

Tmax WebTユーザガイド

WebT v7.0



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 (WebT) 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(WebT)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®, WebT 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®, WebTおよび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, netshmp, 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 WebTユーザガイド

発行日: 2016年8月5日

ソフトウェアバージョン: WebT v7.0

ガイドバージョン: v2.1.7

目次

このガイドについて	ix
第1章 WebTの紹介	1
1.1. 概要	1
1.2. WebTConnectionPool	1
1.3. WebT-Serverシステム	2
第2章 環境設定	5
2.1. WebTの環境設定	5
2.1.1. JEUSMain.xmlの設定	5
2.1.2. webt.propertiesの設定	14
2.1.3. それ以外の環境設定	27
2.2. JTmaxの環境設定	28
2.2.1. JEUSMain.xmlの設定	28
2.2.2. EJBの設定	32
第3章 WebT API	35
3.1. 概要	35
3.2. 接続関連クラス	37
3.2.1. WebtConnection	37
3.2.2. WebtConnectionPool	42
3.2.3. WebtConnectionGroup	47
3.3. サービス呼び出し関連クラス	49
3.3.1. WebtAttribute	49
3.3.2. WebtBuffer	55
3.3.3. WebtCarrayBuffer/WebtStringBuffer	60
3.4. サービス呼び出し関連クラス	65
3.4.1. WebtRemoteService	65
3.4.2. WebtRQService	71
3.4.3. WebtDialogueService	78
3.5. フィールド・データ関連クラス	81
3.5.1. WebtFieldElement	81
3.5.2. WebtFDLKeyTable	85
3.5.3. WebtFieldSet	88
3.5.4. WebtField	95
3.5.5. WebtFieldBuffer	100
3.6. イベント・メッセージ処理関連クラス	102
3.6.1. WebtEventConnecion	102
3.6.2. WebtEventHandler	109
3.7. その他のクラス	111
3.7.1. WebtTransaction	111
3.7.2. WebtSystem	113

第4章 WebT HMS	117
4.1. 概要	117
4.2. JMS API	117
4.3. 環境設定	120
4.3.1. Tmaxの設定	120
4.3.2. JEUSMain.xmlの設定	122
4.3.3. webt.propertiesの設定	123
4.4. プログラムの例	126
第5章 モニタリング	129
5.1. 概要	129
5.2. 環境設定	129
5.3. WebTのモニタリング	133
5.3.1. JEUSコンソール・ツールの使用	133
5.3.2. JEUS WebAdminの使用	139
5.4. JTmaxのモニタリング	141
5.4.1. JEUSコンソール・ツールの使用	141
5.4.2. JEUS WebAdminの使用	143
5.5. JEUS WebtAdminツール	144
5.5.1. WebTのモニタリング	145
5.5.2. JTmaxのモニタリング	147
付録 A. webterrorcodeメッセージ	149
A.1. webterrorcodeメッセージ(1000~)	149
A.2. webterrorcodeメッセージ(2000~)	166
A.3. webterrorcodeメッセージ(3000~)	172
A.4. webterrorcodeメッセージ(4000~)	173
A.5. webterrorcodeメッセージ(5000~)	184
A.6. webterrorcodeメッセージ(6000~)	188
A.7. webterrorcodeメッセージ(7000~)	190
A.8. webterrorcodeメッセージ(8000~)	193
A.9. webterrorcodeメッセージ(9000~)	201
索引	205

図目次

[図 1.1]	WebTとTmax間のサービス・フロー	1
[図 1.2]	WebT-Serverシステム	2
[図 5.1]	JEUS WebAdmin画面	139
[図 5.2]	Webtモニタリングのメイン画面	140
[図 5.3]	Webtモニタリングのコネクション・グループ情報の画面	141
[図 5.4]	JTmax Monitoringのメイン画面	144
[図 5.5]	WebAdmin画面	145
[図 5.6]	Webt Monitoringのメイン画面	146
[図 5.7]	Webt Monitoringのコネクション・グループ情報画面	146
[図 5.8]	JTmax Monitoringのメイン画面	147

このガイドについて

対象読者

本書は、Webを通じてTmax[®](以下、Tmax)のサービスの提供を受けるため、JEUS[®](以下、JEUS)のWebT APIを使用してクライアント・プログラムを開発するプログラム開発者を対象にしています。

WebTはWeb Transactionの略で、クライアント/サーバー(C/S)環境のミドルウェア製品であるTmaxとJavaアプリケーション・プログラム間のトランザクション・サービスをサポートします。Javaアプリケーション・プログラムがTmaxサーバーに接続できるようにするAPIで、ライブラリー形式でデプロイされます。

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取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
- JEUSについて

制限事項

本書では、WebTのAPIを活用してTmaxクライアント・プログラムを開発するために必要な内容をすべて取り扱っていません。本書で扱っているAPI、バッファー、トランザクション、RQなどの詳細については『Tmaxアプリケーション開発ガイド』を参照してください。

本書の構成

本書は5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: WebTの紹介

WebTの基本概念と、WebTConnectionPool、WebT-Server Systemなどの主要構成要素について説明します。

- 第2章 環境設定

WebTを起動する前に必要な環境設定方法について説明します。

- 第3章: WebT API

WebTで提供するAPIをクラス別に説明します。

- 第4章: WebT HMS

WebTを使用してTmaxに提供するHMS(Hybrid Messaging System)に接続してメッセージング・サービスを使用する方法について説明します。

- 第5章: モニタリング

JEUS管理ツールを利用してWebTとJTmaxをモニタリングするための設定および使用方法について説明します。

- 付録A: webterrorcodeメッセージ

webterrorcodeで発生するすべてのメッセージ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
参考	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

関連文書

ガイド	説明
Tmax メッセージリファレンスガイド	Tmax製品の発生時に発生する可能性のあるメッセージ(エラー・メッセージを含む)と、その対応方法について説明しています
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 HMSユーザガイド	HMS(Hybrid Messaging System)の基本概念と環境設定方法、APIの使用方法および例について説明しています
Tmax JTCユーザガイド	JTCの基本概念と環境設定方法、APIの使用方法および例について説明しています

参考資料

- JEUSガイド
JEUS WebAdminガイド
- <http://java.sun.com/j2ee/1.4/docs/api/javax/jms/package-tree.html>

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 WebTの紹介

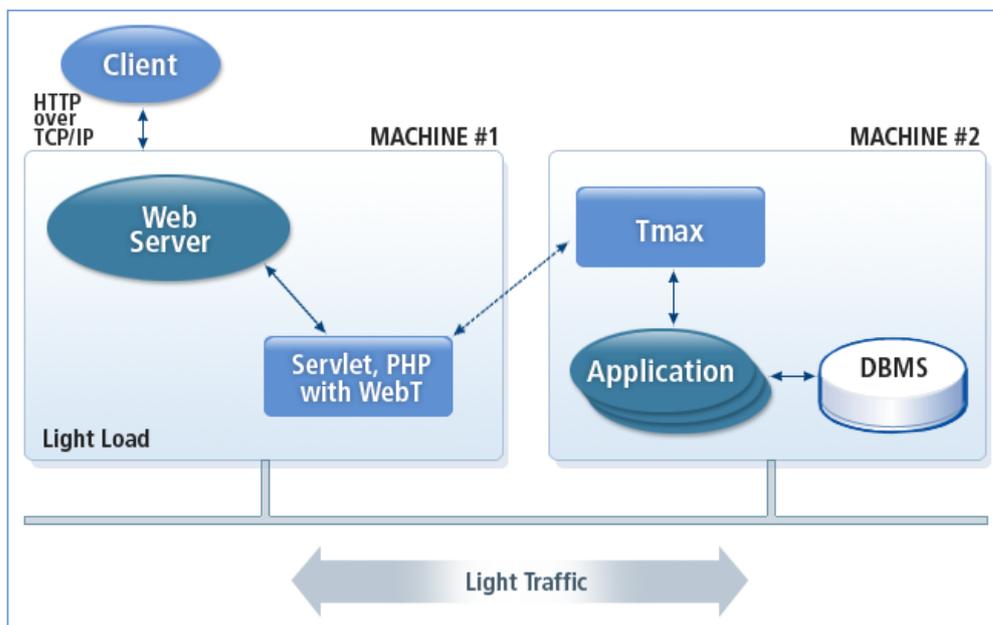
本章は、TmaxとJavaアプリケーション・プログラムとのトランザクション・サービスをサポートするために提供されるWebTについての基本機能と役割について説明します。

1.1. 概要

WebT(Web Transaction)は、クライアント/サーバー環境のミドルウェア製品であるTmaxサーバーとJavaアプリケーション・プログラム間のトランザクション・サービスをサポートするプログラムで、APIライブラリー形式でデPLOYされます。WebTは、JEUSを始めとするWeb基盤環境のWAS(Web Application Server)製品で活用されます。Tmaxのトランザクション処理と負荷調節機能を利用し、Web環境で動的データサービスを提供できるように設計されています。

以下の図は、WebTとTmaxサーバー間におけるサービスの手順を示しています。クライアントから要求を受けると、WebTモジュールを通じてTmaxのサーバー・プログラムが実行され、サービスが行われます。

【図 1.1】 WebTとTmax間のサービス・フロー



1.2. WebTConnectionPool

WebTは、Tmaxの接続を効率的に管理するため、WebTConnectionPoolクラスを提供します。WebTConnectionPoolは、Tmaxサービスを要求する度に接続オブジェクトを新しく作成するのではなく、以

前に使用した接続オブジェクトを再使用します。そのため、Tmaxサーバーにネットワーク接続を設定および終了するのに消費されるリソースおよび時間を節約できます。

WebTConnectionPoolは1個以上のWebTConnectionGroupで構成され、各WebTConnectionGroupは1個のTmaxサーバーと接続されます。クライアント・プログラムでは、WebTConnectionGroupのグループ名を使用してTmaxサーバーに接続できます。

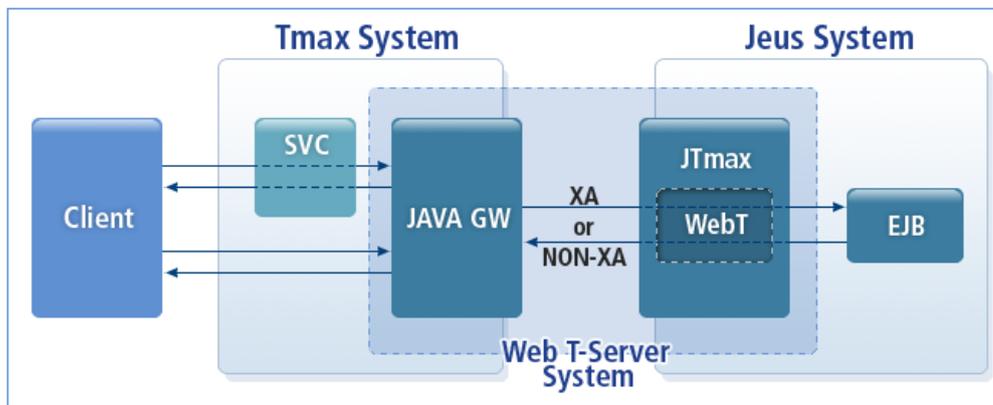
WebTモジュールをJEUSと連動させた場合、JEUSコンテナでWebTConnectionPoolを管理します。JEUSコンテナは、WebT.propertiesの環境設定ファイルまたはJEUSの環境設定ファイルを利用してコネクション・プールを作成します。また、ユーザーがgetConnectionを実行後に返していないコネクションを自動でコネクション・プールに戻します。

1.3. WebT-Serverシステム

WebT-ServerシステムはTmaxシステムとJEUSの間に存在し、TmaxクライアントがJEUSのEJBサービス呼び出せるようにします。

以下の図は、WebT-Serverシステムを通じてEJBサービスが呼び出される手順です。

[図 1.2] WebT-Serverシステム



WebT-Serverシステムは、以下のモジュールで構成されています。

- JAVA GW
TmaxがJEUSに送信するサービス要求を処理します。
- JTmax
JEUSでTmaxのサービス要求を受信するデーモンです。
- WebT Library
TmaxとJEUSが送受信するデータを処理します。
- その他のユーティリティー

webtutil.jarは、jeus.jarとjeusutil.jarでWebTを使用するためのクラスを別途パッケージングしたファイルで、JEUSがインストールされていないマシンでWebTを起動するためのライブラリーです。

第2章 環境設定

本章では、WebTとJTmaxを起動する前に必要な環境設定方法について説明します。

2.1. WebTの環境設定

JEUSでWebTを起動するには、JEUSMain.xmlあるいはwebt.propertiesファイルのうち1つを設定します。

2.1.1. JEUSMain.xmlの設定

JEUSでWebTを起動するため、JEUSMain.xmlにexternal-sourceでWebTConnectionPoolの環境情報を登録します。WebTConnectionPoolを設定するには、JEUSMain.xmlファイルの<external-resource>タグ内に以下のように定義します。

```
<resource>
  ....
  <external-resource>
    <property>
      <key>header-type</key>
      <value>extendedV4</value>
    </property>
    <property>
      <key>log-file-name</key>
      <value>d:\\webt.log</value>
    </property>
    <property>
      <key>log-level</key>
      <value>debug</value>
    </property>
    <property>
      <key>log-valid-day</key>
      <value>1</value>
    </property>
    <property>
      <key>monitor-interval</key>
      <value>30000</value>
    </property>
    <property>
      <key>check-alive</key>
```

```
        <value>true</value>
    </property>
    <property>
        <key>enable-failback</key>
        <value>>false</value>
    </property>
    <property>
        <key>enable-autoclose</key>
        <value>true</value>
    </property>
    <property-group>
        <name-prefix>tmax2</name-prefix>
        <property>
            <key>type</key>
            <value>shared</value>
        </property>
        <property>
            <key>host-name</key>
            <value>192.168.1.123</value>
        </property>
        <property>
            <key>host-port</key>
            <value>8888</value>
        </property>
        <property>
            <key>min</key>
            <value>10</value>
        </property>
        <property>
            <key>max</key>
            <value>20</value>
        </property>
        <property>
            <key>step</key>
            <value>2</value>
        </property>
        <property>
            <key>check-acquired</key>
            <value>true</value>
        </property>
        <property>
            <key>support-xa</key>
            <value>true</value>
        </property>
        <property>
            <key>enable-autoclose</key>
            <value>true</value>
        </property>
    </property-group>

```

```

        </property>
        <property>
            <key>mbean</key>
            <value>>true</value>
        </property>
    </property-group>
    <property-group>
        <name-prefix>tmax</name-prefix>
        <property>
            <key>type</key>
            <value>clustered</value>
        </property>
        <property>
            <key>connect-timeout</key>
            <value>60</value>
        </property>
    </property-group>
    <name>webt</name>
    <class-name>tmax.webt.external.WebtResource</class-name>
</external-resource>
...
</resource>

```

注

JEUSMain.xmlの<resource>タグ内に定義した場合は、JEUSマネージャーが起動した際にこのコネクション・グループが登録されます。特定コネクションを設定した<engine-container>タグの下に定義した場合は、該当コネクションが起動した際に登録されます。

複数の場所に同じ名前を登録した場合は、後に登録されたコネクション・グループの設定が以前に登録されたコネクション・グループの設定を上書きするため、同じ名前を重複して登録しないように注意します。

以下は、JEUSMain.xmlを設定する際に使用する<external-resource>のサブタグについての説明です。

- <property>

すべてのWebTConnectionPoolに適用する属性を定義します。Subタグとして<key>、<value>があり、それについての詳細な説明は[<property-group>/<property>属性](#)を参照してください。

タグ	説明
<key>	WebTConnectionPoolに適用する属性の名前を定義します
<value>	定義した属性に適用可能な値を設定します

- <property-group>

1つのWebTConnectionPoolに適用する属性を定義します。<property-group>タグを1回定義する度にWebTConnectionPoolが1つ作成されます。<property-group>タグを数回定義し、多数のWebTConnectionPoolを作成できます。

タグ	説明
<name-prefix>	アプリケーション・プログラムから接続するコンテキストの名前を定義します

- <name>

WebTConnectionPoolの名前を任意で設定します。ただし、<resource>タグ内部では一意の名前を定義します。

- <class-name>

WebTConnectionPoolを利用するには、WebTConnectionPoolで使用するクラス名の定義が必要です。

<property-group>/<property>属性

<property>と<property-group>内部に<key>、<value>タグを使用してWebTConnectionPoolに適用する属性と適用可能な値を設定します。<key>タグには属性名を、<value>タグには適用可能な値を設定します。

以下は、<key>タグに設定可能な属性名と<value>タグに設定可能な値についての説明です。

<key>の属性	区分	<value>の説明
log-file-name	設定値	String
	デフォルト値	NULL
	説明	ログファイルの相対パスまたは絶対パスを指定します。相対パスを指定した場合は\$JEUS_HOME/logsディレクトリーの下にログファイルが作成され、この値を指定しなかった場合はSTDOUTのログが出力されます。OSがWindowsの場合、パスの区切り子をスラッシュ(\)ではなくダブルスラッシュ(\\)にします
log-level	設定値	none debug info
	デフォルト値	info
	説明	ログレベルを指定します。 <ul style="list-style-type: none"> – none : 値を出力しません。 – debug : request msg、reply msg、xa msgなど、デバッグ関連の内容を出力します – info : request msg、reply msgを出力します
log-file-format	設定値	String

<key>の属性	区分	<value>の説明
	デフォルト値	MMddyyyy
	説明	log-valid-dayが1以上の場合に作成されるログファイル名の形式を指定します
log-buffer-size	設定値	int
	デフォルト値	512
	説明	ログを残す際に使用されるバッファサイズを指定します
log-valid-day	設定値	int
	デフォルト値	-1
	説明	ログファイルを使用する場合、何日周期でログファイル名を更新するかを指定します。0や負数を指定した場合、1つのファイルを使用します
type	設定値	shared clustered
	デフォルト値	shared
	説明	WebTConnectionPoolのタイプを指定します。 <ul style="list-style-type: none"> – shared: 一般的な場合に指定します – clustered: WebTDataSourceのクラスタリングを使用する場合に指定します。tmax-delegation-sourceとconnection-timeoutを<property>として持ちます
tmax-delegation-source	設定値	String
	デフォルト値	NULL
	説明	typeをclusteredと設定した場合、クラスタリングに関わるWebTDataSourceのJNDI名のリストをコンマ(,)で区分して指定します
host-name	設定値	String
	デフォルト値	NULL
	説明	typeをsharedと設定した場合、TmaxサーバーのIPアドレスを指定します
host-port	設定値	int
	デフォルト値	-1
	説明	typeをsharedと設定した場合、TmaxサーバーのTCPポートを指定します。support-xaがTRUEの場合、JAVAGWのポートを必ず指定します
backup-host-name	設定値	String
	デフォルト値	NULL

<key>の属性	区分	<value>の説明
	説明	typeをsharedと設定し、バックアップ・サーバーを使用する場合、バックアップTmaxサーバーのIPアドレスを指定します
backup-port	設定値	int
	デフォルト値	-1
	説明	typeをsharedと設定し、バックアップ・サーバーを使用する場合、バックアップTmaxサーバーのTCPポートを指定します
default-charset	設定値	String
	デフォルト値	使用中のJDKのデフォルト値
	説明	Tmaxバッファの内容に適用される文字セットを指定します
fdl-file	設定値	String
	デフォルト値	NULL
	説明	TmaxFDLバッファを使用するためのフィールド表のパスを指定します。fdlc 6.0でビルドしたバイナリは使用できません。fdlc 6.0以下のバージョンでビルドしたファイルを使用してください
encryption	設定値	Boolean
	デフォルト値	FALSE
	説明	Tmaxとの通信で暗号化機能を使用するかどうかを設定します
user-name	設定値	String
	デフォルト値	NULL
	説明	ユーザー・セキュリティ認証のためのユーザーの名前を設定します
user-password	設定値	String
	デフォルト値	NULL
	説明	ユーザー・セキュリティ認証のためのユーザーの暗号を設定します
domain-name	設定値	String
	デフォルト値	NULL
	説明	ドメイン・セキュリティ認証のためのドメインの名前を設定します
domain-password	設定値	String
	デフォルト値	NULL
	説明	ドメイン・セキュリティ認証のためのドメインの暗号を設定します
min	設定値	int
	デフォルト値	2
	説明	WebTConnectionPoolのデフォルトのコネクション数を設定します
max	設定値	int

<key>の属性	区分	<value>の説明
	デフォルト値	30
	説明	WebTConnectionPoolの最大コネクション数を設定します
step	設定値	int
	デフォルト値	4
	説明	WebTConnectionPoolが不足して新しくコネクションを作成する場合、新規作成するコネクションの数を設定します
period	設定値	int
	デフォルト値	3600000
	説明	コネクションを削除する基準となる時間を設定します。WebTConnectionPoolのコネクション数がminに設定された数より大きい場合、periodに設定された時間内に使用されなかったコネクションをWebTConnectionPoolから削除します
fair-provision	設定値	Boolean
	デフォルト値	TRUE
	説明	WebTConnectionPoolでコネクションを取得するオブジェクトを決定します。 <ul style="list-style-type: none"> – TRUE: 使用していないコネクションを取得します – FALSE: 最近使用したコネクションを取得します
check-acquired	設定値	Boolean
	デフォルト値	FALSE
	説明	WebTConnectionPoolで取得したコネクションのlivenessチェックを行うかどうかを設定します。WebTDataSourceでgetConnectionを実行した際、実際にサーバーへアプリケーションのpingチェックを行うかどうかを設定します。 <ul style="list-style-type: none"> – TRUE: livenessチェックをします – FALSE: livenessチェックをしません
enable-autoclose	設定値	Boolean
	デフォルト値	FALSE
	説明	使用後に返されなかったコネクションを自動でWebTConnectionPoolに戻します。Java ServletとJSPでのみ有効です
connect-timeout	設定値	int
	デフォルト値	20(単位: 秒)

<key>の属性	区分	<value>の説明
	説明	WebTDataSourceからコネクションを取得する際に使用されるタイムアウト時間です。設定された時間内にコネクションを取得した場合、あるいは新規作成できなかった場合は例外を返します
support-xa	設定値	Boolean
	デフォルト値	TRUE
	説明	XAトランザクション・サービスを使用するか否かを決定します。XAトランザクションを使用する場合、RQや対話型サービスの使用は制限されます。コネクションのメイン・アドレスおよびバックアップ・アドレスはJAVAGWのアドレスを必ず使用します
service-timeout	設定値	int
	デフォルト値	-1
	説明	サービスを要求する際、service-timeoutに指定された時間内に応答がない場合は例外を返します
transaction-timeout	設定値	int
	デフォルト値	-1
	説明	トランザクションを使用する場合、transaction-timeoutに指定された時間内に処理されなかったトランザクションをロールバック処理します
transaction-block-timeout	設定値	int
	デフォルト値	60
	説明	トランザクションのコミットやロールバックを要求する際、transaction-block-timeoutに指定された時間内に応答がない場合は例外を返します
event-event	設定値	Boolean
	デフォルト値	FALSE
	説明	イベントを使用するかどうかを決定します
enable-flag	設定値	all broadcast sendtocli notify acall none
	デフォルト値	none
	説明	使用するイベントを設定します。 <ul style="list-style-type: none"> – all: すべてのイベントを使用します – broadcast: broadcastタイプのイベントを使用します – sendtocli: sendtocliタイプのイベントを使用します – notify: notifyタイプのイベントを使用します – acall: acallタイプのイベントを使用します

<key>の属性	区分	<value>の説明
		<ul style="list-style-type: none"> – none: イベントを使用しません <p>2種類以上のvalueを同時に設定する際はコンマ(,)で区分します (例: broadcast,notify)</p>
event-handler	設定値	String(クラス名)
	デフォルト値	NULL
	説明	イベントを受信する際に呼び出されるインターフェースを実装したオブジェクトの名前を設定します
header-type	設定値	default extendedV4
	デフォルト値	default
	説明	<p>Tmaxと通信する際のプロトコルタイプを決定します。</p> <ul style="list-style-type: none"> – default: Tmax 3.xで使用するプロトコルです – extendedV4: Tmax 4.xで使用するプロトコルです。Tmax 4.xは下位プロトコルをサポートするため、defaultと設定しても互換されます
monitor-interval	設定値	int
	デフォルト値	60000(単位:ミリ秒)
	説明	<p>WebtDatasourceの状態をチェックする周期を設定します。最小周期はバージョンによって以下のように異なります。</p> <ul style="list-style-type: none"> – 7.0.0.b8以前のバージョン: 最小周期は20秒です – 7.0.0.b8以降のバージョン: 最小周期は5秒です
check-alive	設定値	Boolean
	デフォルト値	FALSE
	説明	<p>livenessチェックを行うかどうかを設定します。</p> <ul style="list-style-type: none"> – TRUE: livenessチェックをします – FALSE: livenessチェックをしません
enable-failback	設定値	Boolean
	デフォルト値	FALSE
	説明	メインサーバーの問題が解決された場合、バックアップ・アドレスを指定したWebTDDataSourceが再びメインサーバーのサービスを使用するかどうかを指定します

<key>の属性	区分	<value>の説明
mbean	設定値	Boolean
	デフォルト値	FALSE
	説明	WebT AdminのためのMBeanを登録するかどうかを指定します。WebT Admin機能はJEUS 6.0 fix#7以上でサポートします

2.1.2. webt.propertiesの設定

JEUSでWebTを起動するために必要な環境設定は、WebTの環境設定ファイルであるwebt.propertiesを設定することによっても可能です。webt.propertiesファイルを使用してWebTが提供するライブラリーに実装されたクライアント・プログラムを起動する際、環境設定を初期化します。また、アプリケーション・プログラムを初期化する際、WebTSystem APIを使用する代わりにwebt.propertiesファイルで設定できます。

JEUSMain.xmlに以下のように設定します。

```
<engine-container>
  <name>container1</name>
  <id>60</id>
  <base-port>11351</base-port>
  <command-option>
    -Xms256m -Xmx512m -Dwebt.properties=c:\\tmaxsoft\\webt\\config\\webt.properties
  </command-option>
  .....
</engine-container>
```

注

JEUS_HOME/lib/application内にwebt.propertiesを置くとJEUSを起動する際に自動的に環境を読み取りますが、JEUS_HOME/lib/application/webt.propertiesとJEUSMain.xmlで同時にwebt.propertiesを設定すると問題が発生することがあるため注意が必要です。

JEUSMain.xmlに[-D]オプションでJEUSを起動する際、webt.propertiesの環境を設定することができ、JEUS_HOME/lib/application以外のフォルダに設定して使用することを推奨します。この場合、webt.propertiesに指定されたコネクション・グループの設定は[-D]オプションで定義された該当コネクションでのみ使用できます。

WebTの環境設定のうち、モニタリング関連の設定(chickAlive、interval、failback)の場合、webt.propertiesとexternal-resource(webt) ConnectionPoolを同時に使用したら、最初に起動されたコネクション・プールのモニタリング関連の設定値が維持されます。なお、その値が更新されないため、注意が必要です。(一度起動されると更新されません)

実際の値が書かれているラインの途中に「#」を入力することができません。入力しても、その設定を認識することができません。

webt.propertiesファイルには以下の値を設定できます。時間値は秒単位で設定されます。

```
#####
# logging related parameters.
#####
log.level=none | debug | info

# printed to standard out
log.dir=log.dir

# set the name of the log file. default is webt.log
log.file=webt.log

# set log buffering size. default is 0
log.bufsize=numer
log.file.date.format=MMddyyyy
log.valid.days=numer

#####
# FDL related parameters
#####
fdl.file=fdl.file

# Tmax 3.11.x extended Header size
headerType=default|extendedV4

# monioring related parameters
monitoring.pool.failback=true|false
monitoring.pool.checkAlive=true|false
monitoring.pool.interval=number

enableFastBackup=true|false
enableJavaSocketKeepAlive=true|false
useRollingDown=true|false

jeus.servlet.webt.autoClose.enable=true|false

#####
# connection pool realted paramters
#####
enableConnectionPool=true|false

connectionPool.groups=connectionPool name
connectionPool.<groupName>.connectTimeou=number
```

```

connectionPool.<groupName>.cluster=group name

connectionPool.<groupName>.hostAddr=hostAddr
connectionPool.<groupName>.hostPort=hostPort
connectionPool.<groupName>.hostBackupAddr=hostBackupAddr
connectionPool.<groupName>.hostBackupPort=hostBackupPort

connectionPool.<groupName>.encryption=true|false
connectionPool.<groupName>.check=true|false
connectionPool.<groupName>.userName=userName
connectionPool.<groupName>.userPasswd=userPasswd
connectionPool.<groupName>.domainName=domainName
connectionPool.<groupName>.domainPasswd=domainPasswd

connectionPool.<groupName>.fair.provision=true|false
connectionPool.<groupName>.initCapacity=number
connectionPool.<groupName>.maxCapacity=number
connectionPool.<groupName>.maxIdleTime=number

connectionPool.<groupName>.tpTimeout=number
connectionPool.<groupName>.txTimeout=number
connectionPool.<groupName>.txBlockTimeout=number
connectionPool.<groupName>.header.type=default | extendedV4
connectionPool.<groupName>.eventSvcType=all | broadcast | sendtocli | notify |
acall
connectionPool.<groupName>.eventHandler=shared | clustered

tmax.webt.xid.log=true | false
webt.mbean=true | false

```

以下は、各設定項目についての設定です。

- log.level = none | debug | info
 - デフォルト値 : info
 - ログレベルを指定します。

設定値	説明
none	値を出力しません。
debug	request msg、reply msg、xa msgなど、デバッグ内容を出力します
info	request msg、reply msgを出力します

- log.dir = String
 - デフォルト値 : NULL

- WebTのログ・ファイルが格納されているディレクトリー・パスを設定します。OSがWindowsの場合、パスの区切り子をスラッシュ(/)ではなくダブルスラッシュ(\\)にします。
- log.file = String
 - デフォルト値 : webt.log
 - WebTのログファイル名を指定します。
- log.bufsize = int
 - デフォルト値 : 512
 - ログを残す際に使用されるバッファのサイズを指定します。
- log.file.date.format = String
 - デフォルト値 : MMddyyyy
 - log.valid.daysが1以上の場合に作成されるログ・ファイル名の形式を指定します
- log.valid.days = int
 - デフォルト値 : -1
 - ログファイルを使用する場合、何日周期でログファイル名を更新するのかを指定します。0や負数を指定した場合、1つのファイルを使用します
- fdl.file = String
 - デフォルト値 : NULL
 - Tmax FDLバッファを使用するためのフィールド表のパスを指定します。fdlc 6.0でビルドしたバイナリは使用できません。fdlc 6.0以下のバージョンでビルドしたファイルを使用してください。
- headerType = default | extendedV4
 - デフォルト値 : default
 - Tmaxと通信する際のプロトコルタイプを決定します。ユーザーが直接作成した接続および接続・グループに適用されるプロトコル・タイプです。

設定値	説明
default	Tmax 3.xで使用するプロトコルです
extendedV4	Tmax 4.xで使用するプロトコルです。Tmax 4.xは下位プロトコルをサポートするため、defaultと設定しても互換されます

- monitoring.pool.failback = Boolean

– デフォルト値 : FALSE

– バックアップ・アドレスを指定し、メインサーバーの問題が解決された場合、再びメインサーバーのサービスを使用するかどうかを指定します。

設定値	説明
TRUE	メインサーバーのサービスを使用します
FALSE	メインサーバーのサービスを使用しません

- monitoring.pool.checkAlive = Boolean

– デフォルト値 : FALSE

– livenessチェックによる接続の状態チェックを行うかどうかを設定します。

設定値	説明
TRUE	livenessチェックをします
FALSE	livenessチェックをしません

- monitoring.pool.interval = int

– デフォルト値 : 60 (単位: 秒)

– 接続の状態チェック周期を設定します。最小周期はバージョンによって以下のように異なります。

設定値	説明
7.0.0.b8以前のバージョン	最小周期は20秒です
7.0.0.b8以降のバージョン	最小周期は5秒です

- monitoring.pool.disableCheckAliveLog = Boolean

– デフォルト値 : FALSE

– checkAliveを設定する際にコネクション点検時に発生するログを残すかどうかを設定します。

設定値	説明
TRUE	コネクション点検ログを残しません
FALSE	コネクション点検ログを残します

- enableFastBackup = Boolean

– デフォルト値 : FALSE

– タイムアウトが発生するソケットを削除する機能を使用するかどうかを設定します。

設定値	説明
TRUE	タイムアウトが発生するソケットを削除します
FALSE	タイムアウトが発生するソケットを削除しません

- enableJavaSocketKeepAlive = Boolean

– デフォルト値 : FALSE

– データの送信が中止された場合、コネクションの状態を確認するオプションを使用するかどうかを設定します。

設定値	説明
TRUE	コネクションの状態を確認します
FALSE	コネクションの状態を確認しません

- useRollingDown = Boolean

– デフォルト値 : FALSE

– Rolling Downを使用するかどうかを設定します。

設定値	説明
TRUE	Rolling Downを使用します
FALSE	Rolling Downを使用しません

- jeus.servlet.webt.autoClose.enable = Boolean

– デフォルト値 : FALSE

– 使用後に返されなかったコネクションを自動でWebTConnectionPoolに返す機能を使用するかどうかを決定します。

設定値	説明
TRUE	コネクションを自動で返す機能を使用します
FALSE	コネクションを自動で返す機能を使用しません

● enableConnectionPool = Boolean

– デフォルト値 : FALSE

– WebTConnectionPoolを使用するかどうかを決定します。

設定値	説明
TRUE	WebTConnectionPoolを使用します
FALSE	WebTConnectionPoolを使用しません

● connectionPool.groups = String

– デフォルト値 : NULL

– 設定で使用するコネクション・プールの名前を指定します。指定された名前は connectionPool.<groupName>.connectTimeoutの<groupName>で使用されます。

● connectionPool.<groupName>.connectTimeout = int

– デフォルト値 : 20(単位:秒)

– WebTDataSourceからコネクションを取得する際に使用されるタイムアウト時間を指定します。設定された時間内にコネクションを取得した場合や新規作成できなかった場合、例外を返します。

● connectionPool.<groupName>.cluster = String

– デフォルト値 : NULL

– クラスタで管理するコネクション・グループの名前をコンマ(,)で区分して指定します。

● connectionPool.<groupName>.hostAddr = String

- デフォルト値：NULL
- 接続するTmaxサーバーのアドレスを指定します。
- connectionPool.<groupName>.hostPort = String
 - デフォルト値：NULL
 - 接続するTmaxサーバーのポートを指定します。
- connectionPool.<groupName>.hostBackupAddr = String
 - デフォルト値：NULL
 - 接続するTmaxバックアップ・サーバーのアドレスを指定します。
- connectionPool.<groupName>.hostBackupPort = String
 - デフォルト値：NULL
 - 接続するTmaxバックアップ・サーバーのポートを指定します。
- connectionPool.<groupName>.encryption = Boolean
 - デフォルト値：FALSE
 - Tmaxと通信する際に暗号化機能を使用するかどうかを設定します。

設定値	説明
TRUE	暗号化機能を使用します
FALSE	暗号化機能を使用しません

- connectionPool.<groupName>.check = Boolean
 - デフォルト値：FALSE
 - livenessチェックによるコネクションの状態チェックを行うかどうかを設定します。

設定値	説明
TRUE	livenessチェックをします
FALSE	livenessチェックをしません

- `connectionPool.<groupName>.userName = String`
 - デフォルト値 : NULL
 - ユーザー・セキュリティー認証のためにユーザーの名前を設定します。

- `connectionPool.<groupName>.userPasswd = String`
 - デフォルト値 : NULL
 - ユーザー・セキュリティー認証のためにユーザーの暗号を設定します。

- `connectionPool.<groupName>.domainName = String`
 - デフォルト値 : NULL
 - ドメイン・セキュリティー認証のためにドメインの名前を設定します。

- `connectionPool.<groupName>.domainPasswd = String`
 - デフォルト値 : NULL
 - ドメイン・セキュリティー認証のためにドメインの暗号を設定します。

- `connectionPool.<groupName>.fair.provision = Boolean`
 - デフォルト値 : TRUE
 - WebTConnectionPoolからコネクションを取得するポリシーを決定します。

設定値	説明
TRUE	使用していないコネクションを取得します
FALSE	最近使用したコネクションを取得します

- `connectionPool.<groupName>.initCapacity = int`
 - デフォルト値 : 5
 - WebTConnectionPoolのデフォルトのコネクション数を指定します。

- `connectionPool.<groupName>.maxCapacity = int`

- デフォルト値 : 3
 - コネクションが不足して新しくコネクションを作成する場合、新規作成するコネクションの数を設定します。
- connectionPool.<groupName>.maxIdleTime = int
 - デフォルト値 : 600 (: Millisecond)
 - コネクションを削除する基準となる時間を設定します。

WebTConnectionPoolのコネクション数がconnectionPool.<groupName>.initCapacityに設定された数より大きい場合、connectionPool.<groupName>.maxIdleTimeに設定された時間内に使用されなかったコネクションをWebTConnectionPoolから削除します。
- connectionPool.<groupName>.tpTimeout = int
 - デフォルト値 : 60
 - Tmaxサービスを呼び出す際のタイムアウト時間を指定します。
- connectionPool.<groupName>.txTimeout = int
 - デフォルト値 : 180
 - Tmaxトランザクション・サービスを要求する際のタイムアウト時間を指定します。
- connectionPool.<groupName>.txBlockTimeout = int
 - デフォルト値 : 60
 - Tmaxサーバーにコミットまたはロールバックを要求する際のタイムアウト時間を指定します。
- connectionPool.<groupName>.header.type = default | extendedV4
 - デフォルト値 : 対応できるタイプ
 - Tmaxと通信する際のプロトコル・タイプを決定します。groupNameと設定されたコネクション・グループで作成するコネクションに適用されるプロトコル・タイプです。

設定値	説明
default	Tmax 3.xで使用するプロトコルです
extendedV4	Tmax 4.xで使用するプロトコルです。Tmax 4.xは下位プロトコルをサポートするため、defaultと設定しても互換されます

- connectionPool.<groupName>.eventSvcType = all | broadcast | sendtocli | notify | acall

- デフォルト値 : NULL

- 使用するイベントを設定します。2種類以上のvalueを同時に設定する場合はコンマ(,)で区分します。(例: broadcast,notify)

設定値	説明
all	すべてのイベントを使用します
broadcast	broadcastタイプのイベントを使用します
sendtocli	sendtocliタイプのイベントを使用します
notify	notifyタイプのイベントを使用します
acall	acallタイプのイベントを使用します
none	イベントを使用しません

- connectionPool.<groupName>.eventHandler = String

- デフォルト値 : NULL

- イベントを受信する際に呼び出されるインターフェースを実装したオブジェクトの名前を設定します。

- connectionPool.<groupName>.type = shared | clustered

- デフォルト値 : shared

- WebTConnectionPoolのタイプを指定します。

設定値	説明
shared	一般的な場合に指定します
clustered	WebTDataSourceのクラスタリングを使用する場合は、tmax-delegation-sourceとconnection-timeoutを<property>タグで定義します

- tmax.webt.xid.log = boolean

- デフォルト値 : false

- WebTトランザクションを処理する際にWebTのログレベルがinfoの場合も、xa_start、xa_prepare、xa_commit、xa_rollbackがログに出力されます。

- webt.mbean = boolean

- デフォルト値 : false
- WebT AdminのためのMBeanを登録するかどうかを指定します。Webt Admin機能はJEUS 6.0 Fix#7以上でサポートします。

以下は、webt.propertiesを設定する例です。

```
#####
# logging related parameters.
#####
# set log level valid values are none, info, debug. default is none.
log.level=debug

# set directory in which the log file places. if not set, log is
# printed to standard out
log.dir=D:\\

# set the name of the log file. default is webt.log
log.file=webt.log

# set log buffering size. default is 0
log.bufsize=1024

# tmax.webt.xid.log=true

#####
# FDL related parameters
#####
# set the fdl file.
fdl.file=c:\\tmax\\sample\\fdl\\tmax.fdl

# set application wide default character set. default is system default
defaultCharset=euc-kr

#####
# monitoring related parameters
#####
# enable/disable alive check. default is disable(false)
monitoring.pool.checkAlive=true

# set monitoring interval. default is 60sec
```

```
monitoring.pool.interval=5

# set monitoring log file valid-days.default is -1
log.valid.days=1

#####
# connection pool related parameters
#####
# Tmax 3.11.x extended Header size
headerType=default|extendedV4

# enable/disable connection pool. default is disable(false)
enableConnectionPool=true

# WebtConnectionGroup name list
connectionPool.groups=tmax1

# set connection group type valid values are shared, non-shared,
non-shared2. default is shared
connectionPool.tmax1.type=shared

# set Tmax Server Address.
connectionPool.tmax1.hostAddr=192.168.1.123

# set Tmax Server Port.
connectionPool.tmax1.hostPort=8888

# set Backup Tmax Server Address.
connectionPool.tmax1.hostBackupAddr=61.77.153.1

# set Backup Tmax Server Port.
connectionPool.tmax1.hostBackupPort=8889

# set default user name for security.
connectionPool.tmax1.userName=tmax

# set default user password for security.
connectionPool.tmax1.userPasswd=1234

# set default domain name for security.
connectionPool.tmax1.domainName=choco

# set default domain password for security.
connectionPool.tmax1.domainPasswd=1234
```

```
# set initial pool size. default is 10
connectionPool.tmax1.initCapacity=1

# set max pool size. default is 20
connectionPool.tmax1.maxCapacity=10

# set increment step size. default is 5
connectionPool.tmax1.incrementRate=2

# set connection idle time. default is 60sec
connectionPool.tmax1.maxIdleTime=60

# set tptimeout.
connectionPool.tmax1.tpTimeout=40

# set txtimeout.
connectionPool.tmax1.txTimeout=10

# set connection timeout.
connectionPool.tmax1.connectTimeout=10

# enable to event service.

#connectionPool.tmax1.enableEvent=true

# set event service Type.

#connectionPool.tmax1.eventSvcType=all

# set event handler object.

#connectionPool.tmax1.eventHandler=tcpservlet.event.GenericEvent
jeus.servlet.webt.autoClose.enable=true
webt.mbean=true
```

2.1.3. それ以外の環境設定

ローリング・ダウン機能

WebTクライアントでJEUSMain.xmlとクライアントを実行する際に以下のWebTアプリケーションを設定すると、ローリング・ダウン機能を使用できます。

- JEUSMain.xml
 - <command-option>-Dtmax.webt.xid.log=true | false</command-option>

WebTトランザクションを処理する際、WebTのログレベルがinfoの場合にも、xa_start、xa_prepare、xa_commit、xa_rollbackがログに出力されます。(デフォルト値：false)

以下は、JEUSMain.xmlにローリング・ダウン機能を設定する例です。

```
<command-option>
  -DUSE_ROLLING_DOWN=true
</command-option>
```

- WebTアプリケーション

```
System.setProperty("USE_ROLLING_DOWN", "true")
```

2.2. JTmaxの環境設定

本節では、JEUSでJTmaxを起動するためにJEUSMain.xmlを設定する方法について説明します。

2.2.1. JEUSMain.xmlの設定

以下はJEUSMain.xmlを設定する例です。JTmaxを使用するには、<external-resource>タグ内部に<class-name>tmax.webt.external.JTmaxResource</class-name>を定義する必要があります。ただし、以下の例はEJBアプリケーションのclassloading modeがSHAREDの場合にのみ使用可能です。

```
<node>
  <name>nodename</name>
  <engine-container>
    <name>container1</name>
    . . .
    <external-resource>
      <name>jtmax</name>
      <class-name>tmax.jtmax.external.JTmaxResource</class-name>
      <property>
        <key>log-file-name</key>
        <value>c:\jtmax.log</value>
      </property>
      <property>
        <key>log-level</key>
        <value>debug</value>
      </property>
      <property>
        <key>log-valid-day</key>
        <value>1</value>
      </property>
      <property>
```

```
<key>fdl-file</key>
<value>c:\\tmax\\sample\\fdl\\tmax.fdl</value>
</property>
<property-group>
  <name-prefix>jtmax1</name-prefix>
  <property>
    <key>listen-port</key>
    <value>6735</value>
  </property>
  <property>
    <key>min</key>
    <value>2</value>
  </property>
  <property>
    <key>max</key>
    <value>4</value>
  </property>
  <property>
    <key>step</key>
    <value>2</value>
  </property>
  <property>
    <key>period</key>
    <value>600000</value>
  </property>
  <property>
    <key>mbean</key>
    <value>true</value>
  </property>
  <property-group>
    <name-prefix>ejb-service</name-prefix>
    <property>
      <key>name</key>
      <value>ejb01</value>
    </property>
    <property-group>
      <name-prefix>GSVC01</name-prefix>
      <property>
        <key>export-name</key>
        <value>echotest</value>
      </property>
      <property>
        <key>method-name</key>
        <value>setStringEcho</value>
      </property>
      <property>
        <key>method-signature</key>
```

```

        <value>string</value>
    </property>
</property-group>
<property-group>
    <name-prefix>GSVC02</name-prefix>
    <property>
        <key>export-name</key>
        <value>echotest</value>
    </property>
    <property>
        <key>method-name</key>
        <value>setIntEcho</value>
    </property>
    <property>
        <key>method-signature</key>
        <value>int</value>
    </property>
</property-group>
</property-group>
</external-resource>
    . . . .
</engine-container>
</node>

```

<property-group>/<property>の属性

以下は、JTmaxを使用するためにJEUSMain.xmlに設定するタグについての説明です。<key>タグには属性名を、<value>タグには適用可能な値を設定します。

<key>	区分	<value>
listen-port	設定値	int
	デフォルト値	0
	説明	コネクションを設定するポートを指定します
min	設定値	int
	デフォルト値	2
	説明	要求を処理するデフォルトのハンドラ数を指定します
max	設定値	int
	デフォルト値	4
	説明	要求を処理するハンドラの最大数を設定します
step	設定値	int

<key>	区分	<value>
	デフォルト値	2
	説明	要求に対するハンドラが不足した場合、追加で作成するハンドラの数を設定します
period	設定値	int
	デフォルト値	600000(単位:ミリ秒)
	説明	ハンドラを削除する基準となる時間を指定します。ハンドラの数がminに設定した数より大きい場合、ハンドラを使用しない時間がperiodで指定された時間を超えた場合、ハンドラを削除します
log-file-name	設定値	String
	デフォルト値	NULL
	説明	ログ・ファイルの相対パスまたは絶対パスを指定します
log-level	設定値	none info debug
	デフォルト値	info
	説明	ログレベルを指定します
log-file-format	設定値	string(形式)
	デフォルト値	MMddyyyy
	説明	log-valid-dayが1以上の場合に作成されるログファイル名の形式を指定します
log-buffer-size	設定値	int
	デフォルト値	512
	説明	ログを残す際に使用されるバッファのサイズを指定します
log-valid-day	設定値	int
	デフォルト値	-1
	説明	ログ・ファイルを使用する場合、何日周期でファイル名を更新するのかを指定します。0や負数を指定した場合、1つのファイルを使用します
fdl-file	設定値	string
	デフォルト値	NULL
	説明	Tmax FDLバッファを使用するためのフィールド表の位置を指定します。fdlc 6.0でビルドしたバイナリは使用できません。fdlc 6.0以下のバージョンでビルドしたファイルを使用してください
default-charset	設定値	string
	デフォルト値	NULL
	説明	Tmaxバッファの内容に適用される文字セットを指定します
mbean	設定値	true false

<key>	区分	<value>
	デフォルト値	false
	説明	JTmax AdminのためのMBeanを登録するかどうかを指定します。JTmax Admin機能はJEUS v6.0 Fix#7以上でサポートします

2.2.2. EJBの設定

EJBを使用するには、JEUSMain.xmlでEJBサービス情報を登録します。Tmaxクライアントで使用するサービス名とJEUS EJBサービスのメソッド名を対応させて設定します。Tmaxクライアントのサービス名とJEUS EJBサービスのメソッド名が対応する部分は以下のとおりです。

```
<property-group>
  <name-prefix>ejb-service</name-prefix>
  <property>
    <key>name</key>
    <value>ejb01</value>
  </property>
  <property-group>
    <name-prefix>GSVC02</name-prefix>
    <property>
      <key>export-name</key>
      <value>echotest</value>
    </property>
    <property>
      <key>method-name</key>
      <value>setIntEcho</value>
    </property>
    <property>
      <key>method-signature</key>
      <value>int</value>
    </property>
  </property-group>
</property-group>
```

上で<name-prefix>タグに設定されたGSVC02はTmaxで呼び出されるEJBサービス名で、実質的なEJBサービスはIntEchoです。

<property-group>/<property>属性の<key>タグに該当する意味は以下のとおりです。

<key>の属性	区分	<value>の説明
export-name	設定値	string
	デフォルト値	NULL
	説明	EJB名を意味します
method-signature	設定値	char long int double float string bytes

<key>の属性	区分	<value>の説明
	デフォルト値	NULL
	説明	サービスが変数として受け取るパラメータの形式を指定します。2種類以上の値を同時に設定する際はコンマ(,)で区分します(例: int,float)
max	設定値	int
	デフォルト値	4
	説明	要求を処理するハンドラの最大数を設定します

EJBサービスの環境設定

EJBサービスを呼び出すためのTmax環境設定ファイルにSERVICEセクションとGATEWAYセクションを設定します。

● SERVICEセクション

以下は、SERVICEセクションを設定する文法と項目について説明します。

```
*SERVICE
ServiceName SVRNAME = ServerName
```

項目	説明
ServiceName	Tmaxクライアントが呼び出すサービスの名前です。WebTサーバーを使用する場合、JEUSから提供されるEJBサービスのエイリアスです。実際に実行するJEUSのEJBサービスの内容はJEUSサービス・ファイルに設定します
SVRNAME	該当サービスを提供するサーバー・プログラム名です。WebTサーバーを使用する場合、GATEWAYセクションに設定されるJAVAGWの名前(GatewayName)を指定します

● GATEWAYセクション

以下は、GATEWAYセクションを設定する文法と項目についての説明です。

```
*GATEWAY
GatewayName GWTYPE = GWType ,
PORTNO = PortNumber ,
RGWADDR = RemoteIPAddress ,
RGWPORTNO = RemotePortNumber ,
NODENAME = NodeName
```

項目	説明
GatewayName	Tmaxサーバー・プログラムや他のゲートウェイと区分するために使用するゲートウェイの固有名称です

項目	説明
GWTYPE	Tmaxで使用するゲートウェイの種類を指定します。JAVAGWゲートウェイを使用するには「JEUS」と設定します
PORTNO	TmaxとJAVAGWゲートウェイが通信するために使用するポート番号です
RGWADDR	ゲートウェイが接続するサーバーのIPアドレスです。JAVAGWゲートウェイが接続するJEUSサーバーのIPアドレスを設定します
RGWPORTNO	ゲートウェイが接続するサーバーのポート番号です。JEUSのJTmax環境設定ファイルに設定されたコネクション・ポート番号を設定します
NODENAME	該当ローカルノードの名前です

以下は、EJBサービスを呼び出すための設定の例です。

```
*SERVICE
GSVC01 SVRNAME = JAVAGW
GSVC02 SVRNAME = JAVAGW

*GATEWAY
JAVAGW GWTYPE = JEUS,
PORTNO = 6734,
RGWADDR = "127.0.0.1",
RGWPORTNO = 6735,
NODENAME = node_name
```

上のように作成された環境設定ファイルをsample.mとした場合、環境設定ファイルを作成後、JEUSを起動した状態でTmaxサーバーにて以下のようなコマンドでTmaxを起動すると、JTmaxへ接続されます。

```
cfl -i sample.m
```

参考

サーバーを使用するためのTmax環境設定ファイルの設定は、ゲートウェイを使用するための設定と同じです。設定方法の詳細については『Tmax 運用ガイド』を参照してください。

第3章 WebT API

本章では、WebTからTmaxサーバーに接続し、サービスを呼び出すなどの動作をするのに必要なクラスと、各クラスで使用できるフィールドおよびメソッドについて説明します。

3.1. 概要

以下は、WebTからTmaxサーバーに接続し、サービスを呼び出すなどの動作をするのに必要なクラスの一覧です。各クラスについての詳細は各節を参照してください。

- 接続関連クラス

クラス	説明
WebtConnection	TmaxサーバーとWebT間の接続を行うクラスです
WebtConnectionPool	WebtConnectionオブジェクトを管理するクラスです
WebtConnectionGroup	コネクションをプーリングする機能を担当するクラスです

- サービスの呼び出しデータ関連クラス

クラス	説明
WebtAttribute	TmaxサーバーにWebtRemoteServiceのTP関数を呼び出す際に使用するフラグ情報を保持しているクラスです
WebtBuffer	WebTとTmaxサーバー間で送受信するユーザー・データを管理するクラスです
WebtCarrayBuffer	WebtBufferを継承したクラスで、CARRAYタイプのデータを持つクラスです
WebtStringBuffer	WebtBufferを継承したクラスで、STRINGタイプのデータを持つクラスです

- サービスの呼び出し関連クラス

クラス	説明
WebtRemoteService	WebtConnectionクラスおよびその子クラスを通じてAPIを提供し、Tmaxサービスを呼び出す際に必要な属性およびサービス名などを管理するクラスです
WebtRQService	WebtRemoteServiceを継承したクラスで、RQを管理するサービスを使用するためのクラスです
WebtDialogueService	WebtRemotedServiceを継承したクラスで、会話型サービスを提供するクラスです

- フィールドデータ関連クラス

クラス	説明
WebtFieldElement	WebtFieldのフィールドキーに対するフィールド・データを格納しているインターフェースです
WebtFDLKeyTable	Tmaxサービス呼び出す際に使用するバッファ型がWebtBuffer.BT_FIELDの場合、フィールドキーを管理するクラスです
WebtFieldSet	Tmaxサービスに使用するバッファ型がWebtBuffer.BT_FIELDの場合、より簡単にフィールド・データの要素を操作できるようにするユーティリティー・クラスです
WebtField	フィールド・タイプ・バッファで使用するフィールドキーとそのフィールドキー値に該当する多数のデータ(WebtFieldElement)を管理する抽象クラスです
WebtFieldBuffer	WebtBufferの下位クラスで、WebtFieldタイプのデータを保持しているWebtBufferオブジェクトです

- イベント・メッセージ処理関連クラス

クラス	説明
WebtEventConnecion	Tmaxサーバーまたは他のクライアントが送信したイベント・メッセージを処理するクラスです
WebtEventHandler	Tmaxサーバーから受信した非要求メッセージを処理する際に使用するインターフェースです

- その他のクラス

クラス	説明
WebtTransaction	複数のサービスを1つの作業単位にまとめて処理できるようにAPIを提供し、例外の発生を処理するクラスです
WebtSystem	WebTシステムのバージョン、文字セット、フィールドキー・テーブル情報を管理するクラスです

3.2. 接続関連クラス

3.2.1. WebtConnection

TmaxサーバーとWebT間の接続を行うクラスです。Tmaxサーバーが提供するサービスを利用するには、WebtConnectionを必ず使用してネットワークに接続後、サービスを要求します。WebtConnectionクラスは、Tmaxサーバーに接続するための情報を設定し、管理します。

構築子

WebtConnectionの構築子は、パラメータhostAddrに設定されたIPアドレスをもつTmaxサーバーの、パラメータhostPortに設定されたポート番号に接続するコネクションを作成します。

以下は、WebtConnectionクラスの構築子についての説明です。

- 構築子1

- プロトタイプ

```
WebtConnection(String hostAddr,  
                int hostPort,  
                [boolean doConnect,]  
                [int conntimeout])
```

- パラメータ

パラメータ	説明
hostAddr	接続するメインTmaxサーバーのIPアドレスです
hostPort	接続するメインTmaxサーバーのポート番号です
doConnect	コネクションを作成する際、Tmaxサーバーに接続を行うかどうかを設定します。 <ul style="list-style-type: none">– TRUE: 接続します– FALSE: 接続しません
conntimeout	Tmaxサーバーに接続する際のタイムアウト時間を設定します。(単位: 秒) 時間内に接続できなかった場合、tperrnoをTPETIMEとする例外が発生します

- 構築子2

- プロトタイプ

```
WebtConnection(String hostAddr,
                int hostPort,
                String backupAddr,
                int backupPort,
                [boolean doConnect,]
                [int conntimeout])
```

– パラメータ

パラメータ	説明
hostAddr	接続するメインTmaxサーバーのIPアドレスです
hostPort	接続するメインTmaxサーバーのポート番号です
backupAddr	hostAddr:hostPortへの接続が失敗した場合に接続するバックアップ・サーバーのIPアドレスです
backupPort	hostAddr:hostPortへの接続が失敗した場合に接続するバックアップ・サーバーのポート番号です
doConnect	コネクションを作成する際にTmaxサーバーに接続するかどうかを設定します。 – TRUE: 接続します – FALSE: 接続しません
conntimeout	Tmaxサーバーに接続する際のタイムアウト時間を設定します。(単位: 秒) 時間内に接続できなかった場合、tperrnoをTPETIMEとする例外が発生します

● 構築子3

– プロトタイプ

```
WebtConnection(String hostAddr,
                int hostPort,
                String username,
                String userpwd,
                String domainname,
                String domainpwd)
```

– パラメータ

パラメータ	説明
hostAddr	接続するメインTmaxサーバーのIPアドレスです
hostPort	接続するメインTmaxサーバーのポート番号です

パラメータ	説明
username	ユーザー・セキュリティー認証のためのユーザー名です
usernpwd	ユーザー・セキュリティー認証のためのユーザー・パスワードです
domainname	ドメイン・セキュリティー認証のためのドメイン名です
domainpwd	ドメイン・セキュリティー認証のためのドメイン・パスワードです

- 例外

例外	説明
WebtIOException	コネクションを初期化中にネットワーク・エラーが生じた場合や、タイムアウト時間内にサーバーに接続できなかった場合に発生します
WebtServiceException	ネットワークの入出力やトランザクション以外のサービス・エラーが発生した場合に発生します

メソッド

以下は、WebtConnectionクラスのメソッド一覧です。

- boolean checkConnection(int timeout)
 - クライアントの接続状態をチェックする関数です。
 - パラメータ

パラメータ	説明
timeout	現在は使用しません

- boolean close()
 - Tmaxとのソケット接続を終了します。コネクションがコネクション・プールで管理されている場合は接続を終了せず、コネクション・プールに戻して再使用します。
- void connect()
 - Tmaxと接続されていない場合、接続を行います。
 - 例外

例外	説明
WebtIOException	コネクションを初期化中にネットワーク・エラーが生じた場合や、タイムアウト時間内にサーバーに接続できなかった場合に発生します
WebtServiceException	ネットワークの入出力やトランザクション以外のサービス・エラーが発生します

- String getAddress()
 - メインTmaxサーバーのアドレスを返します。

- int getBackupPort()
 - メインTmaxサーバーのポート番号を返します。

- String getBackupAddress()
 - バックアップTmaxサーバーのアドレスを返します。

- String getConnectionId()
 - コネクションのIDを返します。

- String getCurrentHostAddress()
 - 現在接続されているTmaxサーバーのIPアドレスを返します。

- int getCurrentHostPortNo()
 - 現在接続されているTmaxサーバーのポート番号を返します。

- String getDomainName()
 - メインTmaxサーバーに接続されているクライアントの名前を返します。

- String getGroupName()
 - コネクションが属しているコネクション・グループの名前を返します。

- int getPort()
 - メインTmaxサーバーのポート番号を返します。

- `int getTPtimeout()`
 - 現在設定されているブロッキング・タイムアウト時間を返します。(単位: 秒)

- `WebTransaction getTransaction()`
 - コネクションが使用しているWebTransactionオブジェクトを返します。

- `int getTxBlocktimeout()`
 - 現在設定されているトランザクションのコミットとロールバックのタイムアウト時間を返します。

- `getUserName()`
 - メインTmaxサーバーのユーザー認証セキュリティのアカウント名を返します。

- `boolean isAlive()`
 - コネクションが活性化されているかどうかをチェックします。

- `void setTPtimeout(int sec)`
 - ブロッキング・タイムアウト時間を秒単位で設定します。

- `void setTxBlocktimeout(int txBlockTimeout)`
 - トランザクションのコミットとロールバックのタイムアウト時間を設定します。

- `void setTxtimeout(int txTimeout)`
 - トランザクションのタイムアウト時間を設定します。

- `void startReader()`
 - コネクションを通じたメッセージ処理スレッドを開始します。

- `void stopReader()`
 - コネクションを通じたメッセージ処理スレッドを終了します。

例

以下は、WebtConnectionオブジェクトを使用して接続を設定し、サービスを取得する例です。サービスを設定する方法は「[3.4.1. WebtRemoteService](#)」を参照してください。

```
...
WebtConnection conn = new WebtConnection("61.77.153.1",8888);
WebtRemoteService svc = new WebtRemoteService("TOUPPER",conn);
...
conn.close()
```

3.2.2. WebtConnectionPool

WebtConnectionPoolは、Tmaxサーバーへのネットワーク接続であるWebtConnectionオブジェクトを管理します。WebtConnectionPoolクラスを利用すると、Tmaxサービスが要求されるたびにWebtConnectionオブジェクトを新しく作成するのではなく、以前に使用したWebtConnectionオブジェクトを再使用できます。そのため、Tmaxサーバーにネットワーク接続を設定および終了するのに消費されるリソースおよび時間を節約できます。

WebtConnectionPoolは以下のような機能があります。

- 1つ以上のTmaxサーバーと接続されたコネクション・プールを管理できます。
- コネクション・プールをモニタリングし、指定された時間に遊休状態となっている接続を解除するか、接続が終了した接続を探して新しい接続を設定します。

createGroup

createGroupメソッドはWebtConnectionGroupクラスのオブジェクトを作成します。

以下は、WebtConnectionGroupの作成メソッドについての説明です。

- 作成メソッド1
 - プロトタイプ

```
public static void createGroup(String groupname,
                               String hostAddr,
                               int hostPort,
                               [String username,]
                               [String userpwd,]
                               [String domainname,]
                               [String domainpwd,]
                               int init,
```

```
int max,  
int inc)
```

– パラメータ

パラメータ	説明
groupname	コネクション・グループの名前です。同一名をもつコネクション・グループが存在した場合、コネクション・グループを作成しません
hostAddr	接続するTmaxサーバーのホスト名です
hostPort	接続するTmaxサーバーのホスト番号です
username	ユーザー・セキュリティー認証のためのユーザー名です
userpwd	ユーザー・セキュリティー認証のためのユーザー・パスワードです
domainname	ドメイン・セキュリティー認証のためのドメイン名です
domainpwd	ドメイン・セキュリティー認証のためのドメイン・パスワードです
init	コネクションの初期接続数です
max	コネクションの最大接続数です
inc	使用可能なコネクションがない状態で要求を受信した際に新規作成するコネクション数です

● 作成メソッド2

– プロトタイプ

```
public static void createGroup(String groupname,  
                               String hostAddr,  
                               int hostPort,  
                               String backupAddr,  
                               int backupPort,  
                               [String username,]  
                               [String userpwd,]  
                               [String domainname,]  
                               [String domainpwd,]  
                               int init,  
                               int max,  
                               int inc,  
                               [int idle])
```

– パラメータ

パラメータ	説明
groupname	コネクション・グループの名前です。同一名をもつコネクション・グループが存在する場合、コネクション・グループを作成しません
hostAddr	接続するTmaxサーバーのホスト名です
hostPort	接続するTmaxサーバーのポート番号です
backupAddr	接続するTmaxサーバーのバックアップ・ホスト名です
backupPort	接続するTmaxサーバーのバックアップ・ポート番号です
username	ユーザー・セキュリティ認証のためのユーザー名です
userpwd	ユーザー・セキュリティ認証のためのユーザー・パスワードです
domainname	ドメイン・セキュリティ認証のためのドメイン名です
domainpwd	ドメイン・セキュリティ認証のためのドメイン・パスワードです
init	コネクションの初期接続数です
max	コネクションの最大接続数です
inc	使用可能なコネクションが状態で要求を受信した際に新規作成するコネクションの数です
idle	コネクションで許容された最長アイドルタイムです。(単位:秒) WebtConnectionPoolMonitorでアイドルタイムをチェックする際に使用します

- 例外

例外	説明
WebtIOException	コネクション・グループでコネクションの作成に失敗した場合に発生します
WebtServiceException	コネクション・プールのcreateGroupメソッドのパラメータgroupNameがNULLの場合に発生します
WebtException	WebTを使用中に発生する例外で、WebtIOExceptionとWebtServiceException以外の例外はすべてWebtExceptionに含まれます

メソッド

- public static void destroy()
 - すべてのコネクション・グループを削除します。
- public static void destroyGroup(String name)

- パラメータnameと同じ名前のコネクション・グループを削除します。該当コネクション・グループに属しているすべてのコネクションの接続が終了します。また、該当コネクション・グループより索引番号が大きいコネクション・グループの索引番号が1ずつ減少します。

- パラメータ

パラメータ	説明
name	コネクション・グループの名前です

- public static void endWait()

- コネクションを継続維持します。

- public static WebtConnection getConnection(String groupname)

- パラメータgroupnameと同じ名前のコネクション・グループに属するコネクションを返します。タイムアウト時間内にコネクションを取得できない場合、WebtException例外が発生します。

- パラメータ

パラメータ	説明
groupname	コネクション・グループの名前です

- 例外

例外	説明
WebtIOException	コネクション・グループに使用可能なコネクションがない場合や、活性化されていない場合に発生します
WebtServiceException	コネクション・グループの名前がNULLの場合や、タイムアウト時間を超えた場合にWebtException例外が発生します

- public static WebtConnection getConnection(String groupname, long timeout_mili)

- パラメータgroupnameと同じ名前のコネクション・グループに属するコネクションを返します。パラメータtimeout_miliに設定されたタイムアウト時間内にコネクションを取得できない場合、WebtException例外が発生します。

- パラメータ

パラメータ	説明
groupname	コネクション・グループの名前です

パラメータ	説明
timeout_mili	コネクションを取得する際のタイムアウト時間です(単位:ミリ秒)

- 例外

例外	説明
WebtIOException	コネクション・グループに使用可能なコネクションがない場合や活性化されていない場合に発生します
WebtServiceException	コネクション・グループの名前がNULLの場合やタイムアウト時間を越えた場合にWebtException例外が発生します

● public static WebtConnectionGroup getGroup(String groupname)

- パラメータgroupnameに指定した値と同じ名前のコネクション・グループを返します。該当コネクション・グループが存在しない場合、NULLを返します。

- パラメータ

パラメータ	説明
groupname	コネクション・グループの名前です

● public static void startMonitor(int interval, boolean checkAlive, boolean failback)

- コネクション・プールをモニタリングするスレッドを作成します。checkAliveとcheckIdleが両方ともFALSEの場合は作成しません。

- パラメータ

パラメータ	説明
interval	モニタリング間隔を設定します(単位:秒)
checkAlive	コネクションが有効かどうかチェックし、コネクションが切れた場合は再接続します
failback	Trueと設定した場合、メインサーバーの状態をチェックし、異常があれば復旧します

● public static void stopMonitor()

- コネクション・プールをモニタリングすることを中止し、モニタリング・スレッドを終了します。

例

以下は、2つのTmaxサーバーで接続・プールを提供できるよう、createGroupメソッドを使用して接続・プールを初期化する例です。createGroupメソッドの1番目のパラメータgroupnameを使用して接続を取得します。

```
....
WebtConnectionPool.createGroup("tmax1", "192.168.128.1", 8888,2,10,2);
WebtConnectionPool.createGroup("tmax2", "192.168.128.2", 8888,1,10,2);
....
```

以下は、上の例で初期化した接続・プールからWebtConnectionオブジェクトを取得する例です。

```
try {
    // tmax1 group (192.168.128.1:8888からWebtConnectionオブジェクトを取得します。)
    WebtConnection con = WebtConnectionPool.getConnection("tmax1");
} catch (WebtIOException wioe) {
    System.out.println("fail to get WebtConnection of tmax1");
    return;
}
WebtRemoteService service = new WebtRemoteService("MYSVC", con);
....
// WebtConnectionオブジェクトの返却
WebtConnectionPool.putConnection(con);
```

3.2.3. WebtConnectionGroup

接続をプーリングする機能を行う抽象クラスです。同じIPアドレスとポート番号をもつ多数の接続のプーリングを管理します。

構築子

ユーザーが直接WebtConnectionGroupのオブジェクトを作成することはできず、WebtConnectionPoolのcreateGroup作成メソッドを使用してオブジェクトを作成できます。WebtConnectionGroupのオブジェクトの作成については、[「3.2.2. WebtConnectionPool」](#)のcreateGroupに関する説明を参照してください。

メソッド

以下は、WebtConnectionGroupクラスのメソッド一覧です。

- int available()
 - 現在接続・プールに残っている接続の数を返します。

- void destroy()

- 現在の接続・プールを削除します。

- WebtConnection getConnection()

- 指定されたタイムアウト時間中は接続を取得するために待機し、使用可能な接続を取得すれば返します。(デフォルト値:20秒)

- 例外

例外	説明
WebtException	接続・プールが活性化されなかった場合や指定された時間に接続を取得できなかった場合に発生します

- WebtConnection getConnection(long timeout)

- 指定されたタイムアウト時間中は接続を取得するために待機し、使用可能な接続を取得すれば返します。(デフォルト値:20秒)

- パラメータ

パラメータ	説明
timeout	接続を取得するために待機する時間です(単位:ミリ秒)

- 例外

例外	説明
WebtException	接続・プールが活性化されなかった場合や指定された時間内に接続を取得できなかった場合に発生します

- tmax.webt.io.WebtConnectionPoolInfo getConnectionPoolInfo()

- 接続・プールの情報を返します。

- String getName()

- 接続プールの名前を返します。

- boolean isActive()

- 現在接続・プールが活性かしているかどうかを確認します。

- int size()
 - 現在プールで管理中のコネクションの数を返します。

3.3. サービス呼び出し関連クラス

3.3.1. WebtAttribute

TmaxサーバーにWebtRemoteServiceのtpcall()、tpacall()、tpgetrply()などのTP関数を呼び出す際に使用するフラグ情報を保持しているクラスです。

構築子

WebtAttributeオブジェクトを作成します。以下はWebtAttribute構築子についての説明です。

- 構築子1
 - 属性が設定されていないWebtAttributeオブジェクトを作成します。
 - プロトタイプ

```
public WebtAttribute()
```

- 構築子2
 - flagに設定された属性をもつWebtAttributeオブジェクトを作成します。
 - プロトタイプ

```
public WebtAttribute(int flag)
```

- パラメータ

パラメータ	説明
flag	WebtAttributeクラスの各フィールドをフラグとして使用できます

フィールド

以下は、WebtAttributeクラスのフィールド一覧です。

- public static final int TPBLOCK

- WebtRemoteService.tpacallメソッドを呼び出す際に使用されます。
- TPBLOCK属性を設定した場合、サービスやサーバーの異常有無についての応答を受け取ることができません。TPBLOCK属性が設定され、WebtRemoteService.tpacallメソッドを呼び出すのに失敗すると、WebtServiceExceptionあるいはWebtIOExceptionが発生します。
- public static final int TPFUNC
 - WebtRQService.tpenqメソッドやWebtRQService.tpdeqメソッドを呼び出す際に使用されます。TPFUNC属性をサービス別RQデータを管理する際に使用されます。
- public static final int TPGETANY
 - WebtRemoteService.tpgetrplyメソッドを呼び出す際に使用されます。
 - TPGETANY属性が設定され、WebtRemoteService.tpgetrplyメソッドを呼び出す際に記述子(cd)値に一致するWebtBufferが受信されなかった場合、tpgetrplyのパラメータとして送信される記述子の値は無視され、先に受信したWebtBufferを返します。
- public static final int TPNOBLOCK
 - WebtRemoteService.tpgetrply()メソッドで主に使用されます。
 - TPNOBLOCK属性が設定され、TPメソッドを呼び出す際、ネットワークがブロックされた場合はWebtServiceException(tperrno=TPETIME)が発生します。
- public static final int TPNOCHANGE
 - WebtRemoteServiceのtpcallメソッドとtpgetrplyメソッドを呼び出す際に使用されます。
 - TPNOCHANGE属性が設定されると、tpcallメソッドとtpgetrplyメソッドを呼び出す際に指定した受信バッファ型と実際に受信したWebtBufferのバッファ型が異なる場合はWebtServiceException(tperrno=TPEOTYPE)が発生します。
- public static final int TPNOFLAGS
 - 属性が何も設定されていないことを意味します。
- public static final int TPNOREPLY
 - TPNOREPLY属性が設定された場合、要求を送っても応答を受け取らないことを意味します。TPNOREPLY属性が設定され、WebtRemoteService.tpacallメソッドが正常に完了した場合、記述子として0が返されます。この記述子はWebtRemoteService.tpgetrplyメソッドで使用できません。

- WebtRemoteService.tpacallメソッドがトランザクション状態の場合は必ず応答を受け取る必要があるため、TPNOREPLY属性を使用できません。TPNOREPLY属性を使用するには、TPNOTRAN属性と一緒に使用し、トランザクションの領域(Transaction Boundary)から除外させます。

- public static final int TPNOTIME
 - TPNOTIME属性を設定した場合、Tmaxサーバー設定ファイルに設定されたBLOCKTIME値やWebtConnection.setTPtimeout(int)メソッドによって設定されたブロッキング・タイムアウト時間が無視され、無限大となります。つまり、応答が来るまで待機します。

 - トランザクション領域に属しているtpcall、tpacall、tpgetrplyなどのメソッドにTPNOTIME属性を設定した場合、トランザクション・タイムアウト時間が適用され、TPNOTIME属性は無視されます。

- public static final int TPNOTRAN
 - WebtRemoteServiceのtpcall、tpacallメソッドで使用します。WebtTransactionによってトランザクション領域に属しているtpcall、tpacallメソッドであっても、TPNOTRAN属性が設定されている場合はトランザクション領域から除外されます。

- static final int TPRETRY
 - Javaゲートウェイでユーザー・メソッドを設定する属性です。

- public static final int TPRQS
 - WebtRQService.tpenqメソッドやWebtRQService.tpdeqメソッドの呼び出しに使用されます。

 - TPRQS属性を設定した場合、RQサービスを使用する際、Request Queueに記録してReply Queueからデータを受け取ります。

参考

各属性に関する詳細については『Tmax アプリケーション開発ガイド』を参照してください。

メソッド

以下は、WebtAttributeクラスのメソッド一覧です。

- public void clearAttribute()
 - すべての属性を解除します。

- `public WebtAttribute duplicate()`
 - `WebtAttribute`オブジェクトをコピーします。
- `public boolean getAttribute(int attr)`
 - 特定属性の設定有無をチェックします。パラメータ`attr`に指定された属性がすべて設定されている場合は`TRUE`を返し、そうでない場合は`FALSE`を返します。
 - パラメータ

パラメータ	説明
<code>attr</code>	<code>TPNOBLOCK</code> 、 <code>TPNOREPLY</code> 、 <code>TPNOTRAN</code> 、 <code>TPNOTIME</code> 、 <code>TPGETANY</code> 、 <code>TPNOCHANGE</code> 、 <code>TPBLOCK</code> 、 <code>TPRQS</code> 、 <code>TPFUNC</code> のうち1つの値を設定します。 それ以外の値の場合は無視し、1つまたは1つ以上の値を <code>OR()</code> 演算で設定が可能です

- `public String toString()`
 - 該当オブジェクトを文字列で表現した値を返します。
- `public boolean isTPBLOCK()`
 - `TPBLOCK`属性の設定有無をチェックします。
- `public boolean isTPFUNC()`
 - `TPFUNC`属性の設定有無をチェックします。
- `public boolean isTPGETANY()`
 - `TPGETANY`属性の設定有無をチェックします。
- `public boolean isTPNOBLOCK()`
 - `TPNOBLOCK`属性の設定有無をチェックします。
- `public boolean isTPNOCHANGE()`
 - `TPNOCHANGE`属性の設定有無をチェックします。
- `public boolean isTPNOREPLY()`

- TPNOREPLY属性の設定有無をチェックします。
- public boolean isTPNOTIME()
 - TPNOTIME属性の設定有無をチェックします。
- public boolean isTPNOTRAN()
 - TPNOTRAN属性の設定有無をチェックします。
- public boolean isTPRQS()
 - TPRQS属性の設定有無をチェックします。
- public void setTPBLOCK(boolean value)
 - TPBLOCK属性を設定または解除します。
 - パラメータ

パラメータ	説明
value	TRUEの場合はTPBLOCK属性を設定し、FALSEの場合はTPBLOCK属性の設定を解除します

- public void setTPFUNC(boolean value)
 - TPFUNC属性を設定または解除します。
 - パラメータ

パラメータ	説明
value	TRUEの場合はTPFUNC属性を設定し、FALSEの場合はTPFUNC属性の設定を解除します

- public void setTPGETANY(boolean value)
 - TPGETANY属性を設定または解除します。
 - パラメータ

パラメータ	説明
value	TRUEの場合はTPGETANY属性を設定し、FALSEの場合はTPGETANY属性の設定を解除します

- public void setTPNOBLOCK(boolean value)

– TPNOBLOCK属性を設定または解除します。

– パラメータ

パラメータ	説明
value	TRUEの場合はTPNOBLOCK属性を設定し、FALSEの場合はTPNOBLOCK属性の設定を解除します

- public void setTPNOCHANGE(boolean value)

– TPNOCHANGE属性を設定または解除します。

– パラメータ

パラメータ	説明
value	TRUEの場合はTPNOCHANGE属性を設定し、FALSEの場合はTPNOCHANGE属性の設定を解除します

- public void setTPNOREPLY(boolean value)

– TPNOREPLY属性を設定または解除します。

– パラメータ

パラメータ	説明
value	TRUEの場合はTPNOREPLY属性を設定し、FALSEの場合はTPNOREPLY属性の設定を解除します

- public void setTPNOTIME(boolean value)

– TPNOTIME属性を設定または解除します。

– パラメータ

パラメータ	説明
value	TRUEの場合はTPNOTIME属性を設定し、FALSEの場合はTPNOTIME属性の設定を解除します

- public void setTPNOTRAN(boolean value)
 - TPNOTRAN属性を設定または解除します。
 - パラメータ

パラメータ	説明
value	TRUEの場合はTPNOTRAN属性を設定し、FALSEの場合はTPNOTRAN属性の設定を解除します

- public void setTPRETRY(boolean value)
 - TPRETRY属性を設定または解除します。
 - パラメータ

パラメータ	説明
value	TRUEの場合はTPRETRY属性を設定し、FALSEの場合はTPRETRY属性の設定を解除します

- public void setTPRQS(boolean value)
 - TPRQS属性を設定または解除します。
 - パラメータ

パラメータ	説明
value	TRUEの場合はTPRQS属性を設定し、FALSEの場合はTPRQS属性の設定を解除します

3.3.2. WebtBuffer

WebTとTmaxサーバー間で送受信するユーザー・データを管理するクラスです。WebtRemoteServiceのTPメソッド(tpcall、tpgetreply)の戻り値とWebtEventHandlerのhandleEventのパラメータに、このクラスが使用されます。tpcall、tpacallメソッドを呼び出す際に必要なユーザ・データはWebtBufferに格納されます。

WebtBufferクラスは抽象クラスのため、ユーザーがオブジェクトを直接作成できません。tpcall、tpacallメソッドの入力データに使用するWebtBufferを取得するためには、WebtRemoteService.createStringBuffer、WebtRemoteService.createCarrayBuffer、WebtRemoteService.createFieldBufferメソッドなどを使用します。

構築子

WebtBufferオブジェクトを作成します。以下は、WebtBufferクラスの構築子についての説明です。

- プロトタイプ

```
public WebtBuffer()
```

フィールド

以下は、WebtBufferクラスのフィールド一覧です。

- static int BT_CARRAY
 - CARRAYバッファ型です。getBufferType()メソッドの戻り値がBT_CARRAYの場合に使用されます。
- static int BT_FIELD
 - FIELDバッファ型です。getBufferTypeメソッドの戻り値がBT_FIELDの場合に使用されます。
- static int BT_STRING
 - STRINGバッファ型です。getBufferTypeメソッドの戻り値がBT_STRINGの場合に使用されます。

参考

バッファ型に関する詳細については、Tmaxガイドの『Tmax 運用ガイド』を参照してください。

メソッド

以下は、WebtBufferクラスのメソッド一覧です。

- public static WebtBuffer createCarrayBuffer()
 - バッファ型がBT_CARRAYのWebtBufferオブジェクトを作成します。
 - 文字セットはWebtSystem.getDefaultCharsetメソッドの戻り値を使用します。

- `public static WebtBuffer createCarrayBuffer(String charset)`
 - バッファー型がBT_CARRAYのWebtBufferオブジェクトを作成します。
 - 文字セットはWebtSystem.getDefaultCharsetメソッドの戻り値を使用します。

- `public static WebtBuffer createFieldBuffer()`
 - バッファー型がBT_FIELDのWebtBufferオブジェクトを作成します。
 - 文字セットはWebtSystem.getDefaultCharsetメソッドの戻り値を使用します。

- `public static WebtBuffer createFieldBuffer(String charset)`
 - バッファー型がBT_FIELDのWebtBufferオブジェクトを作成します。
 - 文字セットはWebtSystem.getDefaultCharsetメソッドの戻り値を使用します。

- `public static WebtBuffer createStringBuffer()`
 - バッファー型がBT_STRINGのWebtBufferオブジェクトを作成します。
 - 文字セットはWebtSystem.getDefaultCharsetメソッドの戻り値を使用します。

- `public static WebtBuffer createStringBuffer(String charset)`
 - バッファー型がBT_STRINGのWebtBufferオブジェクトを作成します。
 - 文字セットはWebtSystem.getDefaultCharsetメソッドの戻り値を使用します。

- `public String getBufferTypeString()`
 - バッファー型に該当する文字列を返します。

- `public int getBufferType()`
 - バッファー型を返します。

- `public int getCallDescriptor()`
 - WebtBufferのcall descriptorを返します。call descriptorはWebtRemoteService.tpacall関数が返す値で、tpgetrplyメソッドのパラメータに使用されます。有効な値は0以上です。

- `public int getDataLength()`
 - `byte stream`に変換されたデータの長さを返します。

- `public String getDefaultCharset()`
 - バッファで使用するデフォルトの文字セットを返します。戻り値がNULLの場合、システムのデフォルトの文字セットを使用し、文字セットの変換を行います。

- `public String getServiceName()`
 - 現在設定されているサービス名を返します。

- `public String getString()`
 - バッファ型がBT_STRING、BT_CARRAYの場合、データの開始部分から0x00までの内容を文字列に変換して返します。0x00がバイト配列にない場合、データ全体を文字列に変換して返します。文字列に変換する際、エンコードはgetDefaultCharsetメソッドが返した文字セットを使用します。

– 例外

例外	説明
WebtBufferException	getBufferTypeメソッドの戻り値がBT_STRING、BT_CARRAYでない場合や、getDefaultCharsetメソッドの戻り値がシステムでサポートしていない文字セットの場合に発生します

- `public int getUserReturnCode()`
 - Tmaxサービス・ルーチンで `tpreturn`メソッドを実行する際、`urcode`に設定した値を返します。

- `public void setDefaultCharset(String charset)`
 - バッファで使用する文字セットを指定します。バッファを始める前に指定しなければ文字セットが適用されません。指定していない場合、WebtRemoteService.getDefaultCharsetメソッドの戻り値が適用されます。

- `public void setString(String val)`
 - Tmaxにサービスを要求するデータを設定します。

 - バッファ型がBT_STRING、BT_CARRAYの場合のみ使用できます。そうでない場合、WebtBufferException (`tperrno = TPEINVAL`)が発生します。バイト配列に変換されたデータの終わり

には0x00が追加されます。文字セットの変換にはgetDefaultCharacterSetメソッドが返した値を使用します。

– パラメータ

パラメータ	説明
val	バッファーに設定する内容です

– 例外

例外	説明
WebtBufferException	getBufferTypeメソッドの戻り値がBT_STRINGやBT_CARRAYでない場合や、getDefaultCharacterSetメソッドの戻り値がシステムでサポートしていない文字セットの場合に発生します

例

以下は、WebtBufferを使用してTmaxサービスが提供される例です。バッファー型はBT_STRINGです。

```
WebtConnection connection = null;

try {
    connection = new WebtConnection("localhost", 8888);

    WebtRemoteService service = new WebtRemoteService("TEST", connection);
    WebtBuffer request = service.createStringBuffer();
    request.setString("test string");

    WebtAttribute attr = new WebtAttribute();
    attr.setTPNOTIME();
    WebtBuffer response = service.tpcall(request, attr);
    System.out.println("response : " + response.getString());
}

catch (WebtIOException ioe) {
    System.out.println("webt connection closed");
    ioe.printStackTrace();
}

catch (WebtServiceFailException sfe) {
    System.out.println("server returns TP_FAIL");
}
}
```

```
catch (WebtServiceException se) {
    System.out.println("service fail. tperrno = " + se.getTPErrorMessage());
    se.printStackTrace();
}

finally {
    connection.close();
}
```

3.3.3. WebtCarrayBuffer/WebtStringBuffer

WebtBufferを継承したクラスで、CARRAY型、STRING型のデータを持つように宣言されたオブジェクトです。

構築子

WebtCarrayBufferオブジェクトとWebtStringBufferオブジェクトを作成します。以下は、WebtCarrayBuffer/WebtStringBuffer構築子についての説明です。

- WebtCarrayBuffer

- プロトタイプ

```
WebtCarrayBuffer()
```

- WebtStringBuffer

- プロトタイプ

```
WebtStringBuffer()
```

メソッド

以下は、WebtCarrayBufferクラスとWebtStringBufferクラスのメソッド一覧です。

- public void clear()

- 内部のすべてのWebtFieldオブジェクトを削除します。

- public byte[] getBytes()

- バッファ型がBT_CARRAY、BT_STRINGの場合、データ開始部分から終了部分までの内容を返します。getBytes(false)と同じです。

- 例外

例外	説明
WebtBufferException	バッファ型がBT_CARRAYまたはBT_STRINGでない場合に発生します

● public byte[] getBytes(boolean clone)

- バッファ型がBT_CARRAYの場合、データ開始部分から終了部分までの内容を返します。

- パラメータ

パラメータ	説明
clone	TRUEの場合、内容をコピーした新しいバイト配列を返します

- 例外

例外	説明
WebtBufferException	バッファ型がBT_CARRAYでない場合に発生します

● public byte[] getBytes(int offset, int length)

- バッファ型がBT_STRING、BT_CARRAYの場合、パラメータoffsetに設定された開始点からパラメータlengthに設定された長さ分の内容をバイト配列に返します。

- パラメータ

パラメータ	説明
offset	バッファから取得する開始点です
length	読み取るデータの長さです

- 例外

例外	説明
WebtBufferException	バッファ型がBT_STRING、BT_CARRAYではない場合や文字セットをシステムでサポートしない場合に発生します

● public String getString(int offset, int length)

- バッファ型がBT_STRING、BT_CARRAYの場合、パラメータoffsetに設定された開始点からパラメータlengthに設定された長さ分の内容をString型で返します。

– パラメータ

パラメータ	説明
offset	バッファから取得する開始点です
length	読み取るデータの長さです

– 例外

例外	説明
WebtBufferException	バッファ型がBT_STRING、BT_CARRAYでない場合や、文字セットをシステムでサポートしない場合に発生します

● public String getString(String charset)

- バッファ型がBT_STRING、BT_CARRAYの場合、データ開始部分から0x00までの内容をString型に変換して返します。0x00がバイト配列にない場合、データ全体をString型に変換して返します。

–

パラメータ	説明
charset	バイト配列をString型に変換する際に使用する文字セットです。 WebtBuffer.getDefaultCharsetメソッドが返した値を使用します。

– 例外

例外	説明
WebtBufferException	バッファ型がBT_STRING、BT_CARRAYでない場合や、文字セットをシステムでサポートしない場合に発生します

● public void setBytes(byte[] val)

- Tmaxにサービスを要求するデータを設定します。
- バッファ型がBT_CARRAYの場合のみ使用できます。そうでない場合、WebtBufferException(tperrno = TPEINVAL)が発生します。
- setBytes(val, false)メソッドを実行した場合と戻り値が同じです。

– パラメータ

パラメータ	説明
val	バッファに設定する内容です

– 例外

例外	説明
WebtBufferException	バッファ型がBT_CARRAYでない場合に発生します

● public void setBytes(byte[] val, boolean clone)

– Tmaxにサービスを要求するデータを設定します。

– バッファ型がBT_CARRAYの場合のみ使用できます。そうでない場合、WebtBufferException (tperrno = TPEINVAL)が発生します。

– パラメータ

パラメータ	説明
clone	内容をコピーするかどうかを設定します。TRUEの場合、valと同じバイト配列を内部に作成して内容をコピーします

– 例外

例外	説明
WebtBufferException	バッファ型がBT_CARRAYでない場合に発生します

● public int setBytes(byte[] value, int off, int len)

– valueの内容をoffからコピーして保存し、入力されたデータの長さを返します。

– バッファ型がBT_STRING、BT_CARRAYの場合に使用できます。そうでない場合、WebtBufferException (tperrno = TPEINVAL)が発生します。

– パラメータ

パラメータ	説明
value	保存するデータです
off	読み取る開始点です
len	コピーするデータの長さです

– 例外

例外	説明
WebtBufferException	バッファ型がBT_STRING、BT_CARRAYでない場合や、文字セットをシステムでサポートしない場合に発生します

- public int setString(int offset, String value)

- valueの内容をoffsetからコピーして保存し、入力されたデータの長さを返します。
- バッファ型がBT_STRING、BT_CARRAYの場合に使用できます。そうでない場合、WebtBufferException (tperrno = TPEINVAL)が発生します。
- パラメータ

パラメータ	説明
offset	読み取る開始点です
value	保存するデータです

- 例外

例外	説明
WebtBufferException	バッファ型がBT_STRING、BT_CARRAYでない場合や、文字セットをシステムでサポートしない場合に発生します

- public void setString(String value, String charset)

- ユーザーデータを設定します。
- バッファ型がBT_STRING、BT_CARRAYの場合のみ使用できます。そうでない場合、WebtBufferException(tperrno = TPEINVAL)が発生します。バイト配列に変換されたデータの最後に0x00が追加されます。
- パラメータ

パラメータ	説明
value	バッファに設定する内容です
charset	valueをバイト配列に変換する際に使用する文字セットです。NULLの場合、getDefaultCharacterSetメソッドが返す値を使用します

- 例外

例外	説明
WebtBufferException	バッファー型がBT_STRING、BT_CARRAYでない場合や、文字セットをシステムでサポートしない場合に発生します

3.4. サービス呼び出し関連クラス

3.4.1. WebtRemoteService

WebtRemoteServiceクラスは、WebtConnectionクラスおよびその subclasses を通じてTmaxサービスの提供を受けられるようにするAPIを提供し、Tmaxサービス呼び出す際に必要な属性およびサービス名などを管理するクラスです。WebtRemoteServiceクラスの使用方法については「[3.3.2. WebtBuffer](#)」を参照してください。

構築子

WebtRemoteServiceオブジェクトを作成します。以下は、WebtRemoteServiceクラスの構築子についての説明です。

- プロトタイプ

```
WebtRemoteService(String svcname, WebtConnection con)
```

- パラメータ

パラメータ	説明
svcname	Tmaxに要求するサービス名です
con	Tmax接続情報と属性が設定されているWebtConnectionオブジェクトです

メソッド

以下は、WebtRemoteServiceクラスのメソッド一覧です。

- public void clear()
 - デフォルトのサービス名(default service name)とWebtConnectionオブジェクトをNULLに初期化します。デフォルトの文字セット(default character set)をWebtSystem.getDefaultCharsetメソッドの戻り値に初期化します。
- public WebtBuffer createCarrayBuffer([int size])

- バッファ型がBT_CARRAYのWebtBufferオブジェクトを作成します。
- public WebtFieldBuffer createFieldBuffer([int size|WebtFDLKeyTable table])
 - バッファ型がBT_FIELDのWebtBufferオブジェクトを作成します。
 - パラメータ

パラメータ	説明
table	FDL key tableインスタンスです

- public WebtBuffer createStringBuffer([int size])
 - バッファ型がBT_STRINGのWebtBufferオブジェクトを作成します。
- public WebtConnection getConnection()
 - TPメソッド(tpcall、tpgetreply)の呼び出しに使用するWebtConnectionオブジェクトを返します。
- public String getServiceName()
 - デフォルトのサービス名を返します。
- public void setDefaultCharset(String charset)
 - WebtRemoteServiceオブジェクトによって作成されるすべてのWebtBufferオブジェクトに提供される文字セットを設定します。
 - createStringBuffer、createCarrayBuffer、createFieldBufferメソッドを呼び出す前に設定しなければ、作成されたWebtBufferオブジェクトに設定された値が適用されません。
 - パラメータ

パラメータ	説明
charset	適用する文字セットです

- public String getDefaultCharset()
 - WebtRemoteServiceオブジェクトによって作成されるすべてのWebtBufferオブジェクトに適用される文字セットを返します。

- `public void setServiceName(String svcname)`
 - デフォルトのサービス名を設定します。
- `public int tpcall([String svcname,] WebtBuffer tx [,WebtAttribute attr] [,WebtMessageHandler])`
 - Tmaxサーバーにtpcallを呼び出します。

– パラメータ

パラメータ	説明
svcname	サービスの名前です
tx	サービス要求を送信するバッファです
attr	メソッドを呼び出す際に適用する属性です

– 例外

例外	説明
WebtServiceFailException	TmaxサービスがTPFAILを返した際に発生します
WebtTXException	トランザクション・エラーが生じた際に発生します
WebtServiceException	Tmaxサービス・エラーが生じた際に発生します
WebtIOException	ネットワークに関連するTmaxサービス・エラーが生じた際に発生します

- `public WebtBuffer tpcall([String svcname,] WebtBuffer tx [,WebtAttribute attr])`
 - Tmaxサーバーにtpcallメソッドを呼び出します。

– パラメータ

パラメータ	説明
svcname	サービスの名前です
tx	サービス要求を送信するバッファです
attr	メソッドを呼び出す際に適用する属性です

– 例外

例外	説明
WebtServiceFailException	TmaxサービスがTPFAILを返した際に発生します
WebtTXException	トランザクション・エラーが生じた際に発生します

例外	説明
WebServiceException	Tmaxサービス・エラーが生じた際に発生します
WebtIOException	ネットワークに関連するTmaxサービス・エラーが生じた際に発生します

- `public int tpcallsvg(int svgno|String nodename, [String svcName,] WebtBuffer txBuffer [,WebtAttribute attribute])`

– 特定サーバー・グループに属するサービスを呼び出すためのメソッドです。COUSINでまとめたマルチサーバー・グループ環境で、特定サーバー・グループに属しているサービスを指定し、非同期通信でサービス要求を送信します。特定サーバー・グループを指定してサービスを呼び出すこと以外はtpacall ()と同じように動作します。

– パラメータ

パラメータ	説明
svgno	サービスが属しているサーバー・グループの番号です。サーバー・グループの番号はtpgetsvglistメソッドを通じて知ることができます。-1と設定した場合、tpcallメソッドと同じように動作します
nodeName	ノードの名前です
svcName	サービスの名前です
txBuffer	サービス要求を送信するバッファです
attribute	メソッドを呼び出す際に適用する属性です

- `public WebtBuffer tpcallsvg(int svgno|String nodename, [String svcName,] WebtBuffer txBuffer [,WebtAttribute attribute])`

– 特定サーバー・グループに属するサービスを呼び出すためのメソッドです。COUSINでまとめたマルチサーバー・グループ環境で、特定サーバー・グループに属しているサービスを指定し、非同期通信でサービス要求を送信します。特定サーバー・グループを指定してサービスを呼び出すこと以外はtpacall()と同じように動作します。

起動しないサーバー・プロセス(spr)で呼び出す場合には、tpgetreplyメソッドで例外がTPENOENTを返し、ヘッダータイプを「extendedV4」に設定する必要があります。(旧バージョンのヘッダーはサポートしません)

– パラメータ

パラメータ	説明
svgno	サービスが属しているサーバー・グループの番号です。サーバー・グループの番号はtpgetsvglist() APIを通じて知ることができます。-1と設定した場合、既存

パラメータ	説明
	のtpcallメソッドと同じように動作します。サービス名はWebtRemoteServiceオブジェクトを作成する際に決定した名前を使用します
nodeName	ノードの名前です
svcName	サービスの名前です
txBuffer	サービス要求を送信するバッファです
attribute	メソッドを呼び出す際に適用する属性です

- public int tpgetsprlist([String svcName], int svgno, int[] index, WebtAttribute attribute)
 - 特定サーバー・グループに属するサービスのサーバー・プロセス・インデックスを取得するためのメソッドです。サーバー・プロセス単位で呼び出すために使用されます。

– パラメータ

パラメータ	説明
svcName	サービスの名前です。省略された場合はWebtRemoteServiceに指定されているサービス名を使用します
svgno	サービスが属しているサーバー・グループの番号です。サーバー・グループの番号はtpgetsvglistメソッドを通じて知ることができます
index	サーバー・プロセス・インデックスの最小値と最大値を取得するためのint型配列の領域で、少なくとも2つの領域が割り当てられる必要があります。メソッドが正常に実行された場合、index[0]には開始インデックス値、index[1]には終了インデックス値が設定されます
attribute	メソッドを呼び出す際に適用する属性です

- public int tpspracall([String svcName], int spr, WebtBuffer txBuffer, WebtAttribute attribute)
 - 特定サーバー・プロセスのサービスを呼び出すコマンドです。tpgetsprlistメソッドを通じてサーバー・プロセス番号の範囲を取得する必要があります。それ以外には、戻り値を使用してtpgetreplyメソッドで応答を受け取ることができる点はtpacallと同じように使用されます。

起動しないサーバー・プロセス(spr)で呼び出す場合には、tpgetreplyメソッドで例外がTPENOENTを返し、ヘッダータイプを「extendedV4」に設定する必要があります。(旧バージョンのヘッダーはサポートしません)

– パラメータ

パラメータ	説明
svcName	サービスの名前です。省略した場合、WebtRemoteServiceに指定されているサービス名を使用します
spri	サーバー・プロセス・インデックスです。tpgetsprlistメソッドを通じて知ることができます
txBuffer	サービス要求を送信するバッファです
attribute	メソッドを呼び出す際に適用する属性です。現在のバージョンでは属性の設定を無視して動作します

- public void tpcancel(int cd)

- tpcallメソッドが返した呼び出し記述子のcdをキャンセルします。グローバル・トランザクションに関連したサービスはキャンセルできません。サービスの応答が正常にキャンセルされた場合、cdを通じて受け取った応答も無視されます。WebtTPAListnerを使用してtpcallメソッドを呼び出された場合はerrorHandleメソッドを呼び出します。

- パラメータ

パラメータ	説明
cd	call descriptorの略字で、tpcall()の戻り値です

- 例外

例外	説明
WebtServiceException	WebtException.TPEBADDESCdlekエラーが生じた場合に発生します。トランザクション修理中の場合はエラー番号がWebtException.TPETRAN、cd値が存在しない値の場合はエラー番号がWebtException.TPEBADDESCです

- public WebtBuffer tpgetrply(int cd [,WebtAttribute attr])

- tpcall()の応答を受け取るメソッドで、int tpgetrply(int *cd, char **rbuf, long *rlen, long flags)と同じ機能を実行します。

- パラメータ

パラメータ	説明
cd	call descriptorの略字で、tpcall()の戻り値です
attr	メソッドを呼び出す際に適用する属性です。attrに適用する属性は『Tmaxアプリケーション開発ガイド』を参照してください

– 例外

例外	説明
java.lang.IllegalStateException	WebtEventHandlerによってacall replyのコールバックが登録された状態でtpgetrply()を呼び出す際に発生します

- public int[] tpgetsvglist([[[String nodename,] String svcName,] WebtAttribute attribute])

– 該当サービスが属するサーバー・グループと、そのサーバー・グループのCOUSINIに設定されたサーバー・グループについての情報を提供するメソッドです。

– パラメータ

パラメータ	説明
nodeName	ノードの名前です
svcName	サービスの名前です
attribute	現在はサポートされておらず、TOPNOFLAGSを設定します

3.4.2. WebtRQService

WebtRemoteServiceを継承したクラスで、RQ(Reliable Queue)を管理するサービスを使用するためのクラスです。

構築子

Tmax RQサービスを受け取るためのWebtRQServiceオブジェクトを作成します。以下は、WebtRQServiceクラスの構築子についての説明です。

- プロトタイプ

```
public WebtRQService(String qname,  
                    String svcname,  
                    WebtConnection con)
```

- パラメータ

パラメータ	説明
qname	tpend、tpdeqメソッドを呼び出す際のデフォルト値で、使用するTmax RQの名前を指定します。Tmax環境設定ファイルに登録された名前ではなく、NULLにはできません

パラメータ	説明
svcname	penq、tpdeqメソッドを呼び出す際のデフォルト値で、使用するTmaxサービスの名前を指定します。NULLやTmax環境設定ファイルに登録された名前を指定します
con	サービスを受けるためのWebtConnectionオブジェクトです。NULLにはできません

- 例外

例外	説明
java.lang.IllegalArgumentException	conがNULLの場合に発生します

フィールド

以下は、WebtRQServiceクラスのフィールド一覧です。

- public static final String RQ_ANY_QUEUE
 - RQ_ANY_QUEUEを設定してRQサービスを使用した場合、WebtRQService.tpqstat、WebtRQService.tpqsvcstatメソッドを呼び出す際に、Request Queue、Reply Queue、Fail Queueの統計を受信できます。
- public static final String RQ_FAIL_QUEUE
 - RQ_FAIL_QUEUEを設定してRQサービスを使用した場合、WebtRQService.tpqstat、WebtRQService.tpqsvcstatメソッドを呼び出す際に、Fail Queueの統計値を受信できます。
- public static final String RQ_REQ_QUEUE
 - RQ_REQ_QUEUEを設定してRQサービスを使用した場合、WebtRQService.tpqstat、WebtRQService.tpqsvcstatメソッドを呼び出す際に、Request Queueの統計値を受信できます。
- public static final String RQ_RPLY_QUEUE
 - RQ_RPLY_QUEUEを設定してRQサービスを使用した場合、WebtRQService.tpqstat、WebtRQService.tpqsvcstatメソッドを呼び出す際に、Reply Queueの統計値を受信できます。
- public static final int TPERR
 - WebtRQService.tpenqメソッドを呼び出す際、サービスが失敗してFail Queueに送信された場合に受信される値です。

- `WebtRQService.checkError`メソッドを通じてリターン・タイプを確認でき、`WebtBuffer.getUserReturnCode`メソッドを通じてエラーコードを確認できます。
- `public static final int TPFAIL`
 - `WebtRQService.tpenq`メソッドを呼び出す際、サービス名を指定した場合、該当サービスで`treturn`の最初のパラメータに`TPFAIL`が設定された場合に受信される値です。
 - `WebtRQService.checkError`メソッドを通じてリターン・タイプを確認でき、`WebtBuffer.getUserReturnCode`メソッドを通じてエラーコードを確認できます。
- `public static final int TPREQ`
 - `WebtRQService.tpenq`メソッドを呼び出す際、2番目のパラメータにサービス名ではない`NULL`が指定された場合、サービスが正常に行われた場合に受信される値です。
 - `WebtRQService.checkError`メソッドを通じてリターン・タイプを確認でき、`WebtBuffer.getUserReturnCode`メソッドを通じてエラーコードを確認できます。
- `public static final int TPSUCCESS`
 - `WebtRQService.tpenq`メソッドを呼び出す際、サービス名を指定した場合、該当サービスで`treturn`の最初のパラメータに`TPSUCCESS`が設定された場合に受信される値です。
 - `WebtRQService.checkError`メソッドを通じてリターン・タイプを確認でき、`WebtBuffer.getUserReturnCode`メソッドを通じてエラーコードを確認できます。

メソッド

以下は、`WebtRQService`クラスのメソッド一覧です。

- `public static int checkError(WebtBuffer rqbuf)`
 - RQに保存されたデータの詳細情報を受け取るメソッドです。サービス名は`rqbuf.getServiceName`メソッドで知ることができ、エラーコードは`rqbuf.getUserReturnCode`メソッドで知ることができます。
- `public String getQueueName()`
 - 指定されたデフォルトのRQ名を返します。
- `public void setQueueName(String qname)`

- tpenq()、tpdeq()を呼び出す際のデフォルト値で、使用するRQ名を再指定します。

- パラメータ

パラメータ	説明
qname	新しく指定するデフォルトのRQ名です

- 例外

例外	説明
java.lang.IllegalArgumentException	qnameがNULLの場合に発生します

● public WebtBuffer tpdeq(String qname, String svcname, WebtAttribute attr)

- tpenq()サービスを呼び出すメソッドで、RQに保存されたデータを読み取る際に使用します。1度読み取ったデータはRQから削除されるため、1度受信したデータを再び読み取ることはできません。

- パラメータ

パラメータ	説明
qname	データを保存するRQの名前です。Tmax環境設定ファイルに登録された名前であればなりません
svcname	呼び出すサービス名です。Tmax環境設定ファイルに登録されたサービス名でなければなりません。Reply Queueで同じサービス名をもつデータを順次に受信します
attr	メソッドを呼び出す際に適用する属性です。設定可能な値については以下の表を参照してください

以下は、attrに設定可能な値についての説明です。

パラメータ	説明
WebtAttribute.TPRQS	Reply Queueでデータを受信する際に指定します
WebtAttribute.TPFUNC	Reply Queueに保存されたデータを受信する際にWebtAttribute.TPRQSと一緒に指定します。記録だけが行われ、サービスは呼び出しません
WebtAttribute.TPNOFLAGS	結果値をクライアントのバッファーに送信します。tpdeqでも結果値を受信するためにTPNOFLAGSを設定します

- 例外

例外	説明
WebtServiceException	qnameがNullの場合やQueueが空いている場合に発生します

例外	説明
WebServiceFailException	サービスが正しい応答をしなかった場合に発生します
WebtIOException	ネットワーク・エラーが生じた場合に発生します
WebtTXException	トランザクション・エラーが生じた場合に発生します

- public int tpenq(String qname, String svcname, WebtBuffer tx, WebtAttribute attr)

– tpenqサービスを呼び出すメソッドで、RQに保存されたデータを読み取る際に使用します。1度読み取ったデータはRQから削除されるため、再び読み取ることはできません。

– パラメータ

パラメータ	説明
qname	データを保存するRQの名前です。Tmax環境設定ファイルに登録された名前であればなりません
svcname	呼び出すサービス名です。Tmax環境設定ファイルに登録されたサービス名でなければなりません。Reply Queueで同じサービス名をもつデータを順次に受信します
tx	サービスを呼び出す際に送信するデータです。RQで受信するデータを受け取る変数を指定します
attr	メソッドを呼び出す際に適用する属性です。設定可能な値については以下の表を参照してください

以下は、attrに設定可能な値についての説明です。

パラメータ	説明
WebtAttribute.TPRQS	Reply Queueでデータを受信する際に指定します
WebtAttribute.TPFUNC	Reply Queueで保存されたデータを受信する際にWebtAttribute.TPRQSと一緒に指定します。記録だけが行われ、サービスは呼び出しません
WebtAttribute.TPNOFLAGS	結果値をクライアントのバッファーに送信します。tpdeqでも結果値を受信するためにTPNOFLAGSを設定します

– 例外

例外	説明
WebServiceException	qnameがNULLの場合やQueueが空いている場合に発生します
WebServiceFailException	サービスが正しい応答をしなかった場合に発生します
WebtIOException	ネットワーク・エラーが生じた場合に発生します
WebtTXException	トランザクション・エラーが発生した場合に発生します

- int tpenq(WebtBuffer tx [,WebtAttribute attr])

- tpenqサービスを呼び出すメソッドで、サービスを要求して受信したデータをRQに保存する関数です。保存されたデータは到着順に管理され、データの読み込みは取得順に処理されます。

- パラメータ

パラメータ	説明
svcname	呼び出すサービス名です。Tmax環境設定ファイルに登録された名前であればなりません
tx	サービスを呼び出す際に送信が必要なデータを指定します
attr	メソッドを呼び出す際に適用する属性です。設定可能な値については以下の表を参照してください

以下は、attrに設定可能な値についての説明です。

パラメータ	説明
WebtAttribute.TPRQS	Request Queueにデータが記録され、該当サービスを実行後、Reply Queueに結果値が記録されます
WebtAttribute.TPNOREPLY	該当サービスを実行し、結果をReply Queueに保存しません
WebtAttribute.TPFUNC	サービスを呼び出す際にReply Queueに該当データが記録され、サービスを呼び出しません
WebtAttribute.TPNOFLAGS	サービスを実行後、サービス結果とは関係なく該当結果をクライアント・バッファに保存します

- 例外

例外	説明
WebtServiceException	サービスが正しい応答をしなかった場合に発生します
WebtIOException	ネットワーク・エラーが生じた場合に発生します
WebtTXException	トランザクション・エラーが生じた場合に発生します

- public int tpqstat(String qname, String qtype)

- 現在キューに保存されているデータの統計値を求める関数で、qnameがNULLの場合、getQueueNameメソッドを使用します。

- パラメータ

パラメータ	説明
qname	統計値を受信するRQの名前です。Tmax環境設定ファイルに登録された名前である必要があります
qtype	qtypeに設定できる値については以下の表を参照してください

qtypeに設定できる値は以下のとおりです。

パラメータ	説明
RQ_ANY_QUEUE	Request Queue、Reply Queue、Fail Queueの統計を出します
RQ_FAIL_QUEUE	Fail Queueの統計を出します
RQ_REQ_QUEUE	Request Queueの統計を出します
RQ_RPLY_QUEUE	Reply Queueの統計を出します

– 例外

例外	説明
WebtServiceException	qnameがNULLの場合やqtypeが有効な名前ではない場合に発生します
WebtIOException	ネットワーク・エラーが生じた場合に発生します

● int tpqsvstat(String qname, String svcname, String qtype)

- 現在Queue1に保存されているデータの統計値をサービス名別に求める関数で、qnameがNULLの場合はgetQueueNameメソッドを使用します。

– パラメータ

パラメータ	説明
qname	統計値を受信するRQの名前を指定します。Tmax環境設定ファイルに登録された名前であればなりません
svcname	サービスの名前です
qtype	qtypeに設定できる値については以下の表を参照してください

qtypeに設定できる値は以下のとおりです。

パラメータ	説明
WebtRQService.RQ_ANY_QUEUE	Request Queue、Reply Queue、Fail Queueの統計を出します
WebtRQService.RQ_FAIL_QUEUE	Fail Queueの統計を出します
WebtRQService.RQ_REQ_QUEUE	Request Queueの統計を出します
WebtRQService.RQ_RPLY_QUEUE	Reply Queueの統計を出します

– 例外

例外	説明
WebServiceException	qnameがNULLの場合やqtypeが有効な名前ではない場合に発生します
WebtIOException	ネットワーク・エラーが生じた場合に発生します

3.4.3. WebtDialogueService

WebtRemotedServiceを継承したクラスで、会話型サービス(Conversation mode service)を提供します。

構築子

WebtDialogueServiceオブジェクトを作成します。以下は、WebtDialogueServiceクラスの構築子についての説明です。

- プロトタイプ

```
public WebtDialogueService(String svcname,  
                           WebtConnection con)
```

- パラメータ

パラメータ	説明
svcname	Tmaxに要求するサービス名です
con	Tmax接続情報と属性が設定されているWebtConnectionオブジェクトです

- 例外

例外	説明
java.lang.IllegalArgumentException	svcnameまたはconがNULLの場合に発生します

メソッド

以下は、WebtDialogueServiceクラスのメソッド一覧です。

- public boolean isClosed()
 - 現在、会話型サービスでセッションが終了しているかどうかを返します。
- public boolean isReceiveNext()

– 現在tprecvが可能かどうかを返します。

– 例外

例外	説明
WebServiceException	tpconnectができない状態でisReceiveNextメソッドを呼び出す際に発生します

● public boolean isSendNext()

– 現在tpsendが可能かどうかを返します。

– 例外

例外	説明
WebServiceException	tpconnectができない状態でisSendNextメソッドを呼び出す際に発生します

● DialogueQueue tpconnect([WebtBuffer sndbuf,] [WebtAttribute attr,] boolean recvNext)

– 会話型モードサービスの接続を設定します。接続を設定する際、必ず会話の制御権を設定します。

– パラメータ

パラメータ	説明
sndbuf	tpconnectの時点からサーバーに送るメッセージです
attr	送信バッファに適用する属性です。WebtAttributeに定義された属性のみ有効です
recvNext	– FALSE: tpconnect後に接続の制御権を引き続き維持し、tpsendができます。相手側は受信のみできます – TRUE: tpconnect後に接続の制御権を相手側に渡し、受信のみできます

– 例外

例外	説明
WebServiceException	正しい応答が返ってこなかった場合に発生します
WebtIOException	ネットワーク・エラーが生じた場合に発生します
WebtTXException	トランザクション・エラーが生じた場合に発生します

● public void tpdison()

- tpconnect後、接続の制御権を持っている側から会話型サービスの接続を即時終了できるようにします。

- 例外

例外	説明
WebtIOException	ネットワーク・エラーが生じた場合に発生します
WebtServiceException	正しい応答が返ってこなかった場合に発生します

● WebtBuffer tprecv([WebtAttribute attr])

- 接続の制御権を相手側に渡し、受信モード状態でメッセージを受信します。

- パラメータ

パラメータ	説明
attr	受信バッファに適用する属性です。WebtAttributeに定義された属性のみ有効です

- 例外

例外	説明
WebtIOException	ネットワーク・エラーが生じた場合に発生します
WebtServiceException	Queueが設定されていない場合や、タイムアウト時間内に正しい応答が返ってこなかった場合に発生します
WebtDialogueException	会話型サービスの通信状態が受信可能ではない場合に発生します

● void tpsend(WebtBuffer tx, [WebtAttribute attr,] boolean rcvNext)

- 会話型サービスでメッセージを送信します。

- パラメータ

パラメータ	説明
tx	送信するメッセージ・バッファです
attr	設定するユーザー属性です。WebtAttributeに定義された属性のみ有効です
rcvNext	送信後、送信モードに変更するかどうかを設定します - TRUE: 送信モードに変更します

– 例外

例外	説明
WebtIOException	ネットワーク・エラーが生じた場合に発生します
WebtServiceException	Queueが設定されていない場合や、タイムアウト時間内に正しい応答が返ってこなかった場合に発生します
WebtDialogueException	会話型サービスの通信状態が送信可能ではない場合に発生します

3.5. フィールド・データ関連クラス

3.5.1. WebtFieldElement

WebtFieldElementは、WebtFieldのフィールド・キーに対するフィールド・データを格納しているインターフェースです。WebtFieldはフィールド・データをWebtFieldElementクラスで表現して管理します。1つのフィールド・データに1つのWebtFieldElementオブジェクトが存在します。

WebtFieldElementクラスはインターフェースであるためユーザーが直接オブジェクトを作成できません。Tmaxサーバーから受信したフィールド・データの場合、WebTがWebtFieldElementオブジェクトを作成して該当WebtFieldオブジェクトに追加します。一方、ユーザーがWebtFieldElementオブジェクトを作成してWebtFieldに追加するには、WebtField.addメソッドとWebtField.insertメソッドを使用します。

メソッド

以下は、WebtFieldElementクラスのメソッド一覧です。

- byte[] bytesValue()

- フィールド・データの要素(Element)をbyte[]で表現して返します。

- フィールド・タイプに従って、以下の方法でフィールド・データを変換して返します。

- FB_CHAR, FB_CARRAY, FB_STRING

保存されているオリジナル・データをbyte[]で表現して返します。ただし、FB_STRINGの場合、Tmaxサーバーからデータを受信する際にデータの最後が'\0'で終わりますが、その'\0'を除いた状態で返します。

- FB_SHORT, FB_INT, FB_LONG, FB_FLOAT, FB_DOUBLE

stringValue().getBytes()を返します。

– 例外

例外	説明
WebtBufferException	フィールド・データがNULLの場合に発生します

● byte byteValue()

– フィールド・データの要素をbyte型に変換して返します。

– フィールド・タイプに従って、以下の方法でフィールド・データを変換して返します。

- FB_CHAR, FB_CARRAY, FB_STRING

保存されているオリジナル・データをbyte[]で表現して返します。ただし、FB_STRINGの場合、Tmaxサーバーからデータを受信する際にデータの最後が'\0'で終わりますが、その'\0'を除いた状態で返します。

- FB_SHORT, FB_INT, FB_LONG, FB_FLOAT, FB_DOUBLE

stringValue().getBytes()を返します。

– 例外

例外	説明
WebtBufferException	フィールド・データがNULLの場合に発生します

● double doubleValue()

– フィールド・データの要素をdouble型に変換して返します。

– オリジナル・データをvalueとした場合、フィールド・タイプに従って以下の方法でフィールド・データを変換して返します。

– フィールド・タイプに従って、以下の方法でフィールドデータを変換して返します。

- FB_CHAR, FB_CARRAY, FB_STRING

Double.valueOf(stringValue()).doubleValue()と同じ結果を返します。

- FB_SHORT, FB_INT, FB_LONG, FB_FLOAT, FB_DOUBLE

(new Double(value)).doubleValue()と同じ結果を返します。

– 例外

例外	説明
WebtBufferException	WebtFieldElementのフィールド・タイプがFB_STRING、FB_CHAR、FB_CARRAYの場合、数字で表現する際はjava.io.NumberFormatExceptionまたはjava.io.UnsupportedEncodingExceptionが発生することがあります。この場合、WebtBufferExceptionが発生します。または、フィールド・データがNULLの場合にWebtBufferExceptionが発生します

- float floatValue()

- データの要素をfloat型に変換して返します。
- オリジナル・データをvalueとした場合、フィールド・タイプに従って以下の方法でフィールド・データを変換して返します。
 - FB_CHAR, FB_CARRAY, FB_STRING
Float.valueOf(stringValue()).floatValue()と同じ結果を返します。
 - FB_SHORT, FB_INT, FB_LONG, FB_FLOAT, FB_DOUBLE
(new Double(value)).floatValue()と同じ結果を返します。

- 例外

例外	説明
WebtBufferException	WebtFieldElementのフィールド・タイプがFB_STRING、FB_CHAR、FB_CARRAYの場合、数字で表現する際にjava.io.NumberFormatExceptionまたはjava.io.UnsupportedEncodingExceptionが発生することがあります。その場合はWebtBufferExceptionが発生します。または、フィールド・データがNULLの場合にWebtBufferExceptionが発生します

- WebtField getField()

- WebtFieldElementが属するWebtFieldオブジェクトのリファレンスを返します。

- int intValue()

- フィールド・データの要素をint型に変換して返します。
- オリジナル・データをvalueとした場合、フィールド・タイプに従って以下の方法でフィールド・データを変換して返します。

- FB_CHAR, FB_CARRAY, FB_STRING
Integer.valueOf(stringValue()).intValue()と同じ結果を返します。
- FB_SHORT, FB_INT, FB_LONG, FB_FLOAT, FB_DOUBLE
(new Double(value)).intValue()と同じ結果を返します。

– 例外

例外	説明
WebtBufferException	WebtFieldElementのフィールド・タイプがFB_STRING、FB_CHAR、FB_CARRAYの場合、数字で表現する際にjava.io.NumberFormatExceptionまたはjava.io.UnsupportedEncodingExceptionが発生することがあります。その場合はWebtBufferExceptionが発生します。または、フィールド・データがNULLの場合にWebtBufferExceptionが発生します

- int longValue()

- フィールド・データの要素をlong型に変換して返します。
- オリジナル・データをvalueとした場合、フィールド・タイプに従って以下の方法でフィールド・データを変換して返します。

- FB_CHAR, FB_CARRAY, FB_STRING
Integer.valueOf(stringValue()).intValue()と同じ結果を返します。
- FB_SHORT, FB_INT, FB_LONG, FB_FLOAT, FB_DOUBLE
(new Double(value)).intValue()と同じ結果を返します。

– 例外

例外	説明
WebtBufferException	WebtFieldElementのフィールド・タイプがFB_STRING、FB_CHAR、FB_CARRAYの場合、数字で表現する際にjava.io.NumberFormatExceptionまたはjava.io.UnsupportedEncodingExceptionが発生することがあります。その場合はWebtBufferExceptionが発生します。または、フィールド・データがNULLの場合にWebtBufferExceptionが発生します

- int shortValue()

- フィールド・データの要素をshort型に変換して返します。
- オリジナル・データをvalueとした場合、フィールド・タイプに従って以下の方法でフィールド・データを変換して返します。
 - FB_CHAR, FB_CARRAY, FB_STRING
Integer.valueOf(stringValue()).shortValue()と同じ結果を返します。
 - FB_SHORT, FB_INT, FB_LONG, FB_FLOAT, FB_DOUBLE
(new Double(value)).shortValue()と同じ結果を返します。
- 例外

例外	説明
WebtBufferException	WebtFieldElementのフィールド・タイプがFB_STRING、FB_CHAR、FB_CARRAYの場合、数字で表現する際にjava.io.NumberFormatExceptionまたはjava.io.UnsupportedEncodingExceptionが発生することがあります。その場合はWebtBufferExceptionが発生します。または、フィールド・データがNULLの場合にWebtBufferExceptionが発生します

- int sstringValue([String charset])

- フィールド・データの要素をstring型に変換して返します。stringValue(getField().getDefaultCharset())と同じ結果です。
- 例外

例外	説明
WebtBufferException	Tmaxサービスを受信するための送受信バッファ(WebtBuffer)を使用中にエラーが生じた場合に発生します

3.5.2. WebtFDLKeyTable

Tmaxサービスを呼び出す際に使用するバッファ型がWebtBuffer.BT_FIELDの場合、フィールド・キーを使用します。一般のTmaxクライアントと同様に、WebTのフィールドキーの使用方法には以下の2つがあります。

区分	説明
int	フィールド定義ファイルに定義されているフィールドキーの名前をint型で表現した定数を使用する方式です
String	フィールド定義ファイルのフィールドキーの名前をString型で表現して使用する方式です。 フィールド定義ファイル(拡張子: .f)をfdlcユーティリティを使用してコンパイルした結果ファイルのfdlファイル(拡張子: .fdl)を、WebtFDLKeyTableクラスでロードする必要があります。fdlファイルをロードしてWebTで使用できるようにするには、WebtSystem.createDefaultFieldKeyTable(String)メソッドを使用します

構築子

WebtFDLKeyTableオブジェクトを作成します。以下は、WebtFDLKeyTableクラスの構築子についての説明です。

- プロトタイプ

```
WebtFDLKeyTable()
```

メソッド

以下は、WebtFDLKeyTableクラスのメソッド一覧です。

- public int getKey(String keyString)
 - フィールドキーの名前をint型の定数で表現した値を返します。
 - パラメータ

パラメータ	説明
keyString	フィールドキーの名前です

- public String getKeyName(int fldkey)
 - int型の定数で表現されているフィールドキーの名前をString型で表現した値を返します。
 - パラメータ

パラメータ	説明
fldkey	int型の定数です

- public int getType(String keyString)

- フィールドキーのタイプを返します。

- 有効でないフィールドキーの場合は-1を返し、有効なフィールドキーの場合は以下のうち1つを返します。

(FieldAccess.FB_CHAR , FieldAccess.FB_SHORT, FieldAccess.FB_INT, FieldAccess.FB_LONG,
FieldAccess.FB_FLOAT, FieldAccess.FB_DOUBLE, FieldAccess.FB_STRING,
FieldAccess.FB_CARRAY)

- パラメータ

パラメータ	説明
keyString	フィールドキーの名前です

- public void loadFDLKeys(String filepath)

- fdlファイルからフィールドキーをロードします。

- パラメータ

パラメータ	説明
filepath	fdlファイルのパスです

- 例外

例外	説明
java.io.IOException	ファイルの読み込みエラーが生じた場合に発生します

- public static int makeFDLKey(int type, int no)

- アプリケーション・プログラムでユーザーがフィールド・タイプとフィールド番号(field number)を定義した場合、int型の定数で表現したフィールドキーを作成して返します。

- パラメータ

パラメータ	説明
type	フィールド・タイプです。以下のうち1つを設定します。 (FieldAccess.FB_CHAR , FieldAccess.FB_SHORT, FieldAccess.FB_INT, FieldAccess.FB_LONG, FieldAccess.FB_FLOAT, FieldAccess.FB_DOUBLE, FieldAccess.FB_STRING, FieldAccess.FB_CARRAY)

パラメータ	説明
no	フィールド番号です

- public int size()
 - 現在ロードされているフィールドキーの数を返します。

3.5.3. WebtFieldSet

WebtFieldSetクラスは、Tmaxサービスに使用するバッファ型がWebtBuffer.BT_FIELDの場合、より簡単にフィールド・データの要素を操作できるようにするユーティリティー・クラスです。

構築子

与えられたWebtBufferでWebtFieldSetオブジェクトを作成します。

以下は、WebtFieldSetクラスの構築子です。

- プロトタイプ

```
public WebtFieldSet(WebtBuffer buffer)
```

- パラメータ

パラメータ	説明
buffer	WebtBuffer.BT_FIELDタイプのWebtBufferオブジェクトです

- 例外

例外	説明
java.lang.IllegalArgumentException	bufferがNULLの場合に発生します
WebtBufferException	bufferのバッファ型がBT_FIELDタイプでない場合に発生します

メソッド

以下は、WebtFieldSetクラスのメソッド一覧です。

- public void add(int key|String key , byte[] value|double value |long value|String value [,String charset])
 - フィールドキーの値またはフィールドキーの名前がkeyのフィールド・データを追加します。
 - パラメータ

パラメータ	説明
key	フィールドキーの値またはフィールドキーの名前です
value	追加するフィールド・データです
charset	使用する文字セットです

– 例外

例外	説明
WebtBufferException	Tmaxサービスを受信するための送受信バッファ(WebtBuffer)を使用中にエラーが生じた場合に発生します

- public int count(int key)

– フィールドキーの値がkeyと同じフィールド・データの要素数を返します。

– パラメータ

パラメータ	説明
key	フィールドキーの値です

– 例外

例外	説明
WebtBufferException	Tmaxサービスを受信する送受信バッファ(WebtBuffer)を使用中にエラーが発生しました

- public WebtFieldElement get(int key|String key[, int index])

– フィールドキーの値またはフィールドキーの名前がkeyのデータを返します。indexがある場合、index番目のデータを返します。

– パラメータ

パラメータ	説明
key	フィールドキーの値またはフィールドキーの名前です
index	フィールドデータの要素の順番です

– 例外

例外	説明
WebtBufferException	Tmaxサービスを受信するための送受信バッファ(WebtBuffer)を使用中にエラーが生じた場合に発生します

- public byte getByte(int key|String key[, int index])

- フィールドキーの値またはフィールドキーの名前がkeyのフィールド・データをbyte型で表現して返します。indexがある場合、index番目のデータを返します。

- パラメータ

パラメータ	説明
key	フィールドキーの値またはフィールドキーの名前です
index	フィールド・データの要素の順番です

- 例外

例外	説明
WebtBufferException	Tmaxサービスを受信するための送受信バッファ(WebtBuffer)を使用中にエラーが発生しました

- public byte[] getBytes(int key|String key[, int index])

- フィールドキーの値またはフィールドキーの名前がkeyのフィールド・データをbyte[]形式で表現して返します。indexがある場合、index番目のデータを返します。

- パラメータ

パラメータ	説明
key	フィールドキーの値またはフィールドキーの名前です
index	フィールド・データの要素の順番です

- 例外

例外	説明
WebtBufferException	Tmaxサービスを受信するための送受信バッファ(WebtBuffer)を使用中にエラーが生じた場合に発生します

- public Double getDouble(int key|String key[, int index])

- フィールドキーの値またはフィールドキーの名前がkeyのフィールド・データをDouble型で表現して返します。indexがある場合、index番目のデータを返します。

- パラメータ

パラメータ	説明
key	フィールドキーの値またはフィールドキーの名前です
index	フィールド・データの要素の順番です

- 例外

例外	説明
WebtBufferException	Tmaxサービスを受信するための送受信バッファ(WebtBuffer)使用中にエラーが生じた場合に発生します

● public Float getFloat(int key|String key[, int index])

- フィールドキーの値またはフィールドキーの名前がkeyのフィールド・データをFloat型で表現して返します。indexがある場合、index番目のデータを返します。

- パラメータ

パラメータ	説明
key	フィールドキーの値またはフィールドキーの名前です
index	フィールド・データの要素の順番です

- 例外

例外	説明
WebtBufferException	Tmaxサービスを受信するための送受信バッファ(WebtBuffer)使用中にエラーが発生しました

● public Integer getInt(int key|String key[, int index])

- フィールドキーの値またはフィールドキーの名前がkeyのフィールド・データをInteger型で表現して返します。indexがある場合、index番目のデータを返します。

- パラメータ

パラメータ	説明
key	フィールドキーの値またはフィールドキーの名前です

パラメータ	説明
index	フィールド・データの要素の順番です

– 例外

例外	説明
WebtBufferException	Tmaxサービスを受信するための送受信バッファ(WebtBuffer)を使用中にエラーが生じた場合に発生します

- public Long getLong(int key|String key[, int index])

– フィールドキーの値またはフィールドキーの名前がkeyのフィールド・データをLong型で表現して返します。indexがある場合、index番目のデータを返します。

– パラメータ

パラメータ	説明
key	フィールドキーの値またはフィールドキーの名前です
index	フィールド・データの要素の順番です

– 例外

例外	説明
WebtBufferException	Tmaxサービスを受信するための送受信バッファ(WebtBuffer)を使用中にエラーが生じた場合に発生します

- public Short getShort(int key|String key[, int index])

– フィールドキーの値またはフィールドキーの名前がkeyのフィールド・データをShort型で表現して返します。indexがある場合、index番目のデータを返します。

– パラメータ

パラメータ	説明
key	フィールドキーの値またはフィールドキーの名前です
index	フィールド・データの要素の順番です

– 例外

例外	説明
WebtBufferException	Tmaxサービスを受信するための送受信バッファ(WebtBuffer)を使用中にエラーが生じた場合に発生します

- `public String getString(int key|String key[, String charset] [, int index])`

- フィールドキーの値またはフィールドキーの名前がkeyのフィールド・データをString型で表現して返します。indexがある場合、index番目のデータを返します。

- パラメータ

パラメータ	説明
key	フィールドキーの値またはフィールドキーの名前です
charset	使用する文字セットです
index	フィールド・データの要素の順番です

- 例外

例外	説明
WebtBufferException	Tmaxサービスを受信するための送受信バッファ(WebtBuffer)を使用中にエラーが生じた場合に発生します

- `public java.util.Vector getFieldKeyNames()`

- `WebtSystem.createDefaultFieldKeyTable`で作成されたフィールドキーの名前一覧から、String型で表現されているフィールドキー名の一覧を検索して返します。

- `public void insert(int key|String key, byte[] value/ double value/ long value/ String value[,String charset], int index)`

- フィールドキーの値またはフィールドキーの名前がkeyのフィールド・データを挿入します。String型のデータの場合、文字セットを指定できます。

- パラメータ

パラメータ	説明
key	フィールドキーの値またはフィールドキーの名前です
value	挿入するフィールド・データです
charset	使用する文字セットです
index	フィールド・データを挿入する位置です

- 例外

例外	説明
WebtBufferException	keyに該当するWebtFieldオブジェクトがない場合に発生します

● public void remove(int key|String key[, int index])

- フィールドキーの値またはフィールドキーの名前がkeyのフィールド・データの要素を削除します。

- パラメータ

パラメータ	説明
key	フィールドキーの値またはフィールドキーの名前です
index	削除するフィールド・データの要素の順番です

- 例外

例外	説明
WebtBufferException	Tmaxサービスを受信するための送受信バッファ(WebtBuffer)使用中にエラーが生じた場合に発生します

● public void removeAll(int key|String key)

- フィールドキーの値またはフィールドキーの名前がkeyのすべてのフィールド・データの要素を削除します。

- パラメータ

パラメータ	説明
key	フィールドキーの値またはフィールドキーの名前です

- 例外

例外	説明
WebtBufferException	Tmaxサービスを受信するための送受信バッファ(WebtBuffer)使用中にエラーが生じた場合に発生します

例

以下は、WebtFieldSetを使用する例です。

```
WebtConnection connection = new WebtConnection("localhost", 8888);
WebtRemoteService service = new WebtRemoteService("FIELDSET_TEST", connection);
// BT_FIELDタイプのWebtBufferを使用してWebtFieldSetオブジェクトを作成します。
WebtFieldSet sndset = new WebtFieldSet(service.createFieldBuffer());

// WebtFieldSetを使用してWebtBufferにフィールド・データの要素を追加します。
sndset.add("STRING1", "hello!");
// tpcallの呼び出し
WebtBuffer rcvbuf = service.tpcall(sndset.getFieldBuffer());

// Tmaxから受信したWebtBufferを使用してWebtFieldSetオブジェクトを
//作成します。この際、rcvbufのバッファ型はBT_FIELDタイプである必要があります。
WebtFieldSet rcvset = new WebtFieldSet(rcvbuf);

// Tmaxサーバーが送信したフィールド・タイプがINT1のフィールド・データを取り出します。
int value1 = rcvset.getInt("INT1").intValue();

// Tmaxサーバーが送信したフィールド・タイプがINT1の2番目のフィールド・データを
// Stringに変換して取り出します。
String value2 = rcvset.getString("INT1");
...
```

3.5.4. WebtField

WebtFieldクラスは、フィールド・タイプ・バッファで使用するフィールド・キーと、そのフィールドキー値に該当する多数のデータ(WebtFieldElement)を管理する抽象クラスです。特定フィールドキーに対するWebtFieldオブジェクトを取得するには、BufferAccess.createField(int)、BufferAccess.createField(String)、BufferAccess.getField(int)、BufferAccess.getField(String)メソッドを使用します。WebtBufferのバッファ型はWebtBuffer.BT_FIELDである必要があります。

構築子

WebtFieldオブジェクトを作成します。以下は、WebtFieldクラスの構築子についての説明です。

- プロトタイプ

```
WebtField()
```

メソッド

以下は、WebtFieldクラスのメソッド一覧です。

- public void add(<any type> value)

- valueを該当フィールドキー・タイプに変換して追加します。valueがNULLの場合、追加できないので無視します。

- パラメータ

パラメータ	説明
value	フィールドキー・タイプで追加されるデータです

- 例外

例外	説明
WebtBufferException	Tmaxサービスを受信するため、送受信バッファ(WebtBuffer)を使用中にエラーが生じた場合に発生します

- public void insert(<any type> value)

- valueを該当フィールドキー・タイプに変換して挿入します。valueがNULLの場合、挿入できないので無視します。

- パラメータ

パラメータ	説明
value	フィールドキータイプで挿入されるデータです

- 例外

例外	説明
WebtBufferException	Tmaxサービスを受信するため、送受信バッファ(WebtBuffer)を使用中にエラーが生じた場合に発生します

- public WebtFieldElement get()

- 最初のデータを削除して返します。

- 例外

例外	説明
WebtBufferException	Tmaxサービスを受信するため、送受信バッファ(WebtBuffer)を使用中にエラーが生じた場合に発生します

- public WebtFieldElement remove()

- 該当フィールド・タイプのバッファの0番目の値を削除します。

- 例外

例外	説明
WebtBufferException	Tmaxサービスを受信するため、送受信バッファ(WebtBuffer)を使用中にエラーが生じた場合に発生します

- Object[] getAll()

- すべてのデータをWebtFieldElement[]形式で返します。

- 例外

例外	説明
WebtBufferException	Tmaxサービスを受信するため、送受信バッファ(WebtBuffer)を使用中にエラーが生じた場合に発生します

- public String getDefaultCharset()

- WebtFieldオブジェクトに保存されているデータをString型に変換する際や、String型データをbyte[]に変換する際に使用する文字セットを返します。

- public java.util.Enumeration getFieldEnumeration()

- すべてのデータをWebtFieldElementのEnumeration形式で返します。

- 例外

例外	説明
WebtBufferException	Tmaxサービスを受信するため、送受信バッファ(WebtBuffer)を使用中にエラーが生じた場合に発生します

- public int getFieldKey()

- WebtFieldオブジェクトのフィールドキーの値を返します。

- public int getFieldTypes()

- WebtFieldオブジェクトのフィールドキーのタイプを返します。

- public java.util.Vector getFieldVectors()

- すべてのデータをWebtFieldElementのVector型で返します。

- 例外

例外	説明
WebtBufferException	Tmaxサービスを受信するため、送受信バッファ(WebtBuffer)使用中にエラーが生じた場合に発生します

- public boolean isNumericType()

- WebtFieldオブジェクトのフィールドキーが数字型(Numeric Type)であるかをチェックします。

- public void removeAll()

- すべてのデータを削除します。

- public void replace(<any type> value, int index)

- index番目の値を該当フィールド・タイプに変換した値に置換します。valueがNULLの場合は変換できないので無視します。

- パラメータ

パラメータ	説明
value	置換するデータです。データ型がByte、Short、Integer、Long、Float、Double、Stringでなければreplace(value.toString(), index)を実行します
index	置換されるフィールド・データの要素の順番です

- 例外

例外	説明
WebtBufferException	Tmaxサービスを受信するため、送受信バッファ(WebtBuffer)使用中にエラーが生じた場合に発生します

- public void setDefaultCharset(String charset)

- WebtField

オブジェクトに保存されているデータをString型に変換する際や、String型データをbyte[]に変換する際に使用する文字セットを返します。指定していない場合、WebtRemoteServiceで指定した値を使用します。

- パラメータ

パラメータ	説明
charset	指定する文字セットです

- public int size()

- 保存されているデータの数を返します。

- public java.lang.String toString()

- フィールドの内容をユーザーが認識できる文字列で出力します。Tmaxのfbprint()と同じ機能です。

例

以下は、WebtFieldクラスを使用する例です。

```
WebtConnection connection = new WebtConnection("localhost", 8888);
WebtRemoteService service = new WebtRemoteService("TEST", connection);
WebtBuffer fieldBuf = service.createFieldBuffer();

// フィールドキー値がINT1のWebtFieldオブジェクトを作成します。
WebtField field1 = fieldBuf.createField(INT1);

// フィールドキー名が"INT2"のWebtFieldオブジェクトを作成します。
WebtField field2 = fieldBuf.createField("INT2");

// 作成したWebtFieldオブジェクトにデータを追加します。
field1.add("1");
field1.add((int)2);
field1.add(123.45);...

// WebtFieldオブジェクトから上で追加したデータを順番に取り出します。
int data1 = field1.get().intValue(); // == 1
int data2 = field1.get().intValue(); // == 2
```

```
int data3 = field1.get().intValue(); // == 123
....
```

WebtField.addまたはWebtField.insertメソッドを使用して新規フィールド・データを追加する場合や挿入する場合、フィールドキー・タイプと追加するデータ型によって、実際に追加または挿入されるデータの値が異なることがあります。この際に適用される規則についてはWebtField.addメソッドとWebtField.insertメソッドを参照してください。Tmaxサーバーから転送されたフィールド・データにアクセスする際はWebtFieldElementのメソッドを使用します。この際のデータ変換についてはWebtFieldElementを参照してください。

3.5.5. WebtFieldBuffer

WebtBufferの下位クラスで、WebtFieldタイプのデータを保持しているWebtBufferオブジェクトです。

構築子

WebtFieldBufferオブジェクトを作成します。以下は、WebtCarrayBufferクラスの構築子についての説明です。

- プロトタイプ

```
WebtFieldBuffer()
```

メソッド

以下は、WebtFieldBufferクラスのメソッド一覧です。

- public void clear()
 - 内部のすべてのWebtFieldオブジェクトを削除します。
- public WebtField createField(int fldkey|String fldkeyName)
 - fldkeyをフィールドキーの値にするWebtFieldオブジェクトを作成して返します。fldkeyの値と同じフィールドキー値をもつWebtFieldオブジェクトがすでに存在する場合は、そのオブジェクトを返します。
 - fldkeyNameを使用してフィールドキーの名前を入力してWebtFieldオブジェクトを作成する場合は、WebtSystem.createDefaultFieldKeyTable(String)を利用して使用するフィールドキー・テーブルを作成前にロードします。
 - パラメータ

パラメータ	説明
fldkey	作成するWebtFieldオブジェクトのフィールドキーの値です

パラメータ	説明
fldkeyName	作成するWebtFieldオブジェクトのフィールドキーの名前です

– 例外

例外	説明
WebtBufferException	fieldkeyが有効な値でない場合やフィールドキー・テーブルで見つからなかった場合に発生します

- public WebtField getField(int fldkey|String fldkeyName)

– fieldkeyをフィールドキー値にするかfldkeyNameをフィールドの名前にするWebtFieldオブジェクトを返します。存在しない場合はNULLを返します。

– 例外

例外	説明
WebtBufferException	fieldkeyが有効な値でない場合やフィールドキー・テーブルで見つからなかった場合に発生します

- public java.util.Vector getFieldKeyNames()

– String型で表現されるフィールドキー一覧を返します。WebtSystem.createDefaultFieldKeyTable(String)によって作成されたフィールドキー・テーブルから探します。

– 例外

例外	説明
WebtBufferException	Tmaxサービスを受信するため、送受信バッファ(WebtBuffer)を使用中にエラーが生じた場合に発生します

- public java.util.Vector getFields()

– 作成された順序どおりソートしたWebtFieldオブジェクトのVectorを返します。

– 例外

例外	説明
WebtBufferException	Tmaxサービスを受信するため、送受信バッファ(WebtBuffer)を使用中にエラーが生じた場合に発生します

- public WebtField removeField(int fldkey)

– fieldkeyをフィールドキーの値にするWebtFieldオブジェクトを見つけて削除します。削除されるWebtFieldオブジェクトが存在しない場合はNULLを返します。

– 例外

例外	説明
WebtBufferException	Tmaxサービスを受信するため、送受信バッファ(WebtBuffer)を使用中にエラーが生じた場合に発生します

- public WebtField removeField(String fldkeyName)

– fieldkeyをフィールドキーの値にするWebtFieldオブジェクトを見つけて削除します。削除されるWebtFieldオブジェクトが存在しない場合はNULLを返します。

– 例外

例外	説明
WebtBufferException	fieldkeyが有効な値でない場合やフィールドキー・テーブルで見つからなかった場合に発生します

3.6. イベント・メッセージ処理関連クラス

3.6.1. WebtEventConnecion

Tmaxサーバーまたは他のクライアントが転送したイベント・メッセージ(tppost、tpnotify、tpsendtocli、tpbroadcast)を処理するWebtConnectionです。

Tmaxサーバーまたは他のクライアントが送信した非要求メッセージを受信して処理するためには、必ずWebtEventConnecionを使用してTmaxサーバーに接続する必要があります。また、受信したイベント・メッセージを処理するためのcallbackメソッドを定義したWebtEventHandlerを実装し、WebtEventConnecionのsetEventHandlerメソッドを使用してイベント・ハンドラを登録する必要があります。

構築子

WebtEventConnecionオブジェクトを作成します。以下は、WebtEventConnecionクラスの構築子についての説明です。

- 構築子1

– プロトタイプ

```
public WebtEventConnection(String hostAddr,  
                             int hostPort)
```

– パラメータ

パラメータ	説明
hostAddr	hostPortのTmaxサーバーに接続設定したWebtEventConnectionオブジェクトを作成します
hostPort	メインTmaxサーバーのポート番号です

● 構築子2

– プロトタイプ

```
public WebtEventConnection(String hostAddr,  
                             int hostPort,  
                             boolean doConnect)
```

– パラメータ

パラメータ	説明
hostAddr	メインTmaxサーバーのIPアドレスです。hostAddr:hostPortのTmaxサーバーに接続を設定したWebtEventConnectionを作成します
hostPort	メインTmaxサーバーのポート番号です
doConnect	– TRUE: WebtEventConnectionオブジェクトを作成しつつ、Tmaxサーバーに接続を行います – FALSE: WebtEventConnectionオブジェクトを作成しつつ、Tmaxサーバーに接続を行いません

● 構築子3

– プロトタイプ

```
public WebtEventConnection(String hostAddr,  
                             int hostPort,  
                             boolean doConnect,  
                             int conntimeout)
```

– パラメータ

パラメータ	説明
hostAddr	メインTmaxサーバーのIPアドレスです。hostPortのTmaxサーバーに接続を設定したWebtEventConnectionオブジェクトを作成します
hostPort	メインTmaxサーバーのポート番号です
doConnect	- TRUE : WebtEventConnectionオブジェクトを作成しつつ、Tmaxサーバーに接続を行います。タイムアウト時間内に接続を設定できなかった場合は例外(tperrno = TPETIME)が発生します - FALSE: WebtEventConnectionオブジェクトを作成しつつ、Tmaxサーバーに接続を行いません
conntimeout	接続設定時のタイムアウト時間です(単位: 秒)

- 構築子4

- プロトタイプ

```
public WebtEventConnection(String hostAddr,
                           int hostPort,
                           int conntimeout)
```

- パラメータ

パラメータ	説明
hostAddr	メインTmaxサーバーのIPアドレスです。hostAddr:hostPortのTmaxサーバーに接続を設定したWebtEventConnectionオブジェクトを作成します。タイムアウト時間内に接続を設定できなかった場合は例外(tperrno = TPETIME)が発生します
hostPort	メインTmaxサーバーのポート番号です
conntimeout	接続設定時のタイムアウト時間です(単位: 秒)

- 構築子5

- プロトタイプ

```
public WebtEventConnection(String hostAddr,
                           int hostPort,
                           String backupAddr,
                           int backupPort)
```

- パラメータ

パラメータ	説明
hostAddr	メインTmaxサーバーのIPアドレスです。hostPortのTmaxサーバーに接続を設定したWebtEventConnectionオブジェクトを作成します。hostAddr:hostPortに接続を設定できなかった場合、backupAddr:backupPortに接続を行います
hostPort	メインTmaxサーバーのポート番号です
backupAddr	バックアップTmaxサーバーのIPアドレスです
backupPort	バックアップTmaxサーバーのポート番号です

- 構築子6

- プロトタイプ

```
public WebtEventConnection(String hostAddr,
                           int hostPort,
                           String backupAddr,
                           int backupPort,
                           boolean doConnect,
                           int conntimeout)
```

- パラメータ

パラメータ	説明
hostAddr	メインTmaxサーバーのIPアドレスです。hostPortのTmaxサーバーに接続を設定したWebtEventConnectionオブジェクトを作成します。hostAddr:hostPortに接続を設定できなかった場合はbackupAddr:backupPortに接続を行います。doConnectがFALSEの場合は接続を行いません。タイムアウト時間内に接続を設定できなかった場合は例外(tperrno = TPETIME)が発生します
hostPort	メインTmaxサーバーのポート番号です
backupAddr	バックアップTmaxサーバーのIPアドレスです
backupPort	バックアップTmaxサーバーのポート番号です
doConnect	TRUEの場合はWebtEventConnectionオブジェクトを作成しつつ、Tmaxサーバーに接続を行います
conntimeout	接続設定時のタイムアウト時間です(単位: 秒)

- 構築子7

- プロトタイプ

```
public WebtEventConnection(String hostAddr,
                           int hostPort,
```

```
String username,
String userpwd,
String domainname,
String domainpwd)
```

– パラメータ

パラメータ	説明
hostAddr	メインTmaxサーバーのIPアドレスです。hostPortのTmaxサーバーに接続を設定したWebtEventConnectionオブジェクトを作成します。接続を設定する際にパラメータとして与えられた認証情報を使用して認証手続きを実行します。Tmaxサーバーに登録されているusername、userpwd、domainname、domainpwd値と一致した場合に接続が正常に行われます
username	ユーザー認証アカウントの名前です
userpwd	ユーザー認証パスワードです
domainname	クライアントの名前です
domainpwd	システム接続セキュリティ・パスワードです

● 例外

例外	説明
WebtIOException	接続を初期化中にネットワーク・エラーが発生し、hostAddr:hostPortとbackupAddr:backupPortに接続を設定できなかった場合や、タイムアウト時間内に接続を設定できなかった場合に発生します
WebtServiceException	ソケット接続を行う際に割り込みがあった場合に発生します

メソッド

以下は、WebtEventConnecionクラスのメソッド一覧です。

- public boolean close()
 - Tmaxとソケット接続を終了します。コネクション・プールで管理されるコネクションの場合、接続を終了せずにコネクション・プールに返却して再使用します。
- public void setEventHandler(WebtEventHandler event [,int qlen])
 - イベント・メッセージを処理するハンドラを登録します。
 - パラメータ

パラメータ	説明
event	イベント・メッセージを受信した際に処理するハンドラです
qlen	イベント・メッセージを保存する内部キューのサイズです。(デフォルト値:2) キューに空きスロットがない場合は一番古いメッセージを捨てます

– 例外

例外	説明
WebtIOException	ネットワーク・エラーが生じた場合に発生します
WebtServiceException	キューのサイズが0の場合や、ソケット接続を行う際に割り込みがあった場合に発生します

● public void setEventMask(int mask)

- tpbroadcast()、tppost()、tpsendtocli()、tpnotify()が送信したイベント・メッセージのうち受信したいメッセージを選択します。

– パラメータ

パラメータ	説明
mask	WebtEventMask.EVENT_BROADCAST、WebtEventMask.EVENT_POST、WebtEventMask.EVENT_SENDCLI、WebtEventMask.EVENT_NOTIFYをOR()組み合わせをした値をもちます。 該当maskがonになると、該当メッセージを受信して渡します。この関数を実行しない場合、デフォルト値はWebtEventMask.EVENT_NONE(0)であり、メッセージを何も渡しません。WebtEventMask.EVENT_ALLの場合、すべてのイベント・メッセージを渡します

– 例外

例外	説明
WebtIOException	ネットワーク・エラーが生じた場合に発生します
WebtServiceException	正しくないパラメータが入力された場合に発生します

● void tpsubscribe(String eventName [,WebtAttribute attr])

- tppost()メッセージを受信するためにpost groupに加入します。

– パラメータ

パラメータ	説明
eventName	受信するtpost()メッセージのpost groupの名前です
attr	tpsubscribe()を実行する際に適用する属性です。現在は意味のあるフラグが存在しません

– 例外

例外	説明
WebtIOException	ネットワーク・エラーが生じた場合に発生します
WebtServiceException	eventNameが長過ぎる場合や割り込みがあった場合に発生します

- void tpunsubscribe(String eventName [,WebtAttribute attr])

– tpostメッセージを受信しないために、加入したpost groupから脱退します。

– パラメータ

パラメータ	説明
eventName	脱退するpost groupの名前です
attr	tpunsubscribeの実行時に適用する属性です。現在は意味のあるフラグが存在しません

– 例外

例外	説明
WebtIOException	ネットワーク・エラーが生じた場合に発生します
WebtServiceException	割り込みがあった場合に発生します

例

以下のプログラムは、tpostメッセージを受信するためのイベント・ハンドラの例です。

```
import tmax.webt.*;

public class EventSample implements WebtEventHandler {
    private WebtEventConnection connection;
    public EventSample(WebtEventConnection connection) {
        this.connection = connection;
    }
}
```

```

// tpbroadcastイベントのみを受信するためにmaskを設定します。
connection.setEventMask(WebtEventConnection.EVENT_POST);

// イベント・ハンドラを登録します。
connection.setEventHandler(this);

// eventに加入します。
connection.tpsubscribe("posttest");
}

public void destroy() {
    connection.tpunsubscribe("posttest");
    connection.close();
}

// WebtEventHandlerの関数です。
// イベント・メッセージを受信した場合、このメソッドがコールバックされます。
public void handleEvent(int type, WebtBuffer buf, int len, int flags) {
    System.out.println("event received. type = " + type + ",
        buffer type = " + buf.getBufferType());
    System.out.println("event msg : " + buf.getString());
}

// WebtEventHandlerのメソッドです。WebTがイベントを処理中に例外に直面すると
// このメソッドをコールバックします。WebtIOExceptionが発生した場合、Tmaxサーバーへの接続が
// 切断している可能性があるため、再接続します。
public void handleError(WebtException e) {
    System.err.println("event error occurred");
    e.printStackTrace();
    if (e instanceof WebtIOException) {
        if (e.getTPError() == WebtException.TPECLOSE) {
            connection.close();
            connection.connect();
        }
    }
}
}
}

```

3.6.2. WebtEventHandler

WebtEventHandlerは、Tmaxサーバーから受信した非要求メッセージを処理する際に使用するインターフェースです。非要求メッセージを受信するためには、このインターフェースを実装する必要があります。非要求メッセージには、tpsendtocli、tpbroadcast、tpnotify、tppost acall_replyがあります。

フィールド

- final int ACALLREPLY_TYPE
 - tpacallの応答で受信したデータを意味します。
- static final int BROADCAST_TYPE
 - tpbroadcastが送信したデータを意味します。
- static final int NOTIFY_TYPE
 - tnotifyが送信したデータを意味します。
- static final int POST_TYPE
 - tppostが送信したデータを意味します。
- static final int SENDTOCLI_TYPE
 - tpsendtocliが送信したデータを意味します。

メソッド

以下は、WebtEventHandlerクラスのメソッド一覧です。

- void handleError(WebtException e)
 - 非要求メッセージ処理中にエラーが発生した場合、このhandleErrorメソッドが呼び出されます。発生するエラーは大部分がネットワーク・エラーです。

– パラメータ

パラメータ	説明
e	Exceptionオブジェクトです

- void handleEvent(int type, WebtBuffer buf, int len, int flags)
 - WebtEventConnection.setEventMask(int)によって選択された非要求メッセージが受信されると、このhandleEventメソッドがコールバックされます。

– パラメータ

パラメータ	説明
type	POST_TYPE、BROADCAST_TYPE、NOTIFY_TYPE、SENDTOCLI_TYPE、ACALLREPLY_TYPEのうち1つです
buf	非要求メッセージの内容を格納しているWebtBufferです
len	非要求メッセージの長さです
flags	非要求メッセージに設定されているフラグ値です

3.7. その他のクラス

3.7.1. WebtTransaction

WebtTransactionクラスは、複数のサービスを1つの作業単位にまとめて処理できるようにAPIを提供し、例外の発生を処理します。

構築子

WebtTransactionオブジェクトを作成します。以下は、WebtTransactionクラスの構築子についての説明です。

- プロトタイプ

```
public WebtTransaction(WebtConnection conn)
```

- パラメータ

パラメータ	説明
conn	トランザクションが実行されるWebtConnectionオブジェクトです

メソッド

以下は、WebtTransactionクラスのメソッド一覧です。

- public int begin()
 - グローバル・トランザクションを開始する関数です。TX関数のint tx_begin()と同じ機能を実行します。
 - 例外

例外	説明
WebtTXException	すでにトランザクションが開始された場合に発生します

- public int commit()

- グローバル・トランザクションをコミットする関数です。TX関数のint tx_commit()と同じ機能を実行します。

- 例外

例外	説明
WebtTXException	トランザクションが開始されていない状態である場合、ネットワークにエラーが生じた場合、Tmaxサーバーでトランザクション関連エラーが生じた場合に発生します

- public int rollback()

- グローバル・トランザクションをロールバックする関数です。TX関数のint tx_rollback()と同じ機能を実行します。

- 例外

例外	説明
WebtTXException	tx_beginができない状態の場合、ネットワークにエラーが生じた場合、Tmaxサーバーでトランザクション・エラーが生じた場合に発生します

- public int getBlockTimeout()

- commitメソッドとrollbackメソッドを呼び出す際に適用されるタイムアウト時間を返します。

- int getTimeout()

- トランザクションのタイムアウト時間を返します。

- public boolean isChained()

- chain controlの可否を返します。

戻り値	説明
TRUE	TX_CHAINEDが設定されていることを意味します
FALSE	TX_UNCHAINEDが設定されていることを意味します

- `public void setBlockTimeout(int timeout)`

- `commit`メソッドと`rollback`メソッドを呼び出す際に適用されるタイムアウト時間を指定します。

- パラメータ

パラメータ	説明
timeout	タイムアウト時間です(単位:ミリ秒)

- `public void setChained(boolean value)`

- `chain control`を実行します。

- パラメータ

パラメータ	説明
value	chain controlの値です。 - TRUE : TXメソッド <code>int tx_set_transaction_control(TX_CHAINED)</code> と同じ機能を実行します - FALSE : TXメソッド <code>int tx_set_transaction_control(TX_UNCHAINED)</code> と同じ機能を実行します

- `public void setTimeout(int timeout)`

- トランザクションのタイムアウト時間を指定します。`setTimeout`関数によってタイムアウト時間を別途指定しなければ、Tmax環境設定ファイルのTXTIMEに指定されている値が適用されます。

- パラメータ

パラメータ	説明
timeout	タイムアウト値です(単位:ミリ秒)

参考

TX関数の詳細については『Tmax アプリケーション開発ガイド』を参照してください。

3.7.2. WebtSystem

WebTシステムのバージョン、文字セット、フィールドキー・テーブルなどの情報を管理するクラスです。

構築子

WebtSystemオブジェクトを作成します。以下は、WebtSystemクラスの構築子についての説明です。

- プロトタイプ

```
public WebtSystem()
```

メソッド

以下は、WebtSystemクラスのメソッド一覧です。

- public static void createDefaultFieldKeyTable(String fdlfile)

– フィールドキー・テーブルを作成します。

– パラメータ

パラメータ	説明
fdlfile	fdlファイルのパスです

– 例外

例外	説明
java.io.IOException	fdlファイルを見付けられなかった場合や、読み込む際にエラーが生じた場合に発生します

- public static String getDefaultCharset()

– WebTシステムの基本文字セットを返します。

- public static WebtFDLKeyTable getDefaultFieldKeyTable()

– 基本フィールドキー・テーブルを返します。

- public static void main(String[] argv)

– WebtSystemを実行するとWebTバージョン情報を標準出力(standard out)します。

- public static void setDefaultCharset(String charset)

– WebTで使用する基本文字セットを指定します。

- WebtServiceオブジェクトやWebtRemoteServiceオブジェクトを作成する際に文字セットを指定していない場合、setDefaultCharsetメソッドで指定した値をデフォルト値とします。

- パラメータ

パラメータ	説明
charset	基本文字セットです

- public static String version()
 - WebTバージョンを返します。

第4章 WebT HMS

本章では、WebTを使用してTmaxに提供されるHMS(Hybrid Messaging System)に接続して、メッセージング・サービスを使用する方法について説明します。

4.1. 概要

メッセージング・システムとは、送信元と受信元間の疎結合を可能にする通信媒体です。各自送受信するメッセージがとどまるチャンネルについての情報だけを使って通信を行う疎結合が可能な媒体として、送信元と受信元は仮想のチャンネルについての情報だけで相互通信が可能です。TmaxはHMSを提供し、Tmaxサーバーでの設定を通じてメッセージング・システムの接続チャンネルを起動できます。

WebTからHMSへの接続およびメッセージの送受信のための全ての動作は、Sun Microsystemsで定めたJava基盤のメッセージング・システム標準であるJava Messaging Service(以下、JMS)によって定義されたインターフェースによって行われます。WebTでは、JMSインターフェースを使用するクラスを提供するため、ユーザーはJMSで定義されているクラスと関数を使用することで、HMSを使用したTmaxサーバーとのメッセージの送受信を実装することが可能です。

参考

TmaxでのHMS構造についての詳細は『Tmax HMSユーザガイド』を参照してください。

4.2. JMS API

JMSに定義されているクラスの役割は以下のとおりです。

- ConnectionFactory
 - ユーザーがJMSを使用するために接続する最上位クラスで、メッセージング・システムに接続するコネクションを作成します。
 - WebTではWebtConnectionGroupクラスがConnectionFactoryクラスの役割を行います。既存の方式どおり設定された環境から取得したWebtConnectionGroupをConnectionFactoryにキャストすると、WebTでConnectionFactoryを使用できます。
 - 下位クラス: QueueConnectionFactory, TopicConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory
- Connection

- メッセージング・システムに接続されているコネクションを意味します。
- ここでは、メッセージを送受信する過程のトランザクションを管理できるセッションを作成します。
- 下位クラス: QueueConnection, TopicConnection, XAQueueConnection, XATopicConnection

- Session
 - トランザクション単位を処理できるメッセージング・システムとの接続コンテキストです。
 - メッセージの送受信に必要な主要クラスのクライアント、メッセージ、送信先を作成します。
 - 下位クラス: QueueSession, TopicSession, XAQueueSession, XATopicSession

- MessageProducer
 - メッセージを作成し、送信するクライアントです。
 - 下位クラス: QueueSender, TopicPublisher

- MessageConsumer
 - メッセージを受信するクライアントです。
 - 下位クラス: QueueReceiver, TopicSubscriber

- QueueBrowser
 - キュー形式の送信先に限定する受信元で、メッセージを消費せずに照会できるクライアントです。

- Destination
 - メッセージがとどまるメッセージング・システムのチャンネルで、キューとトピックの2つの形式があります。
 - 下位クラス: Queue, Topic

- Message
 - 送受信されるメッセージを意味し、複数の形式を提供します。
 - 下位クラス: BytesMessage, MapMessage, ObjectMessage, StreamMessage, TextMessage

WebT JMSのサポート

現在WebTは以下のようなクラスをサポートしています。

- Connection

- 下位クラス : QueueConnection, TopicConnection, XAConnection, XAQueueConnection, XATopicConnection

- 以下の項目を除いたすべてのメソッドをサポートします。

```
ConnectionConsumer createConnectionConsumer(Destination destination,  
                                             String messageSelector,  
                                             ServerSessionPool sessionPool,  
                                             int maxMessages)
```

```
ConnectionConsumer createDurableConnectionConsumer(Topic topic,  
                                                    String subscriptionName,  
                                                    String messageSelector,  
                                                    ServerSessionPool sessionPool,  
                                                    int maxMessages)
```

- ConnectionFactory

- 下位クラス : QueueConnectionFactory, TopicConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

- すべてのメソッドをサポートします。

- Session, QueueSession, TopicSession, XASession, XAQueueSession, XATopicSession

- すべてのメソッドをサポートします。

- MessageProducer, QueueSender, TopicPublisher

- すべてのメソッドをサポートします。

- MessageConsumer, QueueReceiver, TopicSubscriber

- すべてのメソッドをサポートします。

- QueueBrowser

- すべてのメソッドをサポートします。
- Message, BytesMessage, TextMessage, MapMessage, StreamMessage, ObjectMessage
 - すべてのメソッドをサポートします。
- MessageListener
 - ユーザーの実装インターフェースをサポートします。
- ExceptionListener
 - ユーザーの実装インターフェースをサポートします。

参考

各クラスおよびメソッドの正確な意味については <http://java.sun.com/j2ee/1.4/docs/api/javax/jms/package-tree.html>を参照してください。

4.3. 環境設定

TmaxのHMSに接続するには、Tmaxの環境設定を通じてHMSを起動し、WebTで作成したConnection GroupをHMSに接続できるように、webt.propertiesファイルあるいはJEUSのJeusMain.xmlファイルにHMS関連の設定を行います。

4.3.1. Tmaxの設定

HMSを使用するためには、Tmax環境ファイルのDOMAIN、NODE、SVRGROUP、HMSセクションの設定が必要です。

以下は、HMSを使用する環境ファイルの例です。

```
*DOMAIN
Domain1      SHMKEY=0x11936,
              TMMLOGLVL = "DEBUG1",
              CLHLOGLVL = "DEBUG1",
              MINCLH = 1,
              MAXCLH = 1,
              TPORTNO = 7783,
              MAXSESSION = 1024

*NODE
```

```

Node1      TMAXDIR = "/data/tmax",
           APPDIR  = "/data/tmax/appbin/",
           MAXSESSION = 1024

*SVRGROUP
svghms     NODENAME = "node1", CPC=1, SVGTYPE="HMS", RESTART=N,
           OPENINFO = "ORACLE_XA+Acc=P/scott/tiger+SesTm=60+Threads=true",
           HMSSUBSCFG = "/data/tmax/config/hmssubconfig",
           HMSNAME = "hms_ora", HMSSINDEX = 1,
           HMSMSGLIVE = 1, HMSMAXTHR = 4, HMSMAXDBTHR = 10

*SERVER
...

*SERVICE
...

*HMS
queue1     SVGNAME = svghms,
           TYPE = "QUEUE",
           BOOT = "WARM",
           GLOBAL = N
topic1     SVGNAME = svghms,
           TYPE = "TOPIC",
           BOOT = "WARM",
           GLOBAL = N

```

- DOMAINセクション、NODEセクション

MAXSESSION項目の設定が可能です。

- SVRGROUPセクション

SVRGROUP、NODENAME、SVGTYPE、HMSNAME、OPENINFO、HMSINDEX、HMSMAXTHR、HMSMAXDBTHR項目の設定が必要で、HMSSUBSCFG、HMSMSGLIVE、HMSPORT、HMSHEARTBEAT、HMSGQINT項目の設定が可能です。

- HMSセクション

SVRNAME、TYPE項目の設定が必要で、BOOT、GLOBAL、GQTHR、GQMAXREQ、GQFULL項目の設定が可能です。

参考

各設定についての詳細は『Tmax』の「第2章 HMSの環境設定」を参照してください。

4.3.2. JEUSMain.xmlの設定

JEUSの環境ファイルのJEUSMain.xmlのexternal-resourceに設定するWebtConnectionPoolにHMS設定を追加すると、HMSに接続できます。

以下は、JEUSMain.xmlを設定する例です。

```
<resource>
  ....
  <external-resource>
    <property-group>
      <name-prefix>tmax2</name-prefix>
      <property>
        <key>type</key>
        <value>shared</value>
      </property>
      <property>
        <key>host-name</key>
        <value>192.168.35.218</value>
      </property>
      <property>
        <key>host-port</key>
        <value>8888</value>
      </property>
      <property>
        <key>min</key>
        <value>10</value>
      </property>
      <property>
        <key>max</key>
        <value>20</value>
      </property>
      <property>
        <key>step</key>
        <value>2</value>
      </property>
      <property>
        <key>check-acquired</key>
        <value>true</value>
      </property>
      <property>
        <key>support-xa</key>
        <value>true</value>
      </property>
      <property>
        <key>enable-autoclose</key>
        <value>true</value>
      </property>
    </property-group>
  </external-resource>
</resource>
```

```

        </property>
        <property>
            <key>connect-timeout</key>
            <value>2</value>
        </property>
        <property>
            <key>jms-name</key>
            <value>hms01</value>
        </property>
        <property>
            <key>jms-type</key>
            <value>queue</value>
        </property>
    </property-group>
</external-resource>
    ....
</resource>

```

以下は、HMS設定と関連する項目についての説明です。

項目	区分	説明
jms-name	設定値	String
	デフォルト値	NULL
	説明	TmaxのHMSサービスを使用するためのHMS名を指定します。この値が指定された場合、Javaメッセージング・システムのインターフェースを使用して該当コネクションのプールからTmaxのHMSに接続できます
jmsType	設定値	queue topic xaqueue xatopic
	デフォルト値	NULL
	説明	接続するメッセージング・システムのタイプを決定します。jms-nameが指定された場合にのみ有効です

4.3.3. webt.propertiesの設定

webt.propertiesに以下の項目を設定すると、WebTでHMSに接続できます。

以下は、webt.propertiesを設定する例です。

```

#####
# logging related parameters.
#####
# set log level valid values are none, info, debug. default is none.
log.level=debug

# set directory in which the log file places. if not set, log is

```

```

# printed to standard out
log.dir=D:\\

# set the name of the log file. default is webt.log
log.file=webt.log

# set log buffering size. default is 0
log.bufsize=1024

# tmax.webt.xid.log=true

#####
# FDL related parameters
#####
# set the fdl file.
fdl.file=c:\\tmax\\sample\\fdl\\tmax.fdl

# set application wide default character set. default is system default
defaultCharset=euc-kr

# Tmax 3.11.x extended Header size

# extendedHeader=true

#####
# monitoring related parameters
#####
# enable/disable alive check. default is disable(false)
monitoring.pool.checkAlive=true

# set monitoring interval. default is 60sec
monitoring.pool.interval=5

# set monitoring log file valid-days.default is -1
monitoring.log.validDays=1

#####
# connection pool related paramters
#####
# enable/disable connection pool. default is disable(false)
enableConnectionPool=true

```

```
# WebtConnectionGroup name list
connectionPool.groups=tmax1

# set connection group type valid values are shared, non-shared,

# non-shared2. default is shared
connectionPool.tmax1.type=shared

# set Tmax Server Address.
connectionPool.tmax1.hostAddr=192.168.1.123

# set Tmax Server Port.
connectionPool.tmax1.hostPort=8888

# set Backup Tmax Server Address.
connectionPool.tmax1.hostBackupAddr=61.77.153.1

# set Backup Tmax Server Port.
connectionPool.tmax1.hostBackupPort=8889

# set default user name for security.
connectionPool.tmax1.userName=tmax

# set default user password for security.
connectionPool.tmax1.userPasswd=1234

# set default domain name for security.
connectionPool.tmax1.domainName=choco

# set default domain password for security.
connectionPool.tmax1.domainPasswd=1234

# set initial pool size. default is 10
connectionPool.tmax1.initCapacity=1

# set max pool size. default is 20
connectionPool.tmax1.maxCapacity=10

# set increment step size. default is 5
connectionPool.tmax1.incrementRate=2

# set connection idle time. default is 60sec
connectionPool.tmax1.maxIdleTime=60

# set tptimeout.
connectionPool.tmax1.tpTimeout=40
```

```

# set txttimeout.
connectionPool.tmax1.txTimeout=10

# set connection timeout.
connectionPool.tmax1.connectTimeout=10

# enable to event service.

#connectionPool.tmax1.enableEvent=true

# set event service Type.

#connectionPool.tmax1.eventSvcType=all

# set event handler object.

#connectionPool.tmax1.eventHandler=tcpservet.event.GenericEvent
jeus.servlet.webt.autoClose.enable=true
connectionPool.tmax1.jmsName = hms01
connectionPool.tmax1.jmsType = queue

```

以下は、HMSの設定に関連する項目についての説明です。

- connectionPool.<groupName>.jmsName = String
 - デフォルト値: NULL
 - TmaxのHMSサービスを使用するためのHMS名を指定します。この値が指定されている場合、Javaメッセージング・システムのインターフェースを使用して該当コネクション・プールからTmaxのHMSに接続できます。
- connectionPool.<groupName>.jmsType = queue | topic | xaqueue | xatopic
 - デフォルト値: NULL
 - 接続するメッセージング・システムのタイプを決定します。jms-nameが指定されている場合にのみ有効です。

4.4. プログラムの例

以下は、WebTでHMSを使用するJavaプログラムの例です。

```

public class JmsTest1 {
    public static void main(String[] argv) throws Exception{
        System.setProperty("webt.properties", "d:\\webt.properties");
    }
}

```

```

new WebtProperties();

QueueConnectionFactory result =
(QueueConnectionFactory)WebtConnectionPool.getGroup("tmax1");

QueueSession qc =
(QueueSession)((QueueConnectionFactory)result).createQueueConnection().createQueueSession(false,
Session.AUTO_ACKNOWLEDGE));

Queue q = qc.createQueue("queue01");
QueueSender qs = (QueueSender)(qc.createSender(q));
QueueSender qs2 = (QueueSender)(qc.createSender(q));
System.out.println("qs2 id : " + ((WebtQueueSender)qs2).getID());
TextMessage msg = qc.createTextMessage();

msg.setText("qwerty");
msg.setIntProperty("NO", 1);
qs.send(msg,DeliveryMode.PERSISTENT,0,0);
QueueReceiver qr = (QueueReceiver)(qc.createReceiver(q));
BytesMessage msg2 = (BytesMessage)(qr.receive());
byte[] data = new byte[256];
msg2.readBytes(data);
System.out.println("body : " + new String(data));
System.out.println("NO : " + msg2.getIntProperty("NO"));
qr.close();
qs.close();
//qs2.close();
qc.close();

}

```


第5章 モニタリング

本章では、JEUSコンソール・ツールとWebAdminを使用し、WebTおよびJTmaxをモニタリングするための環境設定方法とモニタリング方法について説明します。

5.1. 概要

WebTが起動している際、WebTコネクション・グループの現在の環境設定情報の照会、あるいはコネクション・グループの状態、コネクションの呼び出しおよび応答回数の確認などの管理が必要です。また、JTmaxが起動している場合はJTmaxの環境設定情報および動作しているスレッド数とメソッド呼び出し回数を確認する必要があります。

このために、WebTモジュールはJEUSのコンソール・ツールおよびWebAdminを使用してWebTおよびJTmaxの情報を提供します。

参考

JEUS WebAdminおよびコンソール・ツールを使用したWebT管理機能はJEUS v6.0 Fix#7以上からサポートします。

5.2. 環境設定

WebT管理機能を使用するためには、別途でデプロイされたモジュールの適用など、複数の作業を実行します。

JEUS管理ツール(コンソール・ツールおよびWebAdmin)を使用してWebT AdminおよびJTmax Adminを使用するには、JEUSのインストール・ディレクトリーの下のlib/systemディレクトリーに、別途デプロイされたjext_webtadmin.jarファイルを置きます。

それ以外に必要な作業は以下のとおりです。

- モニタリング関連設定(ディレクトリーおよびmbeanの設定)
- webadmin.applicationの設定
- node.xmlの設定
- catalog_ja.xmlの設定
- jeus.cssの設定

モニタリング関連設定

JEUS WebAdminでWebTモニタリングおよびJTmaxモニタリング・メニューを使用するには、別途デプロイされた「jeus」という名前のディレクトリーを、JEUSがインストールされているディレクトリーの下に、以下の位置に上書きします。

```
webhome/admin_home/WEB-INF/classes
```

モニタリング機能を使用するためには、モニタリングしようとする環境設定ファイルに、以下のようにmbean設定を行います。

- WebTの設定

```
webt.properties  
webt.mbean=true
```

- JTmaxの設定

<JEUSMain.xml>

```
<external-resource>  
  <property>  
    <key>mbean</key>  
    <value>>true</value>  
  </property>  
</external-resource>
```

- JTCの設定

```
tux.properties  
tux.mbean=true
```

webadmin.applicationの設定

JEUSがインストールされているディレクトリーの以下のパスに位置するwebadmin.applicationファイルを編集します。

```
webhome/admin_home/WEB-INF
```

このファイルはXML形式になっています。

<application>ノード下の<page>ノードに以下のような内容を挿入します。

- WebTの設定

```
<!--WEBT PAGEの設定-->
<page name="JeusMonitoringWebtMonitoringService_Overview"
specification-path="/jeus/tool/webadmin/page/JeusMonitoringWebtMonitoringService/Overview.page"/>
<page name="JeusMonitoringWebtMonitoringService_Group"
specification-path="/jeus/tool/webadmin/page/JeusMonitoringWebtMonitoringService/Group.page"/>
```

- JTmaxの設定

```
<!--JTMAX PAGEの設定-->
<page name="JeusMonitoringJTmaxMonitoringService_Overview"
specification-path="/jeus/tool/webadmin/page/JeusMonitoringJTmaxMonitoringService/Overview.page"/>
```

- JTCの設定

```
<!--JTC PAGEの設定-->
<page name="JeusMonitoringJTCMonitoringService_Overview"
specification-path="/jeus/tool/webadmin/page/JeusMonitoringJTCMonitoringService/Overview.page"/>
<page name="JeusMonitoringJTCMonitoringService_Remote"
specification-path="/jeus/tool/webadmin/page/JeusMonitoringJTCMonitoringService/Remote.page"/>
```

node.xmlの設定

JEUSがインストールされているディレクトリーの以下のパスに位置するnode.xmlファイルを編集します。

```
webhome/admin_home/WEB-INF/classes/jeus/tool/webadmin/resource
```

このファイルの<node-specification>ノードの下の<node>ノードに以下のような内容を挿入します。

- WebTの設定

```
<!--WEBTの設定-->
<node j2ee-type="WebtConnectionGroup"
    key="NodeTree.JeusMonitoring.WebtMonitoringService"
    page="JeusMonitoringWebtMonitoringService_Overview" target="rightPane"
    static="true">
    <parent j2ee-type="Folder" jeus-type="JeusMonitoring"/>
    <delegator j2ee-type="JeusService" jeus-type="JeusManager"/>

    <icon state="running" icon="webt_monitor"/>
    <icon state="xml" icon="webt_monitor"/>
    <icon state="failed" icon="webt_monitor"/>
</node>
```

- JTmaxの設定

```

<!--JTMAXの設定-->
<node j2ee-type="JTmaxConnectionGroup"
      key="NodeTree.JeusMonitoring.JTmaxMonitoringService"
      page="JeusMonitoringJTmaxMonitoringService_Overview" target="rightPane"

      static="true">
  <parent j2ee-type="Folder" jeus-type="JeusMonitoring"/>
  <delegator j2ee-type="JeusService" jeus-type="JeusManager"/>

  <icon state="running" icon="jtmax_monitor"/>
  <icon state="xml" icon="jtmax_monitor"/>
  <icon state="failed" icon="jtmax_monitor"/>
</node>

```

- JTCの設定

```

<!--JTCの設定-->
<node j2ee-type="JTCCConnectionGroup"
      key="NodeTree.JeusMonitoring.JTCMonitoringService"
      page="JeusMonitoringJTCCMonitoringService_Overview" target="rightPane"
      static="true">
  <parent j2ee-type="Folder" jeus-type="JeusMonitoring"/>
  <delegator j2ee-type="JeusService" jeus-type="JeusManager"/>

  <icon state="running" icon="jtc_monitor"/>
  <icon state="xml" icon="jtc_monitor"/>
  <icon state="failed" icon="jtc_monitor"/>
</node>

```

catalog_ja.xmlの設定

JEUSがインストールされているディレクトリーの以下のパスに位置するcatalog_ja.xmlファイルを編集します。

```
webhome/admin_home/WEB-INF/classes/jeus/tool/webadmiin/resource/catalog
```

このファイルの<catalog>ノードの下に<text>ノードに以下のような内容を入れます。

- JTMAXの設定

```

<!--JTMAXの設定-->
<text key="jtmax.overview.title">JTmaxモニタリング</text>
<text key="NodeTree.JeusMonitoring.JTmaxMonitoringService">JTmaxモニタリング</text>
<text key="JeusMonitoringJTmaxMonitoringService_Overview">概要</text>

```

- JTCの設定

```
<!--JTCの設定-->
<text key="NodeTree.JeusMonitoring.JTCMonitoringService">JTCモニタリング</text>
<text key="JeusMonitoringJTCMonitoringService_Overview">概要</text>
<text key="JeusMonitoringJTCMonitoringService_Remote">Remote Domain</text>
<text key="jtc.overview.title">JTCモニタリング</text>
<text key="jtc.remote.title">JTC Remote Domainモニタリング</text>
```

- WEBTの設定

```
<!--WEBTの設定-->
<text key="NodeTree.JeusMonitoring.WebtMonitoringService">Webtモニタリング</text>
<text key="JeusMonitoringWebtMonitoringService_Overview">概要</text>
<text key="JeusMonitoringWebtMonitoringService_Group">Shared Connection
Group</text>
<text key="webt.overview.title">Webtモニタリング</text>
<text key="webt.group.title">Webt Shared Connection Groupモニタリング</text>
```

jeus.cssの設定

JEUSがインストールされているディレクトリーの以下のパスに位置するjeus.cssファイルを編集します。

```
webhome/admin_home/css
```

このファイルの最後に以下の内容を追加します。

```
.jtc_monitor{background: url("../image/mbean/jtc.gif") no-repeat
scroll;}.webt_monitor{background: url("../image/mbean/webt.gif") no-repeat
scroll;}.jtmax_monitor{background: url("../image/mbean/jt
max.gif") no-repeat scroll;}
```

5.3. WebTのモニタリング

本節ではJEUSコンソール・ツールとWebAdminを使用したWebTのモニタリング方法について説明します。

5.3.1. JEUSコンソール・ツールの使用

JEUSは、コンテナの状態を確認するjeusadminコマンドを提供します。このコマンドは新しく追加され、すべての情報を最も詳しく出力します。

管理ツールを使用するには、コンソール画面で以下のように実行します。すると、jeusadminのコマンド・プロンプト画面が表示されます。

```
jeusadmin container name -U user name -P user passwd
```

jeusadminツールの以下のコマンドを使ってWebTをモニタリングすることができます。

- webtadmin
- webtinfo
- webtinfo2
- webtdetailinfo
- webtshrink
- webtreconfig

webtadmin

webtadminはWebTの状態情報を照会するためのコマンドです。webtadminのモニタリングおよび接続情報に出力される環境設定項目とその値は、ユーザーがJEUSのJEUSMain.xmlやwebt.propertiesファイルに設定した項目および値とファイルに設定した項目および値と一致します。

webtadminの使用形式は以下のとおりです。

```
webtadmin [-C containerName] [-c connectionName] [-m] [-g groupName]
```

各オプションについての説明と実行結果は以下のとおりです。

- オプションなしで実行する場合

別途のオプションなしで実行すると、設定されているすべてのWebTの情報が出力されます。

- [-C containerName]

特定コンテナに属するWebT情報を出力します。

```
container name : <container Name>
```

- [-c connectionName]

特定接続の情報を出力します。connectionNameは接続IDに指定します。IDの一部文字列を変数にして照会することが可能です。

```
-----  
id           ip           port  pool  REQ  REP  PRE  COM  ROL  
-----  
[tmax2.100015] 192.168.33.84 6734 true 0    0    0    0    0
```

結果画面における照会内容の各項目についての説明は以下のとおりです。

項目	説明
pool	プール内にあるか否かを示します
REQ	リクエストを意味し、要求が送られた回数を示します
REP	Replyを意味し、要求に対する応答を受け取った回数を示します
PRE	トランザクションのPrepareを要求した回数を示します
COM	コミットが要求された回数を示します
ROL	ロールバックが要求された回数を示します

- [-m]

モニタリング情報、fdlファイル、ロギング情報を出力します。

```
Monitoring --
failback :      false
checkalive :    true
interval :      30

fdl file        null

Logging --
log dir :       D:\tmax
log file :      jtc.log
log buf size :  1024
log valid days  -1
log file format :MMddyyyy
log level :     debug
```

- [-g *groupName*]

特定コネクション・グループについての情報を出力します。

```
shared connection pool --
tmax2:
host address :  192.168.33.84:6734
backup address : null
user name :     null
user password : null
domain name     null
domain password null
init capacity   30
max capacity    30
increment rate  2
max idle time   360000
connect timeout 60
tp timeout      60
```

```

tx timeout          180
tx block timeout    60
headertype          3
size                30
support_xa          true
encryption          false
check               true
provision           true
autoclose           true
enable event        false
event flag          0
event Handler       null

```

```

clustered connection pool --
-----
tmaxc      tmax1 , tmax2
-----

```

webtinfo

webtinfoはWebtコネクション・グループの概略的な情報を照会するコマンドで、一定回数の分だけ繰り返し、状態をモニタリングできます。

使用形式は以下のとおりです。

```
webtinfo [node or container] [-i interval] [-k repeat_number] [webt_name]
```

以下は、各オプションについての説明です。

オプション	説明
[node or container]	指定した特定ノードの情報を出力します。このオプションなしで、webtinfo単独で使用する場合は、すべてのノードの情報を出力します。 [-i]、[-k]およびwebt_nameオプションと組み合わせて使用する場合は、コンテナの名前の入力により該当コンテナの情報を出力します
[-i interval]	webtinfoコマンドの実行周期を設定します(単位:秒)
[-k repeat_number]	webtinfoコマンドの実行回数を設定します。[-i interval]で設定した周期で設定値の分だけ繰り返して実行します
[webt_name]	特定コネクション・グループの情報を指定して出力します

webtinfoの実行結果は以下のように出力されます。

```

1 Container Name:Joonsoo-PC_container1
Container Name:Joonsoo-PC_container1

```

```

=====
ix      name                min    max    step  current  free  max_idle  waiting
-----
1       tmax2                 30    30    2     30       30   360000    0
=====

```

webtinfo2

webtinfo2はjeusadminのwebtinfoコマンドのような機能を行います。特定ノードを指定し、[-i]、[-k]オプションが使用できるコマンドです。

使用形式は以下のとおりです。

```
webtinfo2 [-n node] [-c container] [-i interval] [-k count] [-w webt]
```

以下は各オプションについての説明です。

オプション	説明
[-n ノード名]	照会対象として特定ノードを指定します。指定されていない場合、ローカル・ノードを照会します
[-c コンテナ名]	照会対象として特定コンテナを指定します。[-n]のように設定されていれば、[-c]を優先します。指定されていない場合は、照会対象ノードのすべてのコンテナを照会します
[-i インターバル]	インターバル時間を数値型で指定します。指定された時間の秒の分待機し、次の状態を出力します(デフォルト値:1)
[-k リピート数]	照会を繰り返す回数を数値型で指定します(デフォルト値:1)
[-w webt名]	照会対象として特定のwebtコネクション・グループを指定します。指定されていない場合には、照会範囲内のすべてのWebTを照会します

webtinfo2の実行結果は以下のように出力されます。

```

1 Container Name:Joonsoo-PC_container1
Container Name:Joonsoo-PC_container1
=====
ix      name                min    max    step  current  free  max_idle  waiting
-----
1       tmax2                 30    30    2     30       30   360000    0
=====

```

webtdetailinfo

webtdetailinfoは、WebTコネクション・グループに属するコネクションの詳細情報をモニタリングするコマンドです。一定の回数分だけ繰り返し、状態をモニタリングできます。

使用形式は以下のとおりです。

```
webtdetailinfo container_name webt_name [-i interval] [-k repeat_number]
```

以下は各オプションについての説明です。

オプション	説明
[-i interval]	webtdetailinfoコマンドの実行周期を設定します(単位: 秒)
[-k repeat_number]	webtdetailinfoコマンドの実行回数を設定します。[-i interval]で設定した周期で設定値の分だけ繰り返して実行します

webtdetailinfoの実行結果は以下のよう出力されます。

```
Container Name:Joonsoo-PC_container1 Webt Name:tmax2
=====
name          index      Connected Host      Connected Port
=====
[tmax2.100003] 1          192.168.33.84      6734
[tmax2.100004] 2          192.168.33.84      6734
```

webtshrink

webtshrinkはコネクションをmin値まで減らす機能を実行します。

```
webtshrink [-c container_name] [-w webt_name] [-all]
```

以下は、各オプションについての説明です。

オプション	説明
[-c container_name]	該当機能を適用するコンテナの名前を指定します
[-w webt_name]	該当機能を適用するWebTコネクション・グループの名前を指定します
[all]	すべてのWebTコネクション・グループに対して該当機能を実行します

webtshrinkの実行結果は以下のよう出力されます。

```
1 Container Name:Joonsoo-PC_container1
webtShrink Joonsoo-PC_container1 successfully
```

webtreconfig

webtreconfigはFailBackおよびコネクションをmin値まで減らす機能を実行します。

```
webtreconfig [-c container_name] [-w webt_name] [-all]
```

以下は、各オプションについての説明です。

オプション	説明
[-c <i>container_name</i>]	該当機能を適用するコンテナの名前を指定します
[-w <i>webt_name</i>]	該当機能を適用するWebTコネクション・グループの名前を指定します
[all]	すべてのWebTコネクション・グループに対して該当機能を実行します

webtreconfigの実行結果は以下のように出力されます。

```
1 Container Name:Joonsoo-PC_container1
webtreconfig Joonsoo-PC_container1 successfully
```

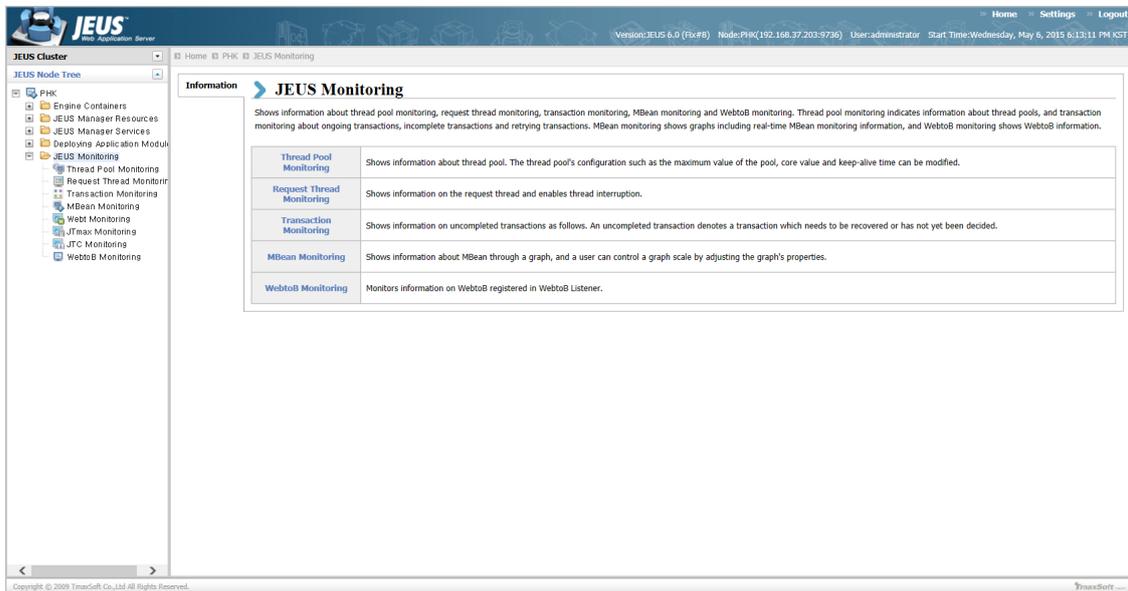
5.3.2. JEUS WebAdminの使用

JEUSはコンテナの状態を確認する方法としてWebブラウザを使ったWebAdminツールを提供します。JEUSを起動した後、Webブラウザに以下のようにアドレスを入力すれば、JEUS WebAdmin画面に移動します。

```
http://<ip address>:9744/webadmin
```

ユーザー名とパスワード名を入力してログインすれば、以下のような画面が表示されます。画面の左側の**JEUSノード・ツリー**の**[JEUS Monitoring]**配下にWebT、JTmax、JTCモニタリングのメニューがあります。

[図 5.1] JEUS WebAdmin画面



参考

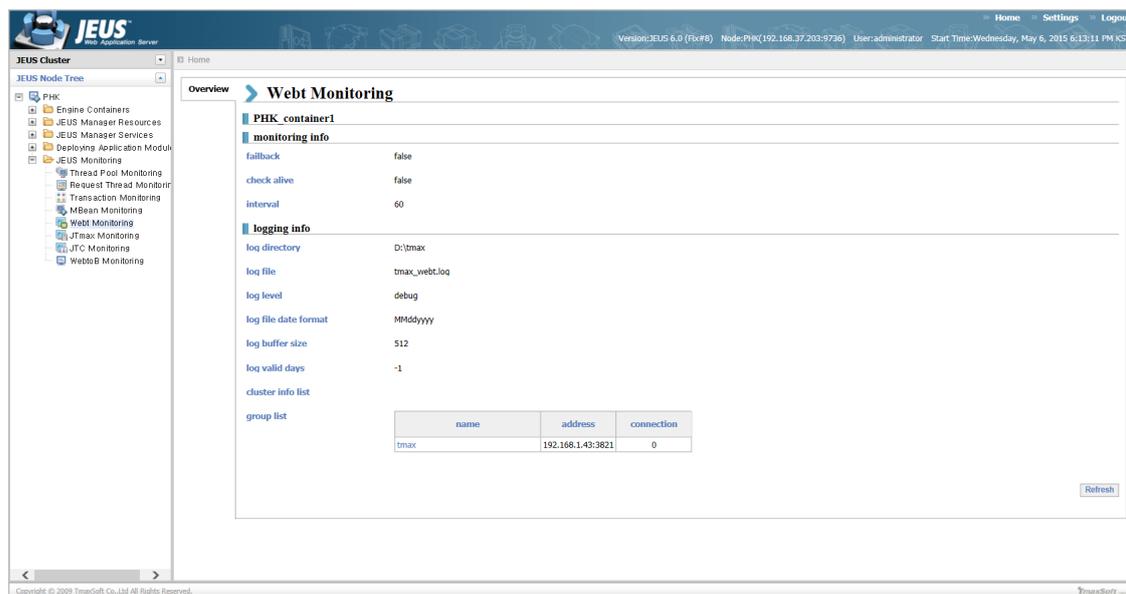
1. JEUS WebAdminの詳細については、JEUSガイドのうち『JEUS WebAdminガイド』を参照してください。

2. JTCモニタリングの詳細については、『Tmax JTCユーザガイド』を参照してください。

JEUS WebAdmin画面のJEUSノード・ツリーから[JEUS Monitoring] > [Webt Monitoring]を選択すれば、以下のようなWebt Monitoring画面が表示されます。

Webt Monitoring画面には、モニタリング情報とロギング情報、そしてコネクション・グループについての簡単な情報が出力されます。

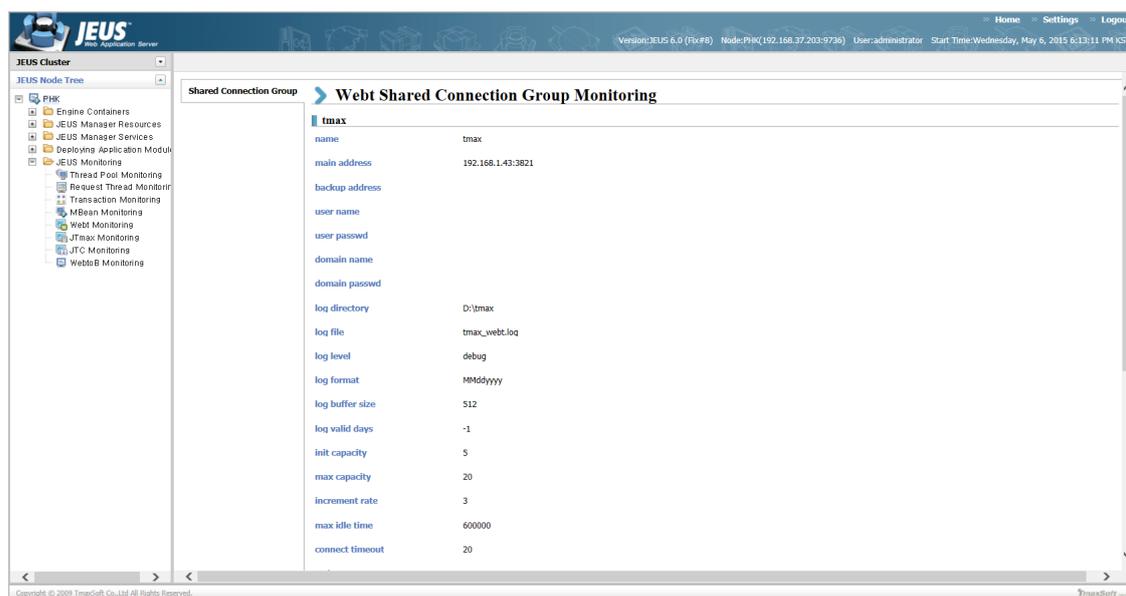
[図 5.2] Webtモニタリングのメイン画面



Webt Monitoring画面の「logging info」領域から「group list」に照会されているグループ名をクリックすれば、当該コネクション・グループの設定環境およびコネクション・グループ内のコネクションの運用情報を確認することができます。

以下は、特定のグループ名をクリックして表示される詳細画面です。この画面に照会される環境設定項目とその値は、ユーザーがJEUSのJEUSMain.xml、またはwebt.propertiesファイルに設定した項目および値と一致します。

【図 5.3】 Webtモニタリングのコネクション・グループ情報の画面



5.4. JTmaxのモニタリング

本節ではJEUSコンソール・ツールとWebAdminを使用したJTmaxのモニタリング方法について説明します。

5.4.1. JEUSコンソール・ツールの使用

JEUSコンソール・ツールを使用してJTmaxをモニタリングするには、以下のように実行してjeusadminに接続します。

```
jeusadmin container name -U user name -P user passwd
```

jeusadminツールのjtmaxadminコマンドを使用してJTmaxをモニタリングすることができます。

jtmaxadmin

jtmaxadminはJTmaxの状態情報を照会するためのコマンドです。jtmaxadminで出力されるモニタリング、ロギングおよび一般情報の環境設定項目とその値は、ユーザーがJEUSのJEUSMain.xmlファイルに設定した項目および値と一致します。

jtmaxadminは別途のオプションがなく、実行結果は以下のとおりです。

```
NodeName> jtmaxadmin
container name : node_container1
Listen port :

jtmax listen port :      0
min thread :            2
```

max thread : 4
increment : 2
period : 600000

Logging :

log directory : d:\
log file : jtmax.log
log buffer size : 512
log file format : MMddyyyy
log level : debug
log valid days : 1

General Infomation :

fdl file : c:\\tmax\\tmax.fdl
default character set : null

EJB Infomation :

service name	export name	method name	call count
JTOUPPER4	echotest	ToUpper	0
TPCALLCARRAY_J	tpcall	tpcallCarray	0
JTOUPPER3	echotest	ToUpper	0
JTOUPPER2	echotest	ToUpper	0
ECHOCHAR_J	echo	echoChar	0
JTOUPPER1	echotest	ToUpper	0
ECHOSHORT_J	echo	echoShort	0
TPCALLFIELD_J	tpcall	tpcallField	0
FDLINSERT_J	tpcall	tpcallFdlInsert	0
ECHOLONG_J	echo	echoLong	0
GSVC03	echotest	setStringEcho2	0
ECHOFLOAT_J	echo	echoFloat	0
GSVC02	echotest	setIntEcho	0
ECHOINT_J	echo	echoInt	0
GSVC01	echotest	setStringEcho	0
INSERT_J	echo	insert	0
TPCALLSTRING_J	tpcall	tpcallString	0
ECHODOUBLE_J	echo	echoDouble	0
ECHOSTRING_J	echo	echoString	0

Condition :

request : 0
reply : 0
prepare : 0

```
commit :      0
rollback :    0
```

5.4.2. JEUS WebAdminの使用

JEUS WebAdminを使用してJTmaxをモニタリングするには、Webブラウザに以下のようにアドレスを入力します。

```
http://<ip address>:9744/webadmin
```

JEUS WebAdmin画面に移動し、ユーザー名とパスワードを入力してログインします。JEUS WebAdminの接続、ログイン、およびメイン画面についてはすでに「WebTのモニタリング」の部分で説明したので、本節では省略します。詳細については、「[5.3.2. JEUS WebAdminの使用](#)」を参照します。

参考

1. JEUS WebAdminの詳細については、『JEUS WebAdminガイド』を参照してください。
 2. JTCモニタリングの詳細については、『Tmax JTCユーザガイド』を参照してください。
-

WebAdmin画面の**JEUSノード・ツリー**から**[JEUS Monitoring] > [JTmax Monitoring]**を選択すれば、以下のような**JTmax Monitoring**画面が表示されます。この画面からはJTmaxのモニタリング情報および運用情報を確認することができます。JTmax Monitoringに出力される環境設定項目とその値は、ユーザーがJEUSのJEUSMain.xmlファイルに設定した項目および値と一致します。

【図 5.4】 JTmax Monitoringのメイン画面

Overview > JTmax Monitoring

PHK_container1

listening info

listen port	8181
min thread	2
max thread	10
increment	2
period	600000

logging info

file name	jtmax.log
level	debug
file format	MMddyyyy
buffer size	1024
valid days	1

general info

fdl file	d:\tmax\tmax.fdl
default character set	EUC-KR

ejb list

tmax service name	ejb service name	request
ECHOSHORT_J	echoShort	0
ECHOCHAR_J	echoChar	0
INSERT_J	insert	0
TPCALLFDL_J	tpcallField	0
ECHODOUBLE_J	echoDouble	0
ECHOINT_J	echoInt	0
ECHOLONG_J	echoLong	0
FDLINSERT_J	tpcallFdInsert	0
ECHOSTRING_J	echoString	0
TPCALLSTRING_J	tpcallString	0
TPCALLCARRAY_J	tpcallCarray	0
ECHOFLOAT_J	echoFloat	0

condition info

current thread	0
running thread	0

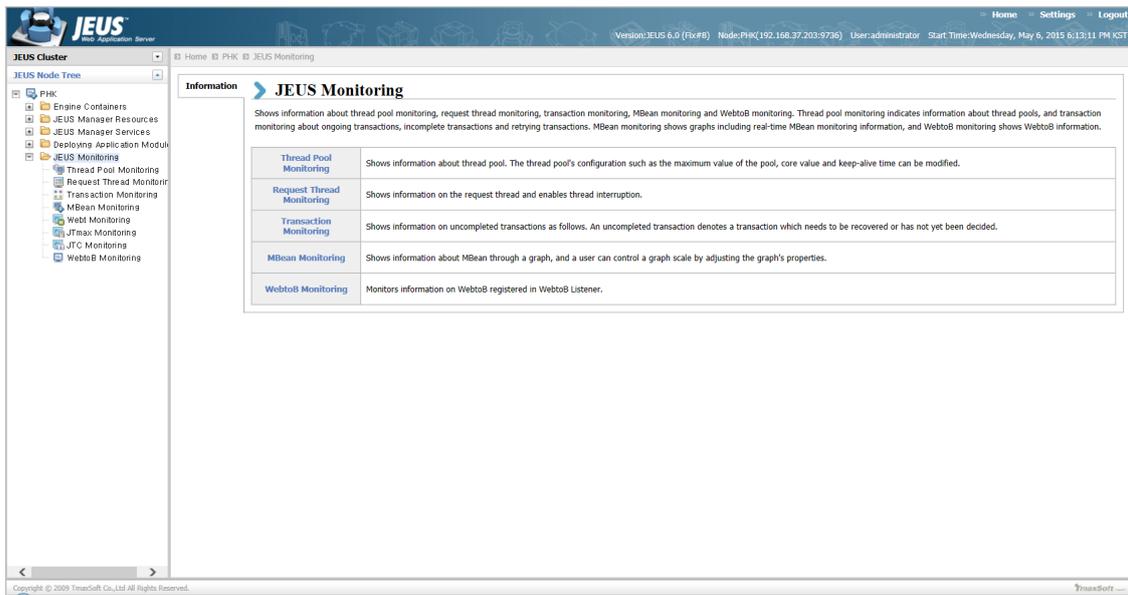
5.5. JEUS WebAdminツール

JEUSはコンテナの状態を確認するための方法で、WebブラウザによるWebAdminツールを提供します。一般的に、JEUSを起動した後、Webブラウザに以下のアドレスを入力するとWebAdmin画面が現れます。

```
http://<ip address>:9744/webadmin
```

ユーザー名とパスワードを入力してログインすると、以下のような画面が現れます。画面左側の**JEUSノード・ツリー**のJEUSモニタリング内にWebtモニタリング、JTmaxモニタリング、JTCモニタリングというメニューがあります。

【図 5.5】 WebAdmin画面



参考

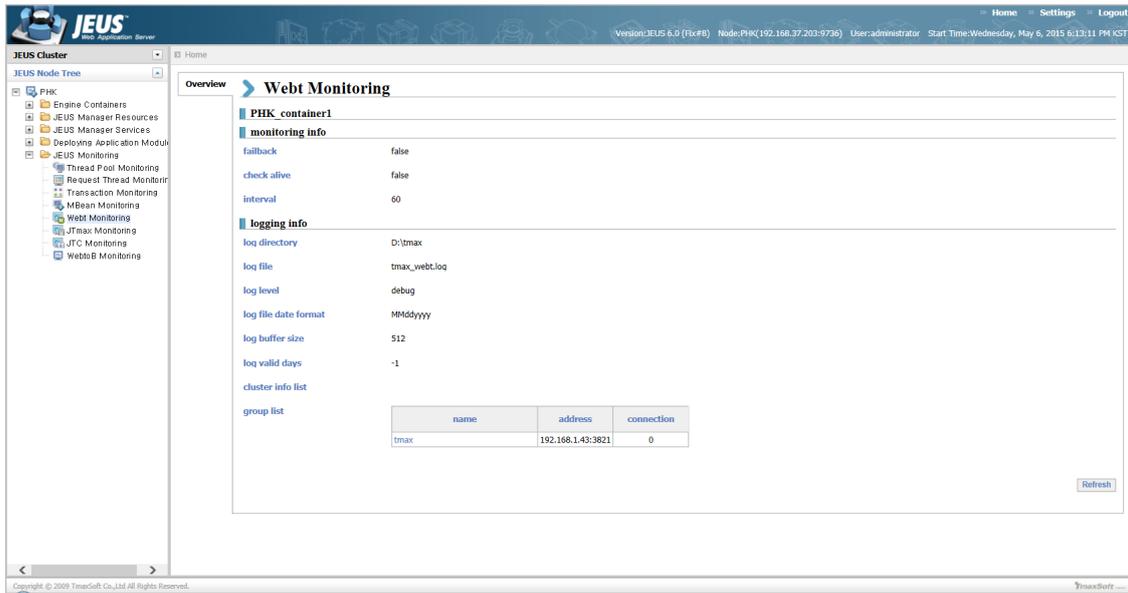
WebAdminについての詳細は『JEUS WebAdminガイド』を参照し、JTCモニタリングについては『Tmax JTCユーザガイド』を参照してください。

5.5.1. WebTのモニタリング

WebAdmin画面のJEUSノード・ツリーで「Webt Monitoring」をクリックすると、以下のようなWebt Monitoring画面が現れます。

Webt Monitoring画面には、モニタリング情報とロギング情報、そしてコネクショングループの簡略情報が出力されます。

【図 5.6】 Webt Monitoringのメイン画面

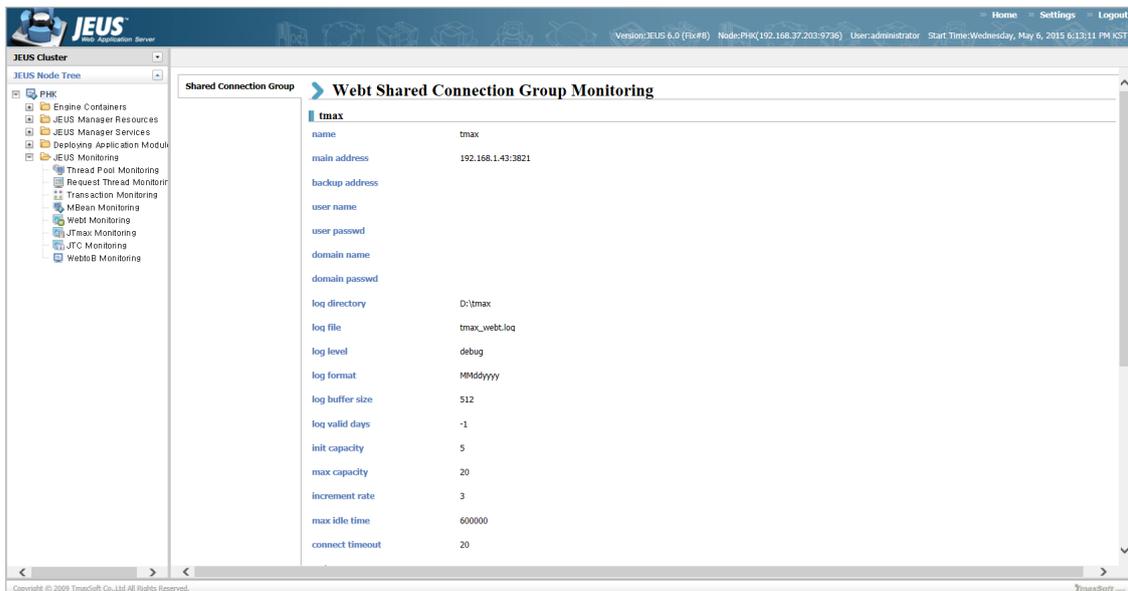


上の画面で「group list」のグループ名をクリックすると、該当コネクション・グループの詳細情報を確認できます。

この画面では、各コネクション・グループの設定された環境および属するコネクションの運用情報を確認できます。Webtモニタリングで出力される環境設定項目とその値は、ユーザーがJEUSのJEUSMain.xmlやwebt.propertiesファイルに設定した項目および値と一致します。

以下は、Webtモニタリングの特定コネクション・グループの詳細画面です。

【図 5.7】 Webt Monitoringのコネクション・グループ情報画面



5.5.2. JTmaxのモニタリング

WebAdmin画面のJEUSノード・ツリーで「JTmaxモニタリング」をクリックすると、以下のようなJTmax Monitoring画面が現れます。

この画面では、JTmaxのモニタリング情報および運用情報をすべて確認できます。JTmax Monitoring画面に出力される環境設定項目とその値は、ユーザーがJEUSのJEUSMain.xmlファイルに設定した項目および値と一致します。

[図 5.8] JTmax Monitoringのメイン画面

The screenshot shows the JEUS Web Administration interface. The main content area is titled "JTmax Monitoring" and is divided into several sections:

- PHK_container1**
 - listening info**
 - listen port: 8181
 - min thread: 2
 - max thread: 10
 - increment: 2
 - period: 600000
 - logging info**
 - file name: jtmax.log
 - level: debug
 - file format: MMddyyyy
 - buffer size: 1024
 - valid days: 1
 - general info**
 - fdl file: d:/tmax/tmax.fdl
 - default character set: EUC-KR
 - ejb list**

tmax service name	ejb service name	request
ECHOSHORT_J	echoShort	0
ECHOCHAR_J	echoChar	0
INSERT_J	insert	0
TPCALLFDL_J	tpcallField	0
ECHODOUBLE_J	echoDouble	0
ECHOINT_J	echoInt	0
ECHOLONG_J	echoLong	0
FDLINSERT_J	tpcallFdlInsert	0
ECHOSTRING_J	echoString	0
TPCALLSTRING_J	tpcallString	0
TPCALLCARRAY_J	tpcallCarray	0
ECHOFLOAT_J	echoFloat	0
 - condition info**
 - current thread: 0
 - running thread: 0

付録 A. webterrorcodeメッセージ

本付録では、webterrorcodeで発生するすべてのメッセージ(1000~9999)について説明します。

A.1. webterrorcodeメッセージ(1000~)

1000 failed to connect to main server {hostaddress:port}

説明	メインサーバーに設定されているTmax接続に失敗しました
対応方法	サーバー状態を確認し、Tmaxを起動します。または、サーバーアドレスを確認し、アドレスを変更します

1001 connected to JeusGW

説明	JEUSゲートウェイに接続します
----	------------------

1002 ensure webt license

説明	Tmaxへのソケット接続は成功しましたが、TPSTARTに失敗しました
対応方法	該当サーバーのライセンスを確認します。または、該当アドレスがTmaxであるかを確認します

1003 tmax session established.. tmax version [{接続したTmaxバージョン}]

説明	Tmax接続に成功しました
----	---------------

1004 failed to subscribe entry [{イベント名}]

説明	tpstartする際にイベント名のサブスクリプションに失敗しました
----	-----------------------------------

対応方法	イベント名が正しいかを確認します
------	------------------

1005 node count [{ノード数}]

説明	接続したTmaxのノード数が出力されます
----	----------------------

1006 rq count [{RQの数}]

説明	接続したTmaxに設定されたRQの数が出力されます
----	---------------------------

1007 topend product count [{topen productの数}]

説明	接続したTmaxのtopen productの数が出力されます
----	---------------------------------

1008 tpstart data [{0}]

説明	現在は使用していません
----	-------------

1009 rq registered [{index}:{RQ名}]

説明	Tpstartする場合にRQ情報をクライアントに格納します
----	-------------------------------

1010 topend product registered [{index}:{topen product名}]

説明	Tpstartする場合にtopen product情報をクライアントに格納します
----	--

1011 compress threshold {メッセージ・サイズ} byte

説明	Tmaxで{メッセージ・サイズ}バイト以上のメッセージは圧縮するように設定します
----	--

1012 failed to initialize tmax session

説明	圧縮情報、ノード情報、RQ情報の設定に失敗しました
対応方法	以前のメッセージを参照し、正しくない数値がないかを確認します

1013 connection is already destroyed

説明	現在は使用していません
----	-------------

1014 connection destroyed

説明	接続情報を削除します
----	------------

1015 failed to end tmax session

説明	tpendの呼び出しに失敗しました
対応方法	接続がすでに切断されている場合、処理する必要はありません

1016 tpcall failed

説明	tpcallする際に失敗した応答を受け取りました
対応方法	Tmaxのサーバーから転送された応答を確認します

1017 tpgetrply failed

説明	tpgetrplyする際に失敗した応答を受け取りました
対応方法	Tmaxのサーバーから転送された応答を確認します

1018 tereceive failed

説明	teReceiveする際に失敗した応答を受け取りました
対応方法	Tmaxのサーバーから転送された応答を確認します

1019 invalid event name {unsubscribeするイベント名}

説明	tpunsubscribeを呼び出す際に{アンサブスクライブするイベント名}の名前は存在しない名前です
対応方法	イベント名が正しいかを確認します

1020 tpconnect failed

説明	tpconnectする際に失敗した応答を受け取りました
対応方法	Tmaxのサーバーから転送された応答を確認します

1021 tpsend failed.. currently receive only

説明	会話型通信のうち、現在recvのみ行うことができる状態でtpsendしました
対応方法	クライアントのsend/recv状態を確認し、それに合わせて動作を修正します

1022 tprecv failed.. currently send only

説明	会話型通信のうち、現在sendのみ行うことができる状態でtprecvしました
対応方法	クライアントのsend/recv状態を確認し、それに合わせて動作を修正します

1023 tpenq failed

説明	tpenqする際に失敗した応答を受け取りました
対応方法	Tmaxのサーバーから転送された応答を確認します

1024 invalid queue name [{ユーザーが入力したRQ名}>::{管理しているRQ名}

説明	tpenq、tpdeqする際にユーザーが入力したRQ名が正しくありません
対応方法	ユーザーが入力したRQ名を、管理されているRQ名に属するようにします

1025 tpenq with !TPRQS flag not issued : queueName [{要求したRQ名}]

説明	tpenqしていないのにtpdeqしようとしています
対応方法	tpenqがあった要求に対してtpdeqしているのかを確認します

1026 tpdeq failed

説明	tpdeqする際に失敗した応答を受け取りました
対応方法	Tmaxのサーバーから転送された応答を確認します

1027 tpqstat failed

説明	tpqstatする際に失敗した応答を受け取りました
対応方法	Tmaxのサーバーから転送された応答を確認します

1028 tpqsvcstat failed

説明	tpqsvcstatする際に失敗した応答を受け取りました
対応方法	Tmaxのサーバーから転送された応答を確認します

1029 not supported service type {サービスタイプ} in xa

説明	該当するタイプの呼び出しはxaを適用できません
対応方法	サービスタイプがisxaであるかを確認します

1031 security handshaking start

説明	暗号化の設定を開始します
----	--------------

1032 security handshaking finished

説明	暗号化の設定が正常に終了しました
----	------------------

1033 security handshaking failed

説明	暗号化の設定に失敗しました
対応方法	以前に残ったログを参照し、設定が失敗した原因を把握します

1034 receive start for {応答待機時間} msec

説明	応答待機時間の分だけTmaxからの応答を待機します(単位: ミリ秒)
----	------------------------------------

1035 buffer received {応答を受け取ったメッセージの情報}

説明	Tmaxから{応答を受け取ったメッセージの情報}の応答を受信しました
----	------------------------------------

1036 receive failed

説明	Tmaxからの受信に失敗しました
対応方法	例外からのエラー原因を把握します。ソケット接続状態を確認します

1037 buffer transmitted {送信したメッセージ}

説明	Tmaxと接続したソケットに{送信したメッセージ}のメッセージを書き込むのに成功しました
----	--

1038 transmit failed

説明	Tmaxと接続したソケットに書き込むのに失敗しました
対応方法	例外からのエラー原因を把握します。ソケット接続状態を確認します

1039 enqueue buffer {格納するメッセージ}

説明	webtで管理する内部的なキューで、受信したメッセージを格納します
----	-----------------------------------

1040 transceiver {0} started

説明	現在は使用していません
----	-------------

1041 matching received.. seqno({予想するseqno}:{受信したseqno}), type({送信したメッセージタイプ}:{受信したメッセージタイプ})

説明	受信したメッセージ情報が出力されます
----	--------------------

1042 matching result.. seqno match = {seqnoが一致するのかを確認}, type match = {メッセージタイプが一致するのかを確認}

説明	受信したメッセージ情報を通じてサービスの成功有無およびタイプなどを判断して出力します
----	--

1043 message discarded {受信したメッセージ}

説明	受信したメッセージ {受信したメッセージ}を削除しました
対応方法	受信したメッセージを確認します

1050 ping start

説明	アプリケーションpingを開始します
----	--------------------

1051 ping failed

説明	pingに失敗しました
対応方法	サーバーへの接続状態を確認します

1052 ping success

説明	pingに成功しました
----	-------------

1053 start cleanup of transceiver

説明	通信情報の初期化を開始します
----	----------------

1054 end cleanup of transceiver

説明	通信情報の初期化を終了します
----	----------------

1055 internal shutdown of dialogue session index {会話型通信ID}

説明	初期化中に通信情報を初期化します
----	------------------

1056 node registered [{index}:{ノードのInid}]

説明	tpstart中にノード情報を格納します
----	----------------------

1057 invalid Inid [{ユーザーが入力したノード名}::{管理しているノード名一覧}]

説明	tpgetsvglisを呼び出す際に該当するノード名が存在しません
対応方法	ユーザーが入力したノード名を管理しているノード名一覧の中に入れます

1058 invalid event mask {0}

説明	現在は使用していません
----	-------------

1059 tpacall failed

説明	tpacallのTPBLOCKの場合に失敗応答を受信します
対応方法	1. WebtServiceFailの場合は、サービスがfailを与える動作を確認します 2. それ以外の場合は、tpacallの失敗理由であるtperrorを確認します

1080 failed to establish session with remote domain

説明	Tuxedoリモート・ドメイン接続に失敗しました
対応方法	ログレベルをdebugにし、Tuxedoからの応答メッセージを確認します

1081 session established

説明	Tuxedoリモート・ドメイン接続に成功しました
----	--------------------------

1082 send failure for xa protocol {転送したメッセージのタイプコード}

説明	TuxedoにXAプロトコル転送に失敗しました
対応方法	タイムアウトなどによりメッセージを転送できないのかを確認し、例外を確認します

1083 recv failure for xa protocol {受信しなければならないメッセージのタイプコード}:{受信したメッセージのタイプコード}

説明	TuxedoでXAプロトコル転送の失敗に対する応答を受信します
対応方法	受信したメッセージのタイプコードを参照し、Tuxedoで転送するメッセージを確認します

1090 cleanup transceiver

説明	Tuxedo readerスレッドを解除します
----	-------------------------

1091 transceiver failed. try reconnection.

説明	Tuxedo readerスレッドの開始に失敗しました
対応方法	前に記録された例外を確認します

1092 transceiver timedout.

説明	Tuxedo readerスレッドを開始する際にタイムアウトが発生しました
対応方法	接続状態を確認します

1093 transceiver run start

説明	Tuxedo readerスレッドを開始します
----	-------------------------

1094 transceiver stop. because sockets[**{socket情報}**]

説明	Tuxedo readerスレッドを終了します
----	-------------------------

1095 transceiver wait state. because no read socket

説明	readerスレッドでROLLING_DOWNが設定されている場合、新しく接続されるまで待機します
----	---

1096 transceiver wait out.

説明	待機状態が終了します
----	------------

1097 transceiver select start.

説明	Selectを開始します
----	--------------

1098 transceiver selector stop.

説明	Selectを終了します
----	--------------

1099 TM_DOWNWARN_REQUEST recive {メッセージ情報}

説明	Tmaxでローリング・ダウンされたというメッセージが受信されます
----	----------------------------------

1100 failed to close socket

説明	ソケットのクローズに失敗しました
対応方法	以前のログを確認し、原因を把握します

1101 failed to connection to server {host:port}

説明	サーバーの接続に失敗しました
対応方法	接続状態を確認します

1102 connection established {socket, pipe}:{host:port}

説明	Tmaxに接続されました
----	--------------

1103 failed to get socket stream

説明	現在は使用していません
----	-------------

1104 failed to set socket timeout to {setsocket設定値} msec

説明	ソケットのsetSoTimeout設定に失敗しました
対応方法	ソケットのsetSoTimeoutが発生した原因を把握します。以前の例外を参照します

1105 socket connection timeout

説明	Tmaxの接続時にタイムアウトが過ぎるまで接続に成功しませんでした
対応方法	サーバーへの接続状態を確認します

1106 socket connection interrupted

説明	現在は使用していません
----	-------------

1107 failed to read from tmax

説明	ソケットでメッセージを読み込む際にエラーが発生しました
対応方法	接続状態を確認します

1108 failed to write to tmax

説明	ソケットにメッセージを書き込む際にエラーが発生しました
対応方法	接続状態を確認します

1109 start to connect server {host:port} for {接続実行時間} msec

説明	Tmaxに接続を行います
----	--------------

1110 connection interrupted {host}:{port}

説明	接続を行う際にInterruptedIOExceptionが発生しました
対応方法	ソケットに接続する際、InterruptedIOExceptionの原因を把握します。以前の例外を参照します

1111 socket list add {ソケット情報}

説明	ソケットに接続します
----	------------

1150 TM_CSSVR_NOREADY recive {メッセージ情報}

説明	Tmaxでローリング・ダウンされて、バックアップTmaxに接続するよにというメッセージを受信します
----	---

1200 put acall reply start {cd値}

説明	acallに対する応答を受信します
----	-------------------

1201 not registered call desc {cd値}

説明	acallに対する応答を受信しましたが、すでに存在しなくなったcd値です
対応方法	前のtpacallが失敗した原因を確認し、それまでtpacall要求が有効であるように修正します

1202 put acall reply end {cd値}

説明	acallに対する応答が正常に処理されました
----	------------------------

1203 get acall reply start {cd値}

説明	tpgetrplyを開始します
----	-----------------

1204 get acall reply end {cd値}

説明	tpgetrplyに成功しました
----	------------------

1205 max acall reply {現在の要求数} reached

説明	tpacallを実行した場合、tpgetrplyを行っていない応答がmaxcallの分だけ発生します
----	--

1300 dialogue session terminated.. event = {現在のイベント}

説明	会話型通信中に通信ができなくなりました
対応方法	会話型サービスがすでに終了しているのかを確認します

1301 put dialogue reply.. current size = {現在入力されているメッセージ数}

説明	会話型通信中に内部的なキューで受信したメッセージが入力されます
----	---------------------------------

1302 start waiting dialoge reply for {メッセージの受信を待機する時間} msec

説明	tprecvを開始します
----	--------------

1303 dialogue event occurred.. event = {last event}

説明	会話型通信中にエラー応答が到着しました
対応方法	会話型サービスがサーバー側で終了しているのかを確認します

1304 invalid event message {メッセージ情報}

説明	会話型通信中に不明のメッセージが到着しました
対応方法	サーバー側で転送したメッセージ内容を確認します

1305 set maximum dialoge session size {サイズ}

説明	対話型通信セッションの最大サイズを設定します
----	------------------------

1306 max dialogue session reached.. max = {設定された数}

説明	tpconnectを行う場合、最大セッション数でセッションが作成されます
----	--------------------------------------

1307 new dialogue session is registered.. id = {セッションID}

説明	tpconnectを行う場合、新しい会話型セッションが登録されます
----	-----------------------------------

1308 dialogue session {セッションID} is unregistered

説明	会話型通信セッションが解除されました
----	--------------------

1309 invalid dialogue session {セッションID}

説明	会話型通信中の応答が不明なセッションです
対応方法	tpconnectが正常な会話型サービスであるのかを確認します

1310 dialogue session closed by peer message {メッセージ情報}

説明	会話型通信中に応答を受けましたが解除されたセッションです
対応方法	tpdisconなどで解除されたセッションであるのかを確認します

1311 connection is closing.. discard message {メッセージ情報}

説明	connectionが終了して、内部的なキューに存在していたメッセージを削除します
----	---

1400 put event msg start

説明	イベント・メッセージ・キューに格納を開始します
----	-------------------------

1401 purged oldest element {メッセージ情報}

説明	一番古いイベント・メッセージを削除します
----	----------------------

1402 put event end [{メッセージ・サイズ}/{最大メッセージ・サイズ}]

説明	イベント・メッセージ・キューへの格納を終了します
----	--------------------------

1403 get event msg start

説明	イベント・メッセージの取得を開始します
----	---------------------

1404 get event end [{メッセージの数}/{メッセージ・キューの最大サイズ}]

説明	イベント・メッセージの取得を終了します
----	---------------------

1500 put rq msg {メッセージ情報}

説明	RQを使用する際、応答を内部的なキューに格納します
----	---------------------------

1501 get rq msg start for {待機時間} msec

説明	RQを使用する際、応答を待機します
----	-------------------

1502 get rq msg end {メッセージ情報}

説明	RQの応答を受信します
----	-------------

1600 invalid rq index {index}.. discard {メッセージ情報}

説明	RQの応答が正しくないメッセージであるため削除します
対応方法	該当メッセージ情報を確認し、RQでどのようなメッセージを送信したのかを確認します

1601 invalid rq index {0}

説明	現在は使用していません
----	-------------

1700 put sync msg {メッセージ情報}

説明	通信中に応答を内部的なキューに格納します
----	----------------------

1701 get sync msg for {待機時間} msec

説明	通信中に応答を待機します
----	--------------

1800 put te msg {メッセージ情報}

説明	topenメッセージをキューに入力します
----	----------------------

1801 get te msg for index {0} for {1} msec

説明	現在は使用していません
----	-------------

1802 get te msg end {メッセージ情報}

説明	topenメッセージの取得を終了します
----	---------------------

1900 message(type:{0}) from remote domain for recover.

説明	現在は使用していません
----	-------------

1901 not expected message.. {メッセージ・キューの値}. discard.

説明	Tuxedoアウトバウンド要求で正しくないメッセージが受信されました
対応方法	Tuxedoで応答したメッセージ内容を確認します

1902 enqueue message {0}

説明	現在は使用していません
----	-------------

1903 handle inbound request {Tuxedoメッセージ情報}

説明	Tuxedoインバウンド要求を処理します
----	----------------------

1904 handle outbound request of key {メッセージ・キーの値}

説明	Tuxedoアウトバウンド要求を処理します
----	-----------------------

1905 register request key {要求キーの値}

説明	Tuxedo要求キーを登録します
----	------------------

1950 failed in service {サービス名}

説明	インバウンド・サービスに失敗しました
対応方法	該当インバウンド・サービスのfail応答の理由を確認します

1951 error in service {サービス名}

説明	インバウンド・サービスのエラーです
対応方法	該当インバウンド・サービスのエラー原因を確認します

1952 failed to destory service instance {サービス名}

説明	サービス・インスタンスの解除に失敗しました
対応方法	発生したexceptionを確認します

A.2. webterrorcodeメッセージ(2000~)

2000 create managed connection

説明	コネクション・プールでコネクションの作成を開始します
----	----------------------------

2001 create managed connection for {待機時間} msec

説明	コネクション・プールでコネクションの作成を開始します
----	----------------------------

2002 create xa connection

説明	コネクション・プールでコネクションの作成を開始します
----	----------------------------

2003 create xa connection for {待機時間} msec

説明	コネクション・プールでコネクションの作成を開始します
----	----------------------------

2100 shared connection pool {プール名} is initiated

説明	コネクション・プールを初期化します
----	-------------------

2101 get connection handle for {待機時間} msec by {Thread}

説明	コネクション・プールでgetConnectionを開始します
----	--------------------------------

2102 peek connection handle for {待機時間} msec

説明	コネクション・プールでgetConnectionを開始します
----	--------------------------------

2103 peek connection handle

説明	コネクション・プールでgetConnectionを開始します
----	--------------------------------

2104 return connection handle of managed connection {コネクション情報} to {Thread}

説明	コネクション・プールからコネクションを返します
----	-------------------------

2105 returning managed connection {コネクション情報} start

説明	ユーザーがコネクション・プールにコネクションの返却を開始します
----	---------------------------------

2106 managed connection {コネクション情報} is not pooled connection

説明	ユーザーが返却したコネクションはプールに存在していたコネクションではありません
----	---

2107 pool name {返却したプールの名前} mismatches with managed connection {コネクション情報}

説明	ユーザーが返却したコネクションはプールに存在していたコネクションではありません
----	---

2108 returning managed connection {コネクション情報} end

説明	コネクション・プールへの返却を終了します
----	----------------------

2109 remove managed connection {コネクション情報} from pool

説明	プールからコネクションを削除します
----	-------------------

2110 destory managed connection pool

説明	コネクション・プールのコネクションを初期化します
----	--------------------------

2111 check connection pool {checkalive可能の可否} : {failback可能の可否}

説明	コネクション・プールのモニタリングを開始します
----	-------------------------

2112 managed connection {コネクション情報} is created in pool

説明	コネクション・プールにコネクションが作成され、追加します
----	------------------------------

2113 prepare managed connection {コネクション情報} for use

説明	コネクション・プールでgetConnectionを行う際にコネクションを初期化します
----	--

2114 prepare managed connection {コネクション情報} for pooling

説明	コネクション・プールに返却する際にコネクションを初期化します
----	--------------------------------

2115 unregister managed connection {コネクション情報} from pool

説明	コネクション登録を解除します
----	----------------

2200 shared connection pool {プール名} is initiated

説明	クラスタ型のコネクション・プールを作成します
----	------------------------

2201 get connection handle for {待機時間} msec

説明	クラスタ・コネクション・プールのgetConnectionを開始します
----	-------------------------------------

2202 peek connection handle for {待機時間} msec

説明	クラスタ・コネクション・プールのgetConnectionを開始します
----	-------------------------------------

2203 peek connection handle

説明	クラスタ・コネクション・プールのgetConnectionを開始します
----	-------------------------------------

2500 internal connection pool {プール名} is initialized

説明	コネクション・プールを初期化します
----	-------------------

2501 create connection direct for {待機時間} msec

説明	現在のスレッドでコネクションを作成します
----	----------------------

2502 create connection delegated for {待機時間} msec

説明	他のスレッドでコネクションを作成します
----	---------------------

2503 set executor {ジェネレータ情報}

説明	スレッド・ジェネレータを設定します
----	-------------------

2504 put connection {コネクション情報}

説明	プールに返却します
----	-----------

2505 remove connection {コネクション情報} from pool

説明	コネクションをプールから削除します
----	-------------------

2506 destroy connection {コネクション情報}

説明	コネクションをプールから削除します
----	-------------------

2507 get connection from pool

説明	現在は使用していません
----	-------------

2508 get connection from pool for {待機時間} msec

説明	コネクション・プールでgetConnectionを開始します
----	--------------------------------

2509 interrupted by administrator

説明	InterruptedExceptionによってgetConnectionに失敗しました
----	--

2510 peek connection from pool for {待機時間} msec

説明	コネクション・プールでgetConnectionを開始します
----	--------------------------------

2511 peek connection from pool

説明	コネクション・プールでgetConnectionを開始します
----	--------------------------------

2512 check connection pool {checkalive可能の可否}:{failback可能の可否}

説明	コネクション・プールのモニタリングを開始します
----	-------------------------

2513 check connection pool by administrator {checkalive可能の可否}:{failback可能の可否}

説明	コネクション・プールのモニタリングを開始します
----	-------------------------

2514 managed connection {コネクション情報} is invalidated by failback

説明	フェイルバックが設定されており、バックアップに接続されてコネクションを削除します
----	--

2515 managed connection {コネクション情報} is invalidated by idle timeout

説明	コネクション数がminより大きい場合で、該当コネクションがアイドルタイムより長時間使用されないため削除します
----	--

2516 managed connection {コネクション情報} is already closed by peer

説明	コネクション接続が終了して削除します
----	--------------------

2517 managed connection {0} is invalidated by webtadmin command

説明	現在は使用していません
----	-------------

A.3. webterrorcodeメッセージ(3000~)

3000 webt event handle started.. message listener : {ハンドラ情報}

説明	イベント処理スレッドを開始します
----	------------------

3001 event work {0} is started

説明	現在は使用していません
----	-------------

3002 event handling started for message {メッセージ情報}

説明	イベント処理を開始します
----	--------------

3003 event handling ended for message {メッセージ情報}

説明	イベント処理を終了します
----	--------------

3004 error handling started

説明	例外処理を開始します
----	------------

3005 error handling ended

説明	例外処理を終了します
----	------------

3006 event work {イベント処理スレッド名} is terminated

説明	イベント処理スレッドを終了します
----	------------------

3007 exception thrown from event handler.. ignored

説明	イベント処理スレッドで例外が発生しましたが無視されました
対応方法	例外の内容を確認します

A.4. webterrorcodeメッセージ(4000~)

4000 invalid connection handle {コネクション情報}

説明	WebtRemoteServiceを作成する際パラメータに移動したコネクションが正しいコネクションではありません
----	--

4001 invalid service name {サービス名}

説明	サービスを設定する際にNULLがパラメータに設定されました
----	-------------------------------

4002 tpcall start with attribute {attribute情報}

説明	tpcallを開始します
----	--------------

4003 illegal buffer type {送信したメッセージタイプ}

説明	tpcallをする際にTPNOCHANGEに設定されましたが、応答を受信したメッセージタイプが異なります
----	--

4004 tpcall end

説明	tpcallを終了します
----	--------------

4005 tpacall start with attribute {attribute情報}

説明	tpacallを開始します
----	---------------

4006 tpacall end

説明	tpacallを終了します
----	---------------

4007 tpgetrply start with attribute {attribute情報}

説明	tpgetrplyを開始します
----	-----------------

4008 tpgetrply end

説明	tpgetrplyを終了します
----	-----------------

4050 invalid domain name {ドメイン名}

説明	ドメイン名が正しくありません
対応方法	ドメイン名が環境に設定されているかを確認します

4100 dialogue session is not yet established

説明	正常なtpconnectなしで会話型通信メソッドが使用されました
----	----------------------------------

4101 tpsend start with attribute {attribute情報}

説明	tpsendを開始します
----	--------------

4102 tpsend end

説明	tpsendを終了します
----	--------------

4103 tprecv start with attribute {attribute情報}

説明	tprecvを開始します
----	--------------

4104 tprecv end

説明	tprecvを終了します
----	--------------

4200 invalid queue name {RQ名}

説明	tpeqを行う場合、正しくないRQ名です
----	----------------------

4201 tpeq start with attribute {attribute情報}

説明	tpeqを開始します
----	------------

4202 tpeq end

説明	tpeqを終了します
----	------------

4203 tpdeq start with attribute {attribute情報}

説明	tpdeqを開始します
----	-------------

4204 tpdeq end

説明	tpdeqを終了します
----	-------------

4205 invalid queue type {RQ名}

説明	tpqstat、tpqsvcstatを呼び出す際のRQ名がありません
----	------------------------------------

4500 invalid topend dialogue id : {dialogue ID}

説明	topend dialogue IDが正しくありません
----	-----------------------------

4501 topend dialogue id is too big : {dialogue ID}

説明	topend dialogue IDが大きすぎます
対応方法	topendinfoのuser dialogue IDを127以下になるようにします

4502 invalid topend function qualifier : {dialogue ID}

説明	topend dialogue IDに該当するdialogueがありません
対応方法	teservicenameのfunction qualifier値を0~8388607の範囲で設定します

4503 topend dialogue status is not SEND_READY

説明	topend dialogueの状態がSEND_READYではありません
対応方法	teeserviceの状態がdial_send_readyあるいはdial_conv_send_readyの際にsendになるようにします

4504 no such topend dialogue id : {dialogue ID}

説明	topend dialogue IDが存在しません
対応方法	該当dialogue IDがサインオンしたのかを確認します

4505 topend dialogue status is not CONV_SEND_READY

説明	topend dialogueの状態がCONV_SEND_READYではありません
対応方法	teeserviceの状態がconv_send_readyになるようにします

4506 no such topend product name : {topend product名}

説明	topend product名が存在しません
対応方法	topend product nameが正しいのかを確認します

4507 invalid TEReturnMessage

説明	topendの応答メッセージが正しくありません
対応方法	topend serviceで正常に応答を受けているのかを確認します

4508 processAttachment not called

説明	processAttachmentが呼び出されませんでした
対応方法	tp_process_attachが呼び出されたのかを確認します

4509 none of topend dialogues are ready

説明	ready状態のtopend dialogueがありません
対応方法	登録されているdialogueがあるのかを確認します

4510 invalid topend dialogue status : {topend dialogueの状態}

説明	topend dialogueの状態が正しくありません
対応方法	teserviceの状態がrecv_ready、conv_recy_rady、sigonlになるように設定します

4511 topend attachment not arrived

説明	topend attachmentが到着しませんでした
対応方法	topend attachmentが到着した状態になるように設定します

4512 invalid topend action flag : {topend action flag}

説明	topend action flagが正しくありません
----	-----------------------------

対応方法	action flagがattach_transferあるいはattach_cancel状態になるように設定します
------	---

4513 tpcall for callback method start with attribute {0}

説明	現在は使用していません
----	-------------

4514 tpcall for callback method end

説明	現在は使用していません
----	-------------

4516 tpbroadcast start with nodeName{0}, usrName{1}, cliName{2}, sendBuffer{3}

説明	現在は使用していません
----	-------------

4517 tpbroadcast end

説明	現在は使用していません
----	-------------

4518 tppost start with eventName{0}, sendBuffer{1}, attribute{2}

説明	現在は使用していません
----	-------------

4519 tppost end

説明	現在は使用していません
----	-------------

4520 tpreissue start with rqname{0}, attribute{1}

説明	現在は使用していません
----	-------------

4521 tpreissue end

説明	現在は使用していません
----	-------------

4522 tpenq_ctl start with rqname{0}, attribute{1}, TMQCTL{2}

説明	現在は使用していません
----	-------------

4523 tpenq_ctl end

説明	現在は使用していません
----	-------------

4524 tpdeq_ctl start with rqname{0}, attribute{1}, TMQCTL{2}

説明	現在は使用していません
----	-------------

4525 tpdeq_ctl end

説明	現在は使用していません
----	-------------

4526 tpextsvcname start with buffer{0}

説明	現在は使用していません
----	-------------

4527 tpextsvcname end

説明	現在は使用していません
----	-------------

4528 tp_sleep start with waittime{0}

説明	現在は使用していません
----	-------------

4529 tp_sleep end

説明	現在は使用していません
----	-------------

4530 tpcancel start with cd{call descriptor値}

説明	tpcancelを開始します
----	----------------

4531 tpcancel end

説明	tpcancelを終了します
----	----------------

4532 tpmcall start with qname{0}, svcname{1}, tx{2}, attribute{3}

説明	現在は使用していません
----	-------------

4533 tpmcall end

説明	現在は使用していません
----	-------------

4534 tpgetunsol start with type{0}, attribute{1}

説明	現在は使用していません
----	-------------

4535 tpgetunsol end

説明	現在は使用していません
----	-------------

4536 tpchkunsol start

説明	現在は使用していません
----	-------------

4537 tpchkunsol end

説明	現在は使用していません
----	-------------

4538 tpacallsvg start with svgno {svgno}, serviceName {サービス名}, attribute {attribute}

説明	tpacallsvgを開始します
----	------------------

4539 tpacallsvg end

説明	tpacallsvgを終了します
----	------------------

4540 tpcallsvg start with svgno {svgno}, serviceName {サービス名}, attribute {attribute}

説明	tpcallsvgを開始します
----	-----------------

4541 tpcallsvg end

説明	tpcallsvgを終了します
----	-----------------

4542 tpgetsvglist start with svcname {サービス名}, attribute {attribute}

説明	tpgetsvglistを開始します
----	--------------------

4543 tpgetsvglist end

説明	tpgetsvglistを終了します
----	--------------------

4544 userlog start with log{0}

説明	現在は使用していません
----	-------------

4545 userlog end

説明	現在は使用していません
----	-------------

4546 ulogsync start

説明	現在は使用していません
----	-------------

4547 ulogsync end

説明	現在は使用していません
----	-------------

4548 UserLog start with log{0}

説明	現在は使用していません
----	-------------

4549 UserLog end

説明	現在は使用していません
----	-------------

4550 tpgethostaddr start with log{0}

説明	現在は使用していません
----	-------------

4551 tpgethostaddr end

説明	現在は使用していません
----	-------------

4552 tmaxlastsvc start with txbuffer{0},rxbuffer{1},attribute{2}

説明	現在は使用していません
----	-------------

4553 tmaxlastsvc end

説明	現在は使用していません
----	-------------

4554 tpgetsvglist start with nodeName {ノード名}, serviceName {サービス名}, attribute {attribute}

説明	tpgetsvglistを開始します
----	--------------------

4555 tpcallsvg start with nodeName {ノード名}, serviceName {サービス名}, attribute {attribute}

説明	tpcallsvgを開始します
----	-----------------

4556 tpcallsvg end

説明	tpcallsvgを終了します
----	-----------------

4557 tpspracall start with serviceName {サービス名}, spri {開始プロセスID}

説明	tpspracallを開始します
----	------------------

4558 tpspracall end

説明	tpspracallを終了します
----	------------------

4559 tpspracall2 start with serviceName {サービス名}, startspri {開始プロセスID}, nth {プロセス順}

説明	tpspracall2を開始します
----	-------------------

4560 tpspracall2 end

説明	tpspracall2を終了します
----	-------------------

4561 tpgetsprlist start with serviceName {サービス名}, svgno {サーバーグループID}

説明	tpgetsprlistを開始します
----	--------------------

4562 tpgetsprlist end

説明	tpgetsprlistを終了します
----	--------------------

A.5. webterrorcodeメッセージ(5000~)

5000 invalid buffer operation. current buffer type : {現在のメッセージタイプ}

説明	メッセージが正しくないメソッドを呼び出しました
----	-------------------------

5001 array index out of bounds : array.length={現在のarray length}, offset={offset}, len={length}

説明	メッセージを配列設定する際に正しくない索引を設定しました
----	------------------------------

5002 unsupported charset : {charset名}. ignored

説明	charsetが正しくありません
----	------------------

5003 contains null value

説明	メッセージのvalueにNULL値が入力されました
----	---------------------------

5004 invalid number format {変換されるメッセージ}

説明	メッセージタイプを変換する際にエラーが発生しました
----	---------------------------

5005 unsupported charset : {charsetの名前}

説明	メッセージをStringに変換する際にサポートしないcharsetです
----	-------------------------------------

5006 array index out of bounds : array.length={array length}, offset={offset}

説明	メッセージを設定する際にエラーが発生しました
----	------------------------

5007 field table is not registered

説明	FDLが設定されませんでした
----	----------------

5008 invalid field key name [{フィールド名}]

説明	フィールド名でフィールドキーが見つかりません
----	------------------------

5009 invalid field magic {0}

説明	現在は使用していません
----	-------------

5010 invalid field key value [{フィールドキー}]

説明	不明なフィールドタイプです
----	---------------

5011 invalid field index {索引}

説明	フィールド・バッファを参照する際に正しくない索引を参照しました
----	---------------------------------

5012 invalid field type {フィールドタイプ}

説明	正しくないフィールドタイプです
----	-----------------

5013 tpcallsvg start with nodeName {0}, serviceName {1}, attribute {2}

説明	現在は使用していません
----	-------------

5014 tpacallsvg start with nodeName {svgno}, serviceName {サービス名}, attribute {attribute}

説明	tpacallsvgを開始します
----	------------------

5015 tpgetactivesvr start with nodeName {0}

説明	現在は使用していません
----	-------------

5016 tpgetactivesvr end

説明	現在は使用していません
----	-------------

5017 tptobackup start

説明	現在は使用していません
----	-------------

5018 tptobackup end

説明	現在は使用していません
----	-------------

5100 FDL file read error : {ファイル・パス}

説明	FDLファイルを参照する際にエラーが発生しました
----	--------------------------

5101 invalid FDL magic {ヘッダ} from {ファイル・パス}

説明	FDLファイルを参照する際にエラーが発生しました
----	--------------------------

5102 FDL file read error. size mismatch : {read size}, {実際に読み込んだサイズ}

説明	FDLファイルを参照する際にエラーが発生しました
----	--------------------------

5103 field key table successfully loaded : {0}

説明	現在は使用していません
----	-------------

5104 fail to open FDL file {ファイル・パス}

説明	FDLファイルを参照する際にエラーが発生しました
----	--------------------------

5105 FDL file {0} is initialized

説明	現在は使用していません
----	-------------

5106 failed to initliaze FDL file {0}

説明	現在は使用していません
----	-------------

A.6. webterrorcodeメッセージ(6000~)

6001 try connect to port {0}

説明	現在は使用していません
----	-------------

6002 fail connect to port {ポート番号}

説明	webtadminでwebtへの接続に失敗しました
対応方法	該当するポートが接続可能な状態になるようにします。あるいはポートを正しく入力します

6003 socket channel configuration fail

説明	WebtAdminでソケット・チャンネルの構成に失敗しました
対応方法	exceptionでソケットのconfigureBlockingのエラー原因を確認します

6101 fail start listener

説明	WebtAdminのリスナー開始に失敗しました
対応方法	exceptionでselectorのopenエラー原因を確認します

6102 before select

説明	WebtAdminリスナーを開始する際、selectを行う前の状態です
----	-------------------------------------

6103 fail select

説明	WebtAdminリスナーを開始する際、selectに失敗しました
----	-----------------------------------

対応方法	exceptionでselectorのselectエラー原因を確認します
-------------	--------------------------------------

6104 webtadmin closed caused by {エラー・メッセージ}

説明	WebtAdminエラーによって強制終了しました
対応方法	exceptionおよびソケット接続状態を確認します

6105 fail analyze request caused by {エラー・メッセージ}

説明	WebtAdminのコマンド構成に失敗しました
対応方法	エラー・メッセージを確認します。コマンドを正常に読み込んでいるのかを確認します

6106 fail apply to Webt System caused by {エラー・メッセージ}

説明	WebtAdminのコマンド処理に失敗しました
対応方法	エラー・メッセージを確認します。正常なコマンド・バッファを読み込んでいるのかを確認します

6107 fail create reply caused by {エラー・メッセージ}

説明	WebtAdminに送信する応答処理に失敗しました
対応方法	エラー・メッセージを確認します。メッセージが正常に構成されているのかを確認します

6108 fail create listener caused by {エラー・メッセージ}

説明	WebtAdminリスナーの作成に失敗しました
対応方法	エラー・メッセージを確認します。ソケットの構成中に問題が発生した部分を確認します

6109 fail accept socket

説明	WebtAdminリスナーでソケットの受け入れに失敗しました
対応方法	ソケットの受け入れに失敗した原因を確認します

6110 accept socket {ソケット・チャンネル情報}

説明	WebtAdminリスナーでソケットが受け入れられました
----	------------------------------

A.7. webterrorcodeメッセージ(7000~)

7000 tx_begin

説明	tx_beginを開始します
----	----------------

7001 topend conversation not ended : error code = {TX_PROTOCOL_ERRORコード}

説明	topend conversationが終了しませんでした
対応方法	topend conversationが終了後にコミットします

7002 tx_commit start

説明	tx_commitを開始します
----	-----------------

7003 tx_commit end

説明	tx_commitを終了します
----	-----------------

7004 tx_rollback start

説明	tx_rollbackを開始します
----	-------------------

7005 tx_rollback end

説明	tx_rollbackを終了します
----	-------------------

7006 transaction timed out

説明	トランザクションをタイムアウトが過ぎた後に呼び出します
----	-----------------------------

7007 tx begin piggy backed

説明	トランザクション情報がピギーバックされます
----	-----------------------

7010 transaction is in progress [{現在のトランザクション状態}]

説明	txbeginを呼び出す際にすでに開始しているトランザクションです
----	-----------------------------------

7011 invalid transaction status [{現在のトランザクション状態}]

説明	txbeginを呼び出す際に不明なトランザクション状態です
----	-------------------------------

7012 transaction is completing [{現在のトランザクション状態}]

説明	現在のトランザクション状態はコミットあるいはロールバック中です
----	---------------------------------

7013 transaction result is unknown

説明	トランザクションが失敗しました
----	-----------------

7014 transaction is rolled back

説明	トランザクションがロールバックされました
----	----------------------

7015 transaction is committed

説明	トランザクションがコミットされました
----	--------------------

7016 invalid transaction result [{結果}]

説明	不明なトランザクション結果です
----	-----------------

7017 transaction is not in progress

説明	トランザクションが活性化されませんでした
----	----------------------

7018 transaction is marked rollback only

説明	ロールバックのみ可能なトランザクションです
----	-----------------------

7019 transaction is committing

説明	トランザクションをコミット中です
----	------------------

7020 transaction is rolling-back

説明	現在は使用していません
----	-------------

7050 failed to get reply for decision {Commit or Rollback}

説明	コミットあるいはロールバックの失敗応答を受け取ります
----	----------------------------

7051 failed to send decision {Commit or Rollback}

説明	コミットあるいはロールバックが失敗しました
----	-----------------------

A.8. webterrorcodeメッセージ(8000~)

8000 xa start {入力したXID}:0x{flag}

説明	xa_startを開始します
----	----------------

8001 xid mapping from {入力したXID} to {変換されたXID}

説明	XIDを変換します
----	-----------

8002 xa start success

説明	xa_startを終了します
----	----------------

8003 xa start failed

説明	xa_startに失敗しました
----	-----------------

8004 xa end {XID}:0x{flag}

説明	xa_endを開始します
----	--------------

8005 xa end success

説明	xa_endに成功しました
----	---------------

8006 xa end failed

説明	xa_endに失敗しました
----	---------------

8007 xa prepare {XID}

説明	xa_prepareを開始します
----	------------------

8008 xa prepare success

説明	xa_prepareに成功しました
----	-------------------

8009 xa prepare failed

説明	xa_prepareに失敗しました
----	-------------------

8010 xa commit {XID}:{flag}

説明	xa_commitを開始します
----	-----------------

8011 xa commit success

説明	xa_commitに成功しました
----	------------------

8012 xa commit failed

説明	xa_commitに失敗しました
----	------------------

8013 xa rollback {XID}

説明	xa_rollbackを開始します
----	-------------------

8014 xa rollback success

説明	xa_rollbackに成功しました
----	--------------------

8015 xa rollback failed

説明	xa_rollbackに失敗しました
----	--------------------

8016 xa recover 0x{flag}

説明	xa_recoverを開始します
----	------------------

8017 recovered txid {XID} is mapped to {external ID}

説明	recoverされたtxidが該当external IDにマッピングされます
----	--

8018 broken mapping for txid {XID}

説明	recoverされたxidのマッピングに失敗しました
対応方法	XIDに該当するトランザクションが開始されているかを確認します

8019 recover failed

説明	recoverに失敗しました
対応方法	例外を確認します。以前に出力されたログでrecoverの失敗原因を把握します

8020 one phase commit failed

説明	one pahse commitに失敗しました
----	-------------------------

8100 failed to create mbean {プール名}.. already exists

説明	mbeanの作成に失敗しました
----	-----------------

8101 failed to create mbean {プール名}.. not expected exception

説明	mbeanの作成に失敗しました
----	-----------------

8102 failed to enlist xa connection {コネクション情報} to transaction

説明	コネクション・プールのgetConnection時に失敗しました
----	----------------------------------

8103 xa resource {XAResource情報} is enlist to transaction {トランザクション・マネージャー情報}

説明	XAResouceトランザクションの参与に成功しました
----	-----------------------------

8104 xa resource {XAResource情報} is delisted from transaction {トランザクション・マネージャー情報}

説明	XAResouceトランザクションが解除されました
----	---------------------------

8105 failed to delist xa connection {0} from transaction

説明	現在は使用していません
----	-------------

8106 not supported action for non-xa connection

説明	non-xa connectionに適用される動作ではありません
対応方法	ソケットの接続状態を確認します

8107 managed connection is already closed. check replaceable

説明	xa要求を行うコネクションがすでに終了し、他の場所へ送信されました
----	-----------------------------------

8108 failed to acquire replacement

説明	xa要求を行うコネクションがすでに終了し、他の場所へ送信しようとしたが失敗しました
----	---

8109 failed to acquire Transaction Resource from JEUS

説明	JEUSのトランザクション・マネージャーの参照に失敗しました
----	--------------------------------

8110 failed to acquire InvocationManagerCenter Resource from JEUS

説明	autocloseの設定に失敗しました
----	---------------------

8200 failed to create data source

説明	JEUSのjnidにコネクション・プール情報のbindに失敗しました
----	------------------------------------

8201 failed to bind jndi resource

説明	現在は使用していません
----	-------------

8202 failed to register mbean server

説明	mbeanサーバーの登録に失敗しました
対応方法	JEUSが起動しているのを確認します

8203 failed to find Jeus J2EE Server

説明	JEUS J2EEサーバーが見つかりません
対応方法	JEUSが起動しているのを確認します

8204 failed to find Jeus TMXAResource

説明	現在は使用していません
----	-------------

8300 recovered xid link {XID} -> {変換されたXID}

説明	ファイルに格納されているXID一覧を出力します
----	-------------------------

8301 next sequencer is {シリアル・ナンバー}

説明	次のトランザクションのシリアル・ナンバーを表示します
----	----------------------------

8302 new GTID mapping {GTID} -> {変換されたGTID}

説明	GTIDを設定します
----	------------

8303 save link {XID} -> {変換されたXID}

説明	XIDを格納します
----	-----------

8304 remove link {XID}

説明	XIDを削除します
----	-----------

8305 get linked xid for {変換されたXID}

説明	XIDを参照します
----	-----------

8306 get all linked xid

説明	recoverするXID一覧を参照します
----	----------------------

8500 try to rollback {Tuxedo XID} for main domain recovery

説明	Tuxedoメインのドメインでロールバックを試みます
----	----------------------------

8501 failed to complete transaction {Tuxedo XID}

説明	Tuxedoドメインでトランザクション(ロールバック)が失敗しました
対応方法	例外を確認します。ロールバックの失敗原因を把握します

8502 transaction {Tuxedo XID} is completed successfully

説明	Tuxedoドメインでトランザクション(ロールバック)が成功しました
----	------------------------------------

8503 set all current transaction rollback by request

説明	Tuxedoドメインですべてのトランザクションのロールバックを開始します
----	--------------------------------------

8504 {Tuxedo XID} set rollback-only

説明	Tuxedoドメインでxidをロールバックします
----	--------------------------

8505 {トランザクション数} transaction is exists. wait 30 sec

説明	Tuxedoドメインでトランザクションの終了を待機します
----	------------------------------

8506 action : {Exception情報} is failed due to XAException.XAER_RMFAIL (communication to RM is failed)

説明	Tuxedoドメイン接続がXAER_RMFAILという理由で失敗しました
対応方法	Tuxedoドメインへの接続状態を確認します。XAER_RMFAILの発生原因を把握します

8507 connection {ドメイン・アドレス} is overruled by {ハンドラ・アドレス}

説明	Tuxedoドメインのコネクションが該当ハンドラによって制御されました
----	-------------------------------------

8508 main address {メイン・アドレス} is revived. try to failback

説明	Tuxedoドメインのメイン・アドレスが有効であるためフェイルバックを試みます
----	---

8509 register outbound xid {Tuxedo XID}

説明	Tuxedoドメインでアウトバウンドxidを登録します
----	-----------------------------

8510 unregister outbound xid {Tuxedo XID}

説明	Tuxedoドメインでアウトバウンドxidの登録を解除します
----	--------------------------------

8600 connection closed. current status {ドメインの現在の状態}

説明	Tuxedoドメインのコネクションが閉じられました
----	---------------------------

8601 start connecting.. current status {ドメインの現在の状態}

説明	Tuxedoドメインに接続を実行中です
----	---------------------

8602 connected to {ドメイン・アドレス}. current status {ドメインの現在の状態}

説明	Tuxedoドメインに接続されます
----	-------------------

8603 negotiation success with {ドメイン・アドレス}. current status {ドメインの現在の状態}

説明	Tuxedoドメインへの接続が設定されます
----	-----------------------

8604 negotiation failure with {ドメイン・アドレス}. current status {ドメインの現在の状態}

説明	Tuxedoドメインへの接続が失敗しました
----	-----------------------

8605 shutdown in progress. current status {ドメインの現在の状態}

説明	Tuxedoドメインの接続をシャットダウン中です
----	--------------------------

8606 failback in progress. current status {ドメインの現在の状態}

説明	Tuxedoドメインのフェイルバックを実行中です
----	--------------------------

8620 failed to acquire connection

説明	Tuxedoドメインのコネクション取得に失敗しました
対応方法	Tuxedoドメインへ接続された状態で実行するようにします

8699 invalid domain status {ドメイン状態}

説明	Tuxedoドメインの状態値が正しくありません
対応方法	ドメインの状態値を確認します。これが可能な状態ではない場合はお問い合わせください

A.9. webterrorcodeメッセージ(9000~)

9000 timed out

説明	要求に対してタイムアウトが発生しました
----	---------------------

9001 not active

説明	コネクション・プールが活動状態ではありません
----	------------------------

9002 invalid magic number 0x{メッセージ}

説明	応答を分析する際に正しくない(使用できない)メッセージを使用しました
----	------------------------------------

9003 unknown buffer type {0}, msgtype {1}

説明	現在は使用していません
----	-------------

9004 invalid connection handle {コネクション情報}

説明	コネクション・ハンドラの取得に失敗しました
対応方法	コネクションが作成されていないか、あるいは閉じた状態であるかを確認し、コネクションが有効な状態で呼び出されるようにします

9005 not supported action

説明	サポートしていないメソッドです
----	-----------------

9006 invalid call descriptor {0}

説明	現在は使用していません
----	-------------

9007 because transaction started can not tpcancel

説明	tpcancellに失敗しました
----	------------------

9008 fail to create ulog file with filename {0}

説明	現在は使用していません
----	-------------

9009 fail to write ulog file with filename {0}, log {1}

説明	現在は使用していません
----	-------------

9500 SSL negotiation failed in 2nd phase.. return 0x{最初の応答}

説明	暗号化に失敗しました
----	------------

9501 SSL negotiation failed in 3rd phase.. return 0x{2番目の応答}

説明	暗号化に失敗しました
----	------------

9502 2nd phase return value.. paramP={設定値}

説明	暗号化の設定をします
----	------------

9503 3rd phase return value.. serverY={設定値}

説明	暗号化の設定をします
----	------------

9504 2nd phase send value.. clientX={設定値}

説明	暗号化の設定をします
----	------------

9505 generated shared secret.. {設定値}

説明	暗号化の設定をします
----	------------

9510 creating digest by mac key {設定値}

説明	暗号化の設定をします
----	------------

9511 created digest {設定値}

説明	暗号化の設定をします
----	------------

9512 invalid mac key

説明	decryptionに失敗しました
----	-------------------

9513 decrypted digest {decryptedの値}

説明	復号情報を表示します
----	------------

9514 original digest {originalの値}

説明	復号情報を表示します
----	------------

9999 invalid magic number {メッセージ}

説明	応答を分析する際に正しくない(使用できない)メッセージを使用しました
----	------------------------------------

索引

E

EJBサービスの環境設定, 33

J

JAVA GW, 2

JEUSMain.xml

- <class-name>, 8
- <command-option>, 27
- <name>, 8
- <property-group>, 7
- <property-group><name-prefix>, 8
- <property>, 7
- <property><key>, 7
- <property><value>, 7
- backup-host-name, 9
- backup-port, 10
- check-acquired, 11
- check-alive, 13
- connect-timeout, 11
- default-charset, 10, 31
- domain-name, 10
- domain-password, 10
- enable-autoclose, 11
- enable-failback, 13
- enable-flag, 12
- encryption, 10
- event-event, 12
- event-handler, 13
- export-name, 32
- fair-provision, 11
- fdl-file, 10, 31
- header-type, 13
- host-name, 9
- host-port, 9
- listen-port, 30
- log-buffer-size, 9, 31

- log-file-format, 8, 31
- log-file-name, 8, 31
- log-level, 8, 31
- log-valid-day, 9, 31
- max, 10, 30, 33
- mbean, 14, 31
- method-signature, 32
- min, 10, 30
- monitor-interval, 13
- period, 11, 31
- service-timeout, 12
- step, 11, 30
- support-xa, 12
- tmax-delegation-source, 9
- transaction-block-timeout, 12
- transaction-timeout, 12
- type, 9
- user-name, 10
- user-password, 10

JMS, 117

JMS API, 117

JTmax, 2

jtmaxadmin, 141

W

WebT, 1

WebT Library, 2

WebT-Serverシステム, 2

webt.properties

- connectionPool.<groupName>.check, 21
- connectionPool.<groupName>.cluster, 20
- connectionPool.<groupName>.connectTimeout, 20
- connectionPool.<groupName>.domainName, 22
- connectionPool.<groupName>.domainPasswd, 22
- connectionPool.<groupName>.encryption, 21
- connectionPool.<groupName>.eventHandler, 24
- connectionPool.<groupName>.eventSvcType, 24
- connectionPool.<groupName>.fair.provision, 22
- connectionPool.<groupName>.header.type, 23
- connectionPool.<groupName>.hostAddr, 20
- connectionPool.<groupName>.hostBackupAddr, 21

connectionPool.<groupName>.hostBackupPort, 21	メソッド, 60
connectionPool.<groupName>.hostPort, 21	WebtConnection
connectionPool.<groupName>.initCapacity, 22	構築子, 37
connectionPool.<groupName>.jmsName, 126	メソッド, 39
connectionPool.<groupName>.jmsType, 126	WebtConnectionGroup
connectionPool.<groupName>.maxCapacity, 22	構築子, 47
connectionPool.<groupName>.maxIdleTime, 23	メソッド, 47
connectionPool.<groupName>.tpTimeout, 23	WebtConnectionPool, 1
connectionPool.<groupName>.txBlockTimeout, 23	WebtConnectionPool
connectionPool.<groupName>.txTimeout, 23	createGroup, 42
connectionPool.<groupName>.type, 24	メソッド, 44
connectionPool.<groupName>.userName, 22	webtdetailinfo, 137
connectionPool.<groupName>.userPasswd, 22	WebtDialogueService
connectionPool.groups, 20	構築子, 78
enableConnectionPool, 20	メソッド, 78
enableFastBackup, 19	webterrorcodeメッセージ
enableJavaSocketKeepAlive, 19	1000, 149
fdl.file, 17	1001, 149
headerType, 17	1002, 149
jeus.servlet.webt.autoClose.enable, 19	1003, 149
log.bufsize, 17	1004, 149
log.dir, 16	1005, 150
log.file, 17	1006, 150
log.file.date.format, 17	1007, 150
log.level, 16	1008, 150
log.valid.days, 17	1009, 150
monitoring.pool.checkAlive, 18	1010, 150
monitoring.pool.failback, 18	1011, 150
monitoring.pool.interval, 18	1012, 151
tmax.webt.xid.log, 24	1013, 151
useRollingDown, 19	1014, 151
webt.mbean, 24	1015, 151
webtadmin, 134	1016, 151
WebtAttribute	1017, 151
構築子, 49	1018, 151
フィールド, 49	1019, 152
メソッド, 51	1020, 152
WebtBuffer	1021, 152
構築子, 56	1022, 152
フィールド, 56	1023, 152
メソッド, 56	1024, 152
WebtCarrayBuffer / WebtStringBuffer	1025, 153
構築子, 60	1026, 153

1027, 153	1103, 159
1028, 153	1104, 159
1029, 153	1105, 159
1031, 153	1106, 160
1032, 154	1107, 160
1033, 154	1108, 160
1034, 154	1109, 160
1035, 154	1110, 160
1036, 154	1111, 160
1037, 154	1150, 160
1038, 154	1200, 161
1039, 155	1201, 161
1040, 155	1202, 161
1041, 155	1203, 161
1042, 155	1204, 161
1043, 155	1205, 161
1050, 155	1300, 161
1051, 155	1301, 162
1052, 156	1302, 162
1053, 156	1303, 162
1054, 156	1304, 162
1055, 156	1305, 162
1056, 156	1306, 162
1057, 156	1307, 162
1058, 156	1308, 163
1059, 157	1309, 163
1080, 157	1310, 163
1081, 157	1311, 163
1082, 157	1400, 163
1083, 157	1401, 163
1090, 157	1402, 163
1091, 158	1403, 163
1092, 158	1404, 164
1093, 158	1500, 164
1094, 158	1501, 164
1095, 158	1502, 164
1096, 158	1600, 164
1097, 158	1601, 164
1098, 158	1700, 164
1099, 159	1701, 164
1100, 159	1800, 165
1101, 159	1801, 165
1102, 159	1802, 165

1900, 165	2510, 171
1901, 165	2511, 171
1902, 165	2512, 171
1903, 165	2513, 171
1904, 165	2514, 171
1905, 166	2515, 171
1950, 166	2516, 172
1951, 166	2517, 172
1952, 166	3000, 172
2000, 166	3001, 172
2001, 166	3002, 172
2002, 166	3003, 172
2003, 167	3004, 172
2100, 167	3005, 173
2101, 167	3006, 173
2102, 167	3007, 173
2103, 167	4000, 173
2104, 167	4001, 173
2105, 167	4002, 173
2106, 168	4003, 173
2107, 168	4004, 174
2108, 168	4005, 174
2109, 168	4006, 174
2110, 168	4007, 174
2111, 168	4008, 174
2112, 168	4050, 174
2113, 169	4100, 174
2114, 169	4101, 174
2115, 169	4102, 175
2200, 169	4103, 175
2201, 169	4104, 175
2202, 169	4200, 175
2203, 169	4201, 175
2500, 169	4202, 175
2501, 170	4203, 175
2502, 170	4204, 175
2503, 170	4205, 176
2504, 170	4500, 176
2505, 170	4501, 176
2506, 170	4502, 176
2507, 170	4503, 176
2508, 170	4504, 176
2509, 171	4505, 176

4506, 177	4550, 182
4507, 177	4551, 182
4508, 177	4552, 183
4509, 177	4553, 183
4510, 177	4554, 183
4511, 177	4555, 183
4512, 177	4556, 183
4513, 178	4557, 183
4514, 178	4558, 183
4516, 178	4559, 184
4517, 178	4560, 184
4518, 178	4561, 184
4519, 178	4562, 184
4520, 178	5000, 184
4521, 179	5001, 184
4522, 179	5002, 185
4523, 179	5003, 185
4524, 179	5004, 185
4525, 179	5005, 185
4526, 179	5006, 185
4527, 179	5007, 185
4528, 179	5008, 185
4529, 180	5009, 185
4530, 180	5010, 186
4531, 180	5011, 186
4532, 180	5012, 186
4533, 180	5013, 186
4534, 180	5014, 186
4535, 180	5015, 186
4536, 180	5016, 186
4537, 181	5017, 187
4538, 181	5018, 187
4539, 181	5100, 187
4540, 181	5101, 187
4541, 181	5102, 187
4542, 181	5103, 187
4543, 181	5104, 187
4544, 182	5105, 187
4545, 182	5106, 188
4546, 182	6001, 188
4547, 182	6002, 188
4548, 182	6003, 188
4549, 182	6101, 188

6102, 188	8013, 194
6103, 188	8014, 194
6104, 189	8015, 195
6105, 189	8016, 195
6106, 189	8017, 195
6107, 189	8018, 195
6108, 189	8019, 195
6109, 189	8020, 195
6110, 190	8100, 195
7000, 190	8101, 195
7001, 190	8102, 196
7002, 190	8103, 196
7003, 190	8104, 196
7004, 190	8105, 196
7005, 190	8106, 196
7006, 191	8107, 196
7007, 191	8108, 196
7010, 191	8109, 197
7011, 191	8110, 197
7012, 191	8200, 197
7013, 191	8201, 197
7014, 191	8202, 197
7015, 191	8203, 197
7016, 192	8204, 197
7017, 192	8300, 198
7018, 192	8301, 198
7019, 192	8302, 198
7020, 192	8303, 198
7050, 192	8304, 198
7051, 192	8305, 198
8000, 193	8306, 198
8001, 193	8500, 198
8002, 193	8501, 199
8003, 193	8502, 199
8004, 193	8503, 199
8005, 193	8504, 199
8006, 193	8505, 199
8007, 194	8506, 199
8008, 194	8507, 199
8009, 194	8508, 200
8010, 194	8509, 200
8011, 194	8510, 200
8012, 194	8600, 200

8601, 200
8602, 200
8603, 200
8604, 201
8605, 201
8606, 201
8620, 201
8699, 201
9000, 201
9001, 201
9002, 202
9003, 202
9004, 202
9005, 202
9006, 202
9007, 202
9008, 202
9009, 203
9500, 203
9501, 203
9502, 203
9503, 203
9504, 203
9505, 203
9510, 203
9511, 204
9512, 204
9513, 204
9514, 204
9999, 204
WebtEventConneccion
 構築子, 102
 メソッド, 106
WebtEventHandler
 フィールド, 110
 メソッド, 110
WebtFDLKeyTable
 構築子, 86
 メソッド, 86
WebtField
 構築子, 95
 メソッド, 96
WebtFieldBuffer

 構築子, 100
 メソッド, 100
WebtFieldElement
 メソッド, 81
WebtFieldSet
 構築子, 88
 メソッド, 88
webtinfo, 136
webtinfo2, 137
webtreconfig, 138
WebtRemoteService
 構築子, 65
 メソッド, 65
WebtRQService
 構築子, 71
 フィールド, 72
 メソッド, 73
webtshrink, 138
WebtSystem
 構築子, 114
 メソッド, 114
WebtTransaction
 構築子, 111
 メソッド, 111

ま

 メッセージング・システム, 117

ら

 ローリング・ダウン機能, 27

