_sal;
    int h_count;
    char h_xid[100];
EXEC SQL END DECLARE SECTION;

SVC2301TX_1_oracle( TPSVCINFO *msg )
{
    str sndbuf;
    char *rcvbuf;
    char tmp[1024];
    TXINFO info;

    sndbuf = (str)msg->data;
    rcvbuf=(char *)tpalloc("STRING", NULL, 4096);
    if(rcvbuf==NULL) {
        printf("tpalloc failed (rcvbuf) : %s\n", tpstrerror(tperrno));
        tpretreturn(TPFAIL, -1, NULL, 0, 0);
    }

    h_empno = h_sal = 0;
    memset( h_ename, 0x00, sizeof( h_ename ) );
    memset( h_job, 0x00, sizeof( h_job ) );
    memset( h_date, 0x00, sizeof( h_date ) );
    memset( tmp , 0x00, sizeof( tmp ) );
    memset( h_xid , 0x00, sizeof( h_xid ) );

    h_empno = sndbuf->empno;
    h_sal = sndbuf->sal;
    strcpy( h_ename, sndbuf->ename );
    strcpy( h_job , sndbuf->job );
    strcpy( h_date , sndbuf->date );

    if (tx_info(&info)==1) {
        sprintf(h_xid, "%x %x %x %x - %x %x %x %x",
            (unsigned char)info.xid.data[0],

```

```

        (unsigned char)info.xid.data[1],
        (unsigned char)info.xid.data[2],
        (unsigned char)info.xid.data[3],
        (unsigned char)info.xid.data[4],
        (unsigned char)info.xid.data[5],
        (unsigned char)info.xid.data[6],
        (unsigned char)info.xid.data[7]);

    printf("xid=[%s] \n", h_xid);

} else {
    printf("Not in transaction \n");
    tpreturn(TPFAIL, -1, NULL, 0, 0);
}

EXEC SQL
INSERT INTO tmax( empno, ename, job, hiredate,sal, xid)
VALUES (:h_empno, :h_ename, :h_job, to_date(:h_date,'yymmdd'), :h_sal, :h_xid
);

if ( sqlca.sqlcode != SQLOK )
{
    sprintf(tmp, "[%s] ORACLE tmax Insert Fail", msg->name);
    strcpy(rcvbuf, tmp);
    printf("[%s] %d \n", rcvbuf, sqlca.sqlcode);
    tpreturn( TPFAIL, sqlca.sqlcode, rcvbuf, strlen(rcvbuf), 0 );
}

sprintf(tmp, "[%s] ORACLE tmax Insert Success", msg->name);
strcpy(rcvbuf, tmp);
return 0;
}

SVC2301TX_2_oracle( TPSVCINFO *msg )
{
    str  sndbuf;
    char *rcvbuf;
    char tmp[1024];

    sndbuf = (str)msg->data;
    rcvbuf=(char *)tpalloc("STRING", NULL, 4096);
    if(rcvbuf==NULL) {
        printf("tpalloc failed (rcvbuf) : %s\n", tpstrerror(tperrno));
        tpreturn(TPFAIL, -1, NULL, 0, 0);
    }
}

```

```

h_empno = h_sal = 0;

memset( h_ename, 0x00, sizeof( h_ename ) );
memset( h_job, 0x00, sizeof( h_job ) );
memset( h_date, 0x00, sizeof( h_date ) );
memset( tmp , 0x00, sizeof( tmp ) );

h_empno = sndbuf->empno;
h_sal   = sndbuf->sal;
strcpy( h_ename, sndbuf->ename );
strcpy( h_job , sndbuf->job );
strcpy( h_date , sndbuf->date );

EXEC SQL UPDATE tmax
      SET      empno = :h_empno+1,
              ename = :h_ename,
              job   = 'UPDATE'
      WHERE    empno = :h_empno;

if ( sqlca.sqlcode != SQLOK )
{
    sprintf(tmp, "[%s] ORACLE tmax Update Fail", msg->name);
    strcpy(rcvbuf, tmp);
    printf("[%s] %d \n", rcvbuf, sqlca.sqlcode);
    tpreturn( TPFAIL, sqlca.sqlcode, rcvbuf, strlen(rcvbuf), 0 );
}

sprintf(tmp, "[%s] ORACLE tmax Update Success", msg->name);
strcpy(rcvbuf, tmp);
return 0;
}

SVC2301TX_3_oracle( TPSVCINFO *msg )
{
    str  sndbuf;
    char *rcvbuf;
    char tmp[1024];

    sndbuf = (str)msg->data;
    rcvbuf=(char *)tpalloc("STRING", NULL, 4096);
    if(rcvbuf==NULL) {
        printf("tpalloc failed (rcvbuf) : %s\n", tpstrerror(tperrno));
        tpreturn(TPFAIL, -1, NULL, 0, 0);
    }

    h_empno = h_sal = 0;

```

```

        memset( h_ename, 0x00, sizeof( h_ename ) );
        memset( h_job, 0x00, sizeof( h_job ) );
        memset( h_date, 0x00, sizeof( h_date ) );
        memset( tmp , 0x00, sizeof( tmp ) );

h_empno = sndbuf->empno;
h_sal   = sndbuf->sal;
        strcpy( h_ename, sndbuf->ename );
        strcpy( h_job , sndbuf->job );
        strcpy( h_date , sndbuf->date );

EXEC SQL DELETE tmax
        WHERE empno = :h_empno;

if ( sqlca.sqlcode != SQLOK )
{
        sprintf(tmp, "[%s] ORACLE tmax Delete Fail", msg->name);
        strcpy(rcvbuf, tmp);
        printf("[%s] %d \n", rcvbuf, sqlca.sqlcode);
        tpreturn( TPFAIL, sqlca.sqlcode, rcvbuf, strlen(rcvbuf), 0 );
}

sprintf(tmp, "[%s] ORACLE tmax Delete Success", msg->name);
strcpy(rcvbuf, tmp);
return 0;
}

SVC2301TX_4_oracle( TPSVCINFO *msg )
{
        str  sndbuf;
        char *rcvbuf;
        char tmp[1024];

        sndbuf = (str)msg->data;
        rcvbuf=(char *)tpalloc("STRING", NULL, 4096);
        if(rcvbuf==NULL) {
                printf("tpalloc failed (rcvbuf) : %s\n", tpstrerror(tperrno));
                tpreturn(TPFAIL, -1, NULL, 0, 0);
        }

        h_empno = h_sal = h_count = 0;

        memset( h_ename, 0x00, sizeof( h_ename ) );
        memset( h_job, 0x00, sizeof( h_job ) );
        memset( h_date, 0x00, sizeof( h_date ) );

```

```

        memset( tmp , 0x00, sizeof( tmp ) );

h_empno = sndbuf->empno;
h_sal   = sndbuf->sal;
        strcpy( h_ename, sndbuf->ename );
        strcpy( h_job   , sndbuf->job   );
        strcpy( h_date  , sndbuf->date  );

EXEC SQL SELECT COUNT(*)
        INTO      :h_count
        FROM      tmax
        WHERE     empno = :h_empno;

if ( ( sqlca.sqlcode != SQLOK ) && ( sqlca.sqlcode != SQLNOTFOUND ) )
{
        sprintf(tmp, "[%s] ORACLE tmax Select Fail", msg->name);
        strcpy(rcvbuf, tmp);
        printf("[%s] %d \n", rcvbuf, sqlca.sqlcode);
        tpreturn( TPFAIL, sqlca.sqlcode, rcvbuf, strlen(rcvbuf), 0 );
}

sprintf(tmp, "[%s] ORACLE tmax Select Success [%d]", msg->name, h_count);
strcpy(rcvbuf, tmp);
return 0;
}

```

DB2の呼び出し関数

以下は、DB2を呼び出す関数を使用したプログラムの例です。

```

#include <stdio.h>
#include <ctype.h>
#include <usrinc/atmi.h>
#include <usrinc/tx.h>
#include "svr1.h"

EXEC SQL INCLUDE SQLCA;

EXEC SQL BEGIN DECLARE SECTION;
        sqlint32  h_empno;
        char h_ename[50];
        char h_job[50];
        char h_date[50];
        float h_sal;
        sqlint32  h_count;

```



```

    char h_xid[100];
EXEC SQL END DECLARE SECTION;

SVC2301TX_1_db2( TPSVCINFO *msg )
{
    str  sndbuf;
    char *rcvbuf;
    char tmp[1024];
    TXINFO info;

    sndbuf = (str)msg->data;
    rcvbuf=(char *)tpalloc("STRING", NULL, 4096);
    if(rcvbuf==NULL) {
        printf("tpalloc failed (rcvbuf) : %s\n", tpstrerror(tperrno));
        tpreturn(TPFAIL, -1, NULL, 0, 0);
    }

    h_empno = h_sal = 0;

    memset( h_ename, 0x00, sizeof( h_ename ) );
    memset( h_job, 0x00, sizeof( h_job ) );
    memset( h_date, 0x00, sizeof( h_date ) );
    memset( tmp , 0x00, sizeof( tmp ) );
    memset( h_xid , 0x00, sizeof( h_xid ) );

    h_empno = sndbuf->empno;
    h_sal   = sndbuf->sal;
    strcpy( h_ename, sndbuf->ename );
    strcpy( h_job , sndbuf->job );
    strcpy( h_date , sndbuf->date );

    if (tx_info(&info)==1) {
        sprintf(h_xid, "%x %x %x %x - %x %x %x %x",
            (unsigned char)info.xid.data[0],
            (unsigned char)info.xid.data[1],
            (unsigned char)info.xid.data[2],
            (unsigned char)info.xid.data[3],
            (unsigned char)info.xid.data[4],
            (unsigned char)info.xid.data[5],
            (unsigned char)info.xid.data[6],
            (unsigned char)info.xid.data[7]);

        printf("xid=[%s] \n", h_xid);
    } else {
        printf("Not in transaction \n");
        tpreturn(TPFAIL, -1, NULL, 0, 0);
    }
}

```

```

    }

EXEC SQL
INSERT INTO tmax( empno, ename, job, hiredate,sal, xid)
VALUES (:h_empno, :h_ename, :h_job, to_date(:h_date,'yymmdd'), :h_sal, :h_xid
);

if ( sqlca.sqlcode != SQLOK )
{
    sprintf(tmp, "[%s] DB2 tmax Insert Fail", msg->name);
    strcpy(rcvbuf, tmp);
    printf("[%s] %d \n", rcvbuf, sqlca.sqlcode);
    tpreturn( TPFAIL, sqlca.sqlcode, rcvbuf, strlen(rcvbuf), 0 );
}

sprintf(tmp, "[%s] DB2 tmax Insert Success", msg->name);
strcpy(rcvbuf, tmp);
return 0;
}

SVC2301TX_2_db2( TPSVCINFO *msg )
{
    str  sndbuf;
    char *rcvbuf;
    char tmp[1024];

    sndbuf = (str)msg->data;
    rcvbuf=(char *)tpalloc("STRING", NULL, 4096);
    if(rcvbuf==NULL) {
        printf("tpalloc failed (rcvbuf) : %s\n", tpstrerror(tperrno));
        tpreturn(TPFAIL, -1, NULL, 0, 0);
    }

    h_empno = h_sal = 0;

    memset( h_ename, 0x00, sizeof( h_ename ) );
    memset( h_job, 0x00, sizeof( h_job ) );
    memset( h_date, 0x00, sizeof( h_date ) );
    memset( tmp , 0x00, sizeof( tmp ) );

    h_empno = sndbuf->empno;
    h_sal   = sndbuf->sal;
    strcpy( h_ename, sndbuf->ename );
    strcpy( h_job , sndbuf->job );
    strcpy( h_date , sndbuf->date );

```

```

EXEC SQL UPDATE tmax
    SET      empno = :h_empno+1,
            ename = :h_ename,
            job   = 'UPDATE'
    WHERE    empno = :h_empno;

if ( sqlca.sqlcode != SQLOK )
{
    sprintf(tmp, "[%s] DB2 tmax Update Fail", msg->name);
    strcpy(rcvbuf, tmp);
    printf("[%s] %d \n", rcvbuf, sqlca.sqlcode);
    tpreturn( TPFAIL, sqlca.sqlcode, rcvbuf, strlen(rcvbuf), 0 );
}

sprintf(tmp, "[%s] DB2 tmax Update Success", msg->name);
strcpy(rcvbuf, tmp);
return 0;
}

SVC2301TX_3_db2( TPSVCINFO *msg )
{
    str  sndbuf;
    char *rcvbuf;
    char tmp[1024];

    sndbuf = (str)msg->data;
    rcvbuf=(char *)tpalloc("STRING", NULL, 4096);
    if(rcvbuf==NULL) {
        printf("tpalloc failed (rcvbuf) : %s\n", tpstrerror(tperrno));
        tpreturn(TPFAIL, -1, NULL, 0, 0);
    }

    h_empno = h_sal = 0;

    memset( h_ename, 0x00, sizeof( h_ename ) );
    memset( h_job, 0x00, sizeof( h_job ) );
    memset( h_date, 0x00, sizeof( h_date ) );
    memset( tmp , 0x00, sizeof( tmp ) );

    h_empno = sndbuf->empno;
    h_sal   = sndbuf->sal;
    strcpy( h_ename, sndbuf->ename );
    strcpy( h_job , sndbuf->job );
    strcpy( h_date , sndbuf->date );

```

```

EXEC SQL DELETE tmax
        WHERE      empno = :h_empno;

if ( sqlca.sqlcode != SQLOK )
{
    sprintf(tmp, "[%s] DB2 tmax Delete Fail", msg->name);
    strcpy(rcvbuf, tmp);
    printf("[%s] %d \n", rcvbuf, sqlca.sqlcode);
    tpreturn( TPFAIL, sqlca.sqlcode, rcvbuf, strlen(rcvbuf), 0 );
}

sprintf(tmp, "[%s] DB2 tmax Delete Success", msg->name);
strcpy(rcvbuf, tmp);
return 0;
}

SVC2301TX_4_db2( TPSVCINFO *msg )
{
    str  sndbuf;
    char *rcvbuf;
    char tmp[1024];

    sndbuf = (str)msg->data;
    rcvbuf=(char *)tpalloc("STRING", NULL, 4096);
    if(rcvbuf==NULL) {
        printf("tpalloc failed (rcvbuf) : %s\n", tpstrerror(tperrno));
        tpreturn(TPFAIL, -1, NULL, 0, 0);
    }

    h_empno = h_sal = h_count = 0;

    memset( h_ename, 0x00, sizeof( h_ename ) );
    memset( h_job, 0x00, sizeof( h_job ) );
    memset( h_date, 0x00, sizeof( h_date ) );
    memset( tmp , 0x00, sizeof( tmp ) );

    h_empno = sndbuf->empno;
    h_sal   = sndbuf->sal;
    strcpy( h_ename, sndbuf->ename );
    strcpy( h_job , sndbuf->job );
    strcpy( h_date , sndbuf->date );

    EXEC SQL SELECT COUNT(*)
        INTO      :h_count
        FROM      tmax
        WHERE      empno = :h_empno;

```

```

if ( ( sqlca.sqlcode != SQLOK ) && ( sqlca.sqlcode != SQLNOTFOUND ) )
{
    sprintf(tmp, "[%s] DB2 tmax Select Fail", msg->name);
    strcpy(rcvbuf, tmp);
    printf("[%s] %d \n", rcvbuf, sqlca.sqlcode);
    tpreturn( TPFAIL, sqlca.sqlcode, rcvbuf, strlen(rcvbuf), 0 );
}

sprintf(tmp, "[%s] DB2 tmax Select Success [%d]", msg->name, h_count);
strcpy(rcvbuf, tmp);
return 0;
}

```

注

MultipleRMではXA動的登録をサポートしないため、SELECT文を呼び出す際、常にtx_begin()を呼び出す必要があります。また、正常な動作のためには、挿入、更新、削除などを実行するサービスを持たない別のサーバーにSELECT文を処理するサービスを分離して処理する必要があります。

3.4.2. 環境設定とコンパイル

TiberoとOracleを使用するMultipleRMの環境設定とコンパイルの例について説明します。

環境設定

以下は、MultipleRMサーバーの環境設定ファイルの例です。

```

*DOMAIN

dom1          SHMKEY =@SHMEMKY@,      MINCLH=1, MAXCLH=1,
              TPORTNO=@TPORTNO@,      BLOCKTIME=60,
              RACPORT = @TRACPORT@,
              MAXTMS=50

*NODE
@HOSTNAME@     TMAXDIR="@TMAXDIR@",
              APPDIR="@TMAXDIR@/appbin",
              PATHDIR = "@TMAXDIR@/path",
              TLOGDIR = "@TMAXDIR@/log/tlog",
              ULOGDIR="@TMAXDIR@/log/ulog",
              SLOGDIR="@TMAXDIR@/log/slog",

*SVRGROUP
svg12301X      NODENAME = @HOSTNAME@,

```

```

        DBNAME = ORACLE,

OPENINFO="Oracle_Xa+Acc=P/scott/tiger+SesTm=60+DbgFl=7+LogDir=@TMAXDIR@/log/xalog",

        TMSNAME = tms_ora,
        SVGTYPE = STMAX

svg12302X    NODENAME = @HOSTNAME@,
            DBNAME = DB2_STATIC,
            OPENINFO = "db=tpctest,uid=tmaxha,pwd=ha0115",
            TMSNAME = tms_db2,
            SVGTYPE = STMAX

svgm1       NODENAME = @HOSTNAME@,
            SVGLIST = "svg12301X, svg12302X",
            SVGTYPE = MTMAX

*SERVER
# Common Insert/Update/Delete/Select
svr2301TX    SVGNAME = svgm1, MIN=5, MAX=5, MAXRSTART=-1
svr2301TX_UCS    SVGNAME = svgm1, MIN=5, MAX=5, MAXRSTART=-1, SVRTYPE=UCS

# Server Transaction
svr2311TX    SVGNAME = svg12301X

# tpforward
svr2321TX    SVGNAME = svg12301X
svr2322TX    SVGNAME = svg12301X

# AUTOTRAN
svr2331TX    SVGNAME = svg12301X

*SERVICE
# Common Insert/Update/Delete/Select
SVC2301TX_1    SVRNAME = svr2301TX, SVCTIME=5
SVC2301TX_2    SVRNAME = svr2301TX, SVCTIME=5
SVC2301TX_3    SVRNAME = svr2301TX, SVCTIME=5
SVC2301TX_4    SVRNAME = svr2301TX, SVCTIME=5
SVC2301TX_1_UCS    SVRNAME = svr2301TX_UCS, SVCTIME=5
SVC2301TX_2_UCS    SVRNAME = svr2301TX_UCS, SVCTIME=5
SVC2301TX_3_UCS    SVRNAME = svr2301TX_UCS, SVCTIME=5
SVC2301TX_4_UCS    SVRNAME = svr2301TX_UCS, SVCTIME=5

```

Makefile

Makefile内のLIBS(TMAXライブラリー)には、**-lnodb**が含まれている必要があります。

以下は、64ビットLinuxでMultipleRMサーバー・プログラムをコンパイルするためのMakefileの例です。

```
# Oracle Env
include $(ORACLE_HOME)/precomp/lib/env_precomp.mk
ORALIBDIR = $(LIBHOME)
ORALIB = $(PROLDLIBS)

# DB2 Env
DB2LIBDIR = $(DB2_HOME)/lib
DB2LIBS    = -ldb2

DB          = TPTEST
DB2USER     = tmaxha
DB2PASS     = ha0115

# Server makefile

TARGET      = $(COMP_TARGET)
APOBJS      = $(TARGET).o
NSDLOBJ     = $(TMAXDIR)/lib64/sdl.o
AP_ORA_OBJS = $(TARGET)_ora.o
AP_DB2_OBJS = $(TARGET)_db2.o

LIBS        = -lsvr -lnsl -lnodb
OBJS        = $(AP_ORA_OBJS) $(AP_DB2_OBJS) $(APOBJS) $(SVCTOBJ)
SVCTOBJ     = $(TARGET)_svctab.o

CFLAGS      = -O -I$(TMAXDIR)

APPDIR      = $(TMAXDIR)/appbin
SVCTDIR     = $(TMAXDIR)/svct
TMAXLIBDIR  = $(TMAXDIR)/lib64

#
.SUFFIXES : .c

.c.o:
    $(CC) $(CFLAGS) -c $<

#
# server compile
```

```

#
all: $(TARGET)

$(TARGET): $(OBJS)
    echo $(ORALIBDIR)
    $(CC) $(CFLAGS) -L$(TMAXLIBDIR) -o $(TARGET) -L$(DB2LIBDIR) -L$(ORALIBDIR)
$(ORALIB) $(DB2LIBS) $(OBJS) $(LIBS) $(NSDLOBJ)
    mv $(TARGET) $(APPDIR)/.
    rm -f $(OBJS)

$(AP_ORA_OBJS):
    echo $(OBJS)
    proc iname=$(TARGET)_ora.pc include=$(TMAXDIR)
    $(CC) $(CFLAGS) -c $(TARGET)_ora.c

$(AP_DB2_OBJS):
    echo $(OBJS)
    db2 connect to $(DB) user $(DB2USER) using $(DB2PASS)
    db2 prep $(TARGET)_db2.sqc bindfile
    db2 bind $(TARGET)_db2.bnd
    db2 connect reset
    db2 terminate
    $(CC) $(CFLAGS) $(LDFLAGS) -I$(DB2_HOME)/include -c $(TARGET)_db2.c

$(APOBJS): $(TARGET).c
    $(CC) $(CFLAGS) -c $(TARGET).c

$(SVCTOBJ):
    cp -f $(SVCTDIR)/$(TARGET)_svctab.c .
    touch ./$(TARGET)_svctab.c
    $(CC) $(CFLAGS) -c ./$(TARGET)_svctab.c

#
clean:
    -rm -f *.o core $(APPDIR)/$(TARGET)

```

参考

OSに応じて、Makefileの内容が異なる場合があります。

索引

M

MTMAX関連の設定

NODENAME, 4

SVGLIST, 4

SVGTYPE, 4

MultipleRM, 1

MultipleRMの制約, 2

S

STMAX関連の設定

DBNAME, 4

NODENAME, 4

OPENINFO, 4

RMID, 4

SVGTYPE, 4

TMSNAME, 4

