

# WebOTX 運用編(ログ)

WebOTX 運用編

バージョン: 7.1

版数: 第 11 版

リリース: 2010 年 9 月

Copyright (C) 1998 - 2010 NEC Corporation. All rights reserved.

---

# 目次

1. はじめに .....	1
2. ロギングの概要 .....	2
2.1. WebOTX ASのログについて .....	2
2.1.1. ドメインを構成するサービスについて .....	2
2.1.2. サービスとログの関係 .....	4
2.2. 業務アプリケーションとの関係 .....	5
2.2.1. Log4Jを利用する場合 .....	5
2.2.2. 標準出力を利用する場合 .....	6
3. 各サービスのログ .....	7
3.1. 運用管理 .....	7
3.1.1. server.log .....	7
3.1.2. webotx_agent.log .....	7
3.1.3. webotx_admin.log .....	8
3.1.4. webotx_security.log .....	8
3.1.5. agent_access.log .....	9
3.2. CHDL .....	9
3.2.1. webotx_chdl.log .....	9
3.3. Deploy .....	10
3.3.1. webotx_deploy.log .....	10
3.4. EJBコンテナ .....	10
3.4.1. webotx_ejbcont.log .....	10
3.5. JDBC .....	10
3.5.1. wojdbc.log .....	10
3.5.2. wojta.log .....	11
3.6. JMS .....	11
3.6.1. wojms.log .....	11
3.6.2. wojmsservice.log .....	12
3.6.3. std.log .....	12
3.6.4. wojmsadmin.log .....	12
3.6.5. wojmspocket.log .....	13
3.6.6. wojmsmessage.log .....	13
3.6.7. wojmserror.log .....	14
3.6.8. wojmssv.pid .....	15
3.6.9. wojmsexec.log .....	15
3.7. JNDI .....	15
3.7.1. webotx_jndisp.log .....	15
3.7.2. webotx_naming.log .....	16

3.8. Object Broker.....	16
3.8.1. objava.log .....	16
3.8.2. webotx_ospi.log .....	16
3.8.3. ObLog.log .....	17
3.8.4. Oad.log .....	17
3.8.5. InterfaceRepository.log.....	18
3.8.6. corbaloc.log.....	18
3.8.7. namesv.log.....	19
3.8.8. cnamesv.log .....	19
3.8.9. message.log .....	20
3.8.10. syslog.log .....	21
3.8.11. oadj.log .....	21
3.9. TPモニタ.....	22
3.9.1. webotx_tpmmgr.log.....	22
3.9.2. \${PGNAME}.\${PID}.log .....	22
3.9.3. \${PGNAME}_sys.\${PID}.log.....	23
3.9.4. \${PGNAME}_recovery.\${PID}.log .....	24
3.9.5. \${APGNAME}_EXIT_sys.\${PID}.log .....	24
3.9.6. \${SYSTEMNAME}_IIOPLsn.log \${SYSTEMNAME}_A_IIOPLsn.log .....	25
3.9.7. \${SYSTEMNAME}_woejout<number>.log.....	25
3.10. Transactionサービス .....	25
3.10.1. webotx_ts.log.....	25
3.10.2. RCS.trc .....	26
3.10.3. LRS.trc .....	26
3.10.4. webotx_ts_cpp.log .....	27
3.11. Webコンテナ .....	27
3.11.1. webotx_catalina.log .....	27
3.11.2. webotx_webcont.log .....	27
3.11.3. webotx_container.log .....	28
3.11.4. webotx_webc_message.log.....	28
3.11.5. webotx_webc_stat_vm.log .....	29
3.11.6. webotx_webc_stat_webap.log.....	29
3.11.7. webotx_struts.log.....	30
3.11.8. mod_jk.log .....	30
3.11.9. mod_jk_20.log .....	30
3.11.10. isapi.log.....	31
3.11.11. nsapi.log .....	31
3.11.12. mod_jk_om.log.....	32
3.11.13. mod_jk_om-20.log .....	32
3.12. Webサーバ.....	32
3.12.1. webotx_websv.log.....	32

3.12.2.	access_log(access.log) .....	33
3.12.3.	error_log(error.log).....	33
3.12.4.	ssl_request_log(ssl_request.log).....	34
3.12.5.	server_access.log.....	34
3.13.	Web Service .....	35
3.13.1.	webservice.log .....	35
3.13.2.	wstrm.log .....	35
3.14.	Working Domain Coordinator.....	35
3.14.1.	wowdc.log .....	35
4.	設定値と設定方法について .....	37
4.1.	ログレベルについて .....	37
4.1.1.	WebOTX ASのログレベル .....	37
4.1.2.	各種ログレベルの対応関係について.....	38
4.1.3.	モジュールログレベルの設定 .....	40
4.1.4.	モジュールログレベルの設定方法 .....	43
4.1.5.	トレースログレベルの設定.....	43
4.1.6.	トレースログレベルの設定方法 .....	47
4.1.7.	Log4otxログレベル .....	47
4.1.8.	Log4otxログレベルの設定方法 .....	47
4.1.9.	ログレベルの適用順序 .....	48
4.2.	Log4otxのローテーションの設定方法 .....	49
4.2.1.	ログファイルの最大サイズの変更.....	49
4.2.2.	ログの世代数の変更 .....	50
4.3.	rotatelogsを利用したログローテーションの設定方法.....	50
4.3.1.	server.logのローテーション.....	50
4.3.2.	Webサーバのログのログローテーション .....	51
4.3.3.	mod_jkのログのローテーション .....	51
4.4.	Log4Jで利用できるJavaシステムプロパティ .....	53
4.4.1.	Log4Jで利用できるJavaシステムプロパティ .....	53
5.	サーバAPのトレースログについて.....	54
5.1.	Standard/Enterprise EditionにおけるサーバAPのログ .....	54
5.1.1.	サーバアプリケーショントレース採取 .....	54
5.1.2.	トレースの読み方(Java) .....	56
5.1.3.	トレースの読み方(C++) .....	58
5.1.4.	サーバアプリケーション内部トレースの採取方法 .....	60
5.1.5.	IIOPリスナ内部トレースの採取方法 .....	61
6.	注意・制限事項について .....	62
6.1.	連続運用時のディスクフルについて .....	62
6.2.	アプリケーションがLog4Jを利用する場合の注意点 .....	62
6.2.1.	WebOTXにバンドルされているLog4Jを利用する場合 .....	62
6.2.2.	Webアプリケーションに含まれるLog4Jを利用する場合 .....	62

6.2.3. マルチプロセスモードでWebアプリケーションのログ出力を行う場合 .....	63
7. 参照 .....	65
7.1. モジュールとロガーとアペンダの一覧 .....	65
7.2. アペンダ一覧.....	66
7.3. ディスクフルの危険性一覧.....	71

# 1.はじめに

本書は WebOTX AS のログについて、その概要や設定方法について記載しています。

## 対象読者

このマニュアルは WebOTX Application Server Web Edition、Standard-J Edition、Standard Edition、Enterprise Edition を使って運用環境を構築するシステムエンジニア、日々の運用を行うオペレータを対象としています。

## 表記について

### パス名表記

本書ではパス名の表記については特に OS を限定しない限りセパレータはスラッシュ '/' で統一しています。Windows 環境においては '\$' に置き換えてください。

### 環境変数表記

インストールディレクトリやドメインルートディレクトリなど環境によって値の異なるものについては環境変数を用いて表します。

`$(env)` または `$(env)` で表しています。

例)

`$(AS_INSTALL)`: インストールディレクトリ

`$(INSTANCE_ROOT)`: ドメインルートディレクトリ

`$(APGNAME)`: アプリケーショングループ名

`$(PGNAME)`: プロセスグループ名

`$(SYSTEMNAME)`: システム名

`$(PID)`: プロセス ID

## コマンド操作について

本書中では運用操作に用いるコマンドの詳細についての説明は省略しています。

コマンドの詳細は「運用管理コマンド」、「運用管理コマンドリファレンス」を参照してください。

## 2. ロギングの概要

ログは日々の運用でのメンテナンスや障害発生時の原因調査など、ログはシステム運営において重要な位置を占めています。本章では WebOTX AS のロギングおよび業務アプリケーションからのログ出力方法について、その概要を説明します。

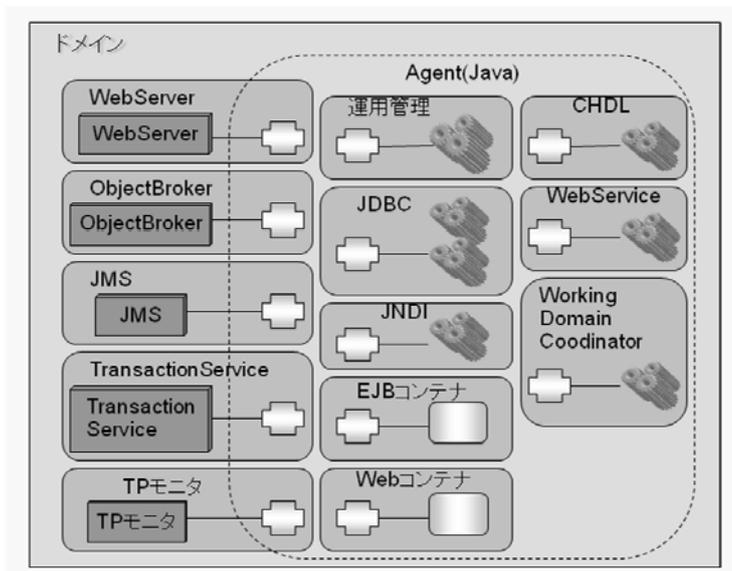
### 2.1. WebOTX ASのログについて

この節では WebOTX AS が出力するログの概要を説明します。

#### 2.1.1. ドメインを構成するサービスについて

WebOTX AS は運用領域をドメインと呼ばれる単位にグルーピングして管理しています。ドメインは各種の機能を提供するサービス群から構成されており、各サービスはその機能を提供するプロセス群またはモジュールと制御モジュールから構成されます。

各サービスは運用管理サービスの中核を担うエージェントプロセス内のモジュールとその制御モジュールから構成されるサービスと、エージェントプロセスの外部のプロセス群とエージェントプロセス内の制御モジュールから構成されるサービスがあります。



1つのドメイン内で出力されるログはそのドメインを構成するサービスが個々の方法を利用して出力しています。そのため、ログの出力方法はサービスにより異なります。

WebOTX AS のログの出力方法は大きくは以下のように分けられます。

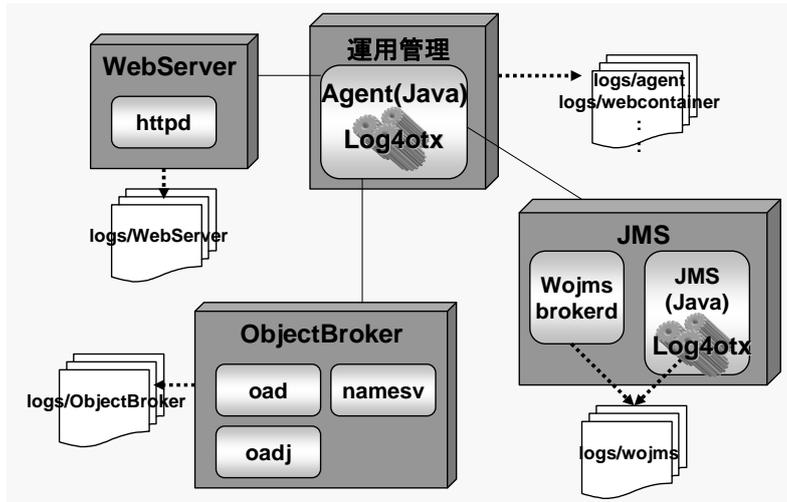
- Log4otx(\*1)によるログ出力
- サービス独自の方法によるログ出力

このログ出力方法の違いにより、WebOTX AS でのログの設定方法が異なります。そこで、WebOTX AS のログの概要を理解するために、まずは WebOTX AS を構成するサービスの関係について説明します。

ここで、WebOTX AS ではエディションによりドメインを構成するサービスが異なるため、以下にエディション別のサービス構成図を記します。

(\*1)Log4Jをベースとした WebOTX AS 独自のログモジュールです

- Web/Standard-J Edition

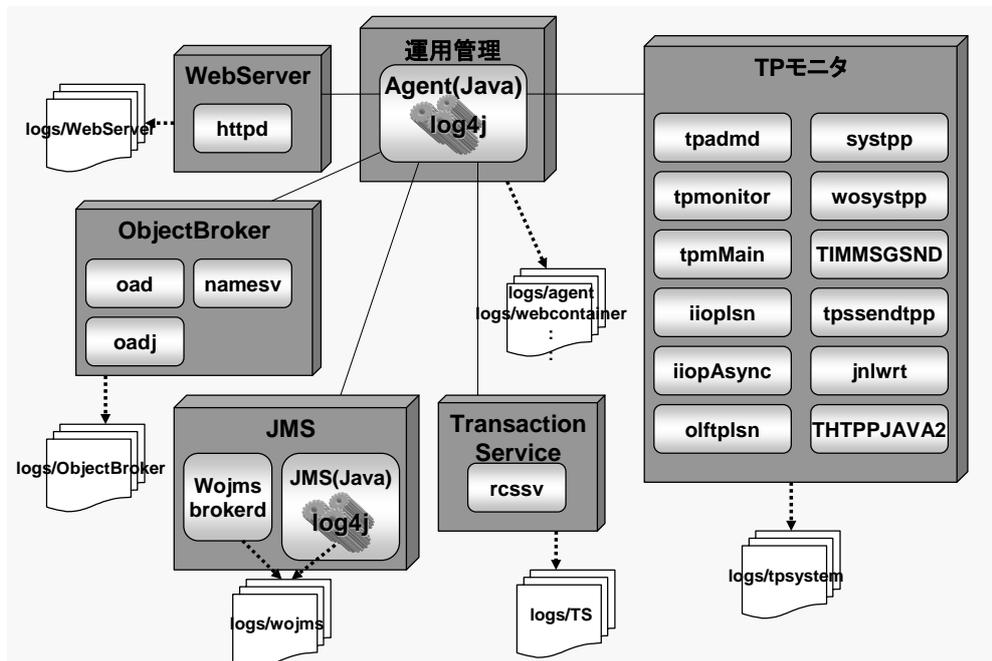


Web Edition および、Standard-J Edition でのドメインは、ドメイン全体を管理する運用管理サービス、JMS 関連の機能を提供する JMS サービス、CORBA 通信関連の機能を提供する Object Broker サービス、Web サーバ関連の機能を提供する Web Server サービスから構成されます。

ここで、運用管理サービスのエージェントプロセスおよび JMS コンポーネントの JMS プロセスはともに Java のプロセスであり、ログの出力には Log4otx を用いています。その他のプロセスは独自の方法を利用してログを出力しています。

各サービスとログの関係の詳細については次節を参照してください。

- Standard/Enterprise Edition



Standard Edition および、Enterprise Edition では、Standard-J Edition を構成するサービスに加え、Transaction の管理機能を提供する Transaction サービス、高信頼実行基盤である TP モニタサービスから構成されます。また、運用管理サービスのエージェントプロセスおよび、JMS コンポーネントの JMS プロセスはともに Log4otx を、その他のプロセスは独自の機構を利用してログを出力しています。

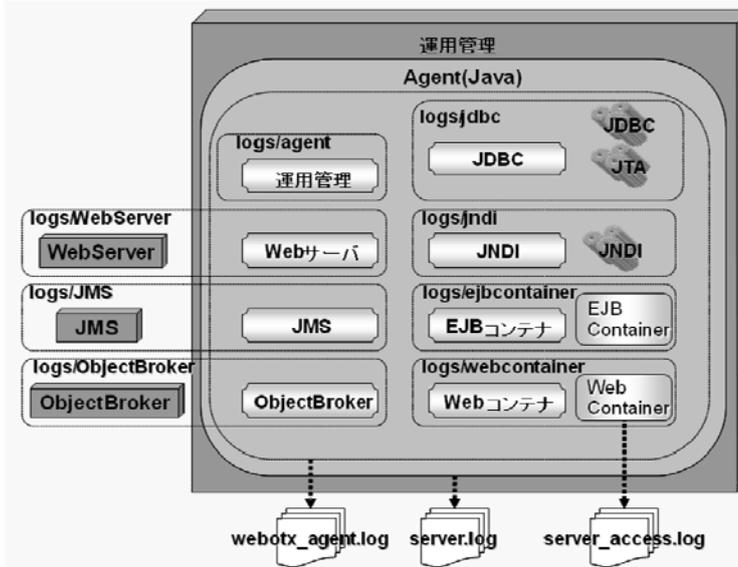
各コンポーネントとログの関係については次節を参照してください。

## 2.1.2. サービスとログの関係

前節でも述べたように、サービスはプロセス群またはモジュールと制御モジュールから構成されています。また、制御モジュールはエージェントプロセスの内部で稼動しています。ここで `$(INSTANCE_ROOT)/logs` にはサービスごとにディレクトリが分けられており、各ディレクトリにはプロセス群またはモジュールが出力するログファイルと制御モジュールが出力するログファイルが格納されます。

以下はエディション別のサービスと出力先ディレクトリの関係図です。

- Web/Standard-J Edition



Web サーバ、JMS、Object Broker のサービスはそれぞれ外部プロセス群とエージェントプロセス内の制御モジュールとから構成され、その他のサービスはエージェントプロセス内のモジュールと制御モジュールから構成されています。また、サービスごとに `$(INSTANCE_ROOT)/logs` 内にディレクトリ区分されログが出力されます。

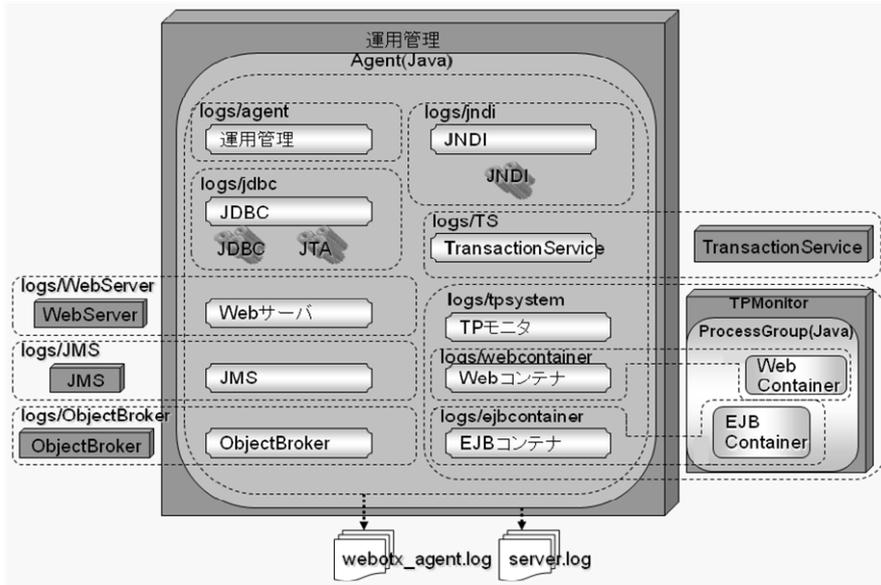
`$(INSTANCE_ROOT)/logs` 直下に出力されるとログファイルについては以下のようなログが出力されます。

`webotx_agent.log` ... エージェントプロセス内の全てのサービスが出力するメッセージが出力されます。

`server.log` ... エージェントプロセスの標準出力、標準エラーが出力されます。

`server_access.log` ... Web コンテナへのアクセスログが出力されます。

- Standard/Enterprise Edition



Web/Standard-J Edition の場合に加え、Transaction サービス、TPMonitor サービスが各々外部プロセス群とサービスから構成され、追加されます。また、Web/Standard-J Edition ではエージェントプロセス内で動作していた Web コンテナ(\*2)、EJB コンテナが TPMonitor 上で動作します。そして、サービスごとに `#{INSTANCE_ROOT}/logs` 内にディレクトリ区分されログが出力されます。

また、`#{INSTANCE_ROOT}/logs` 直下に出力される `webotx_agent.log`、`server.log` については Web/Standard-J Edition と同様です。

(\*2)マルチプロセスモードを選択した場合

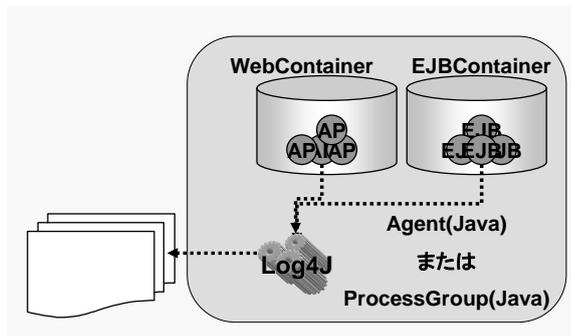
## 2.2. 業務アプリケーションとの関係

この節では WebOTX AS と WebOTX AS 上に配備された業務アプリケーションとの関係について説明します。

WebOTX AS 上に配備された業務アプリケーションからログを出力するには Log4J を利用する方法と WebOTX AS が提供する API を利用する方法および標準出力にメッセージを出力する方法があります。

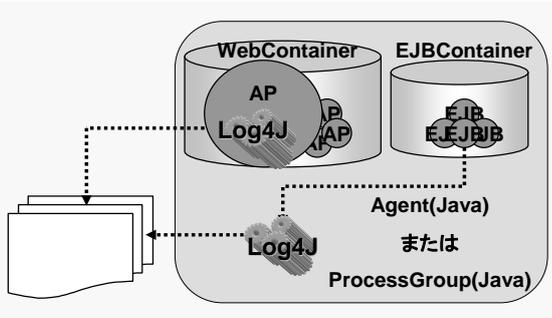
Log4J を利用する場合は次節を、WebOTX AS が提供する API を利用する場合は「API リファレンスマニュアル」を参照してください。

### 2.2.1. Log4Jを利用する場合



WebOTX AS では業務アプリケーションが動作する Java プロセス(エージェントまたはプロセスグループ)上に WebOTX が Web アプリケーション向けに標準でバンドルしている Log4J がロードされます。業務アプリケーションからログを出力したい場合、その Log4J を利用することが可能です。

設定手順については「4.4.1 WebOTX にバンドルされている Log4J を利用する場合」を参照してください。



上記の方法の他に、配備した Web アプリケーションに含まれる Log4J を利用する方法があります。この方法を利用する場合、アプリケーションが利用する log4j 定義ファイルは、WEB-INF/classes 以下に配置する必要があります。

設定手順については「4.4.2 Web アプリケーションに含まれる Log4J を利用する場合」を参照してください。

## 2.2.2. 標準出力を利用する場合

標準出力を利用する場合、エディションまたはドメインの構成により以下のファイルへ出力されます。

- Web/Standard-J Edition および、Standard/Enterprise Edition をシングルプロセスモードでインストールした場合  
Web アプリケーションから標準出力に出力したメッセージは server.log に出力されます。また、EJB から標準出力に出力したメッセージはサーバアプリケーショントレースログに出力されます。
- Standard/Enterprise Edition をマルチプロセスモードでインストールした場合  
標準出力に出力したメッセージはサーバアプリケーショントレースログに出力されます。

## 3. 各サービスのログ

WebOTX の各サービスが出力するログについて説明します。

WebOTX の各オプション製品が出力するログに関しては、各製品のマニュアルを参照してください。

### 3.1. 運用管理

#### 3.1.1. server.log

- **概要**

エージェントプロセス(Java VM)の標準出力、標準エラー出力を出力するログです。

Log4otx は利用せず、サービス独自の方法でエージェントプロセスの標準出力、標準エラー出力をリダイレクトしています。

- **ログサイズ変更方法**

以下のコマンドを実行することで、ローテーションのサイズを変更することができます。

```
otxadmin > set server.log-service.log-rotation-limit-in-bytes=<サイズ(byte)>
```

デフォルトでは 10MB に設定されています。

- **ローテーション設定方法**

ドメイン起動中にローテーションすることはありませんが、ドメイン起動時に設定されたファイルサイズに達していた場合にファイル名に世代番号が付加されてバックアップされます。世代数を越えたバックアップファイルについては削除されます。世代数の変更は以下のコマンドで変更することができます。デフォルトでは 3 となっています。

```
otxadmin > set server.log-service.log-rotation-generations=<世代数>
```

UNIX 系 OS の場合は、WebOTX Web サーバ 2.0(Apache2.0)付属の `rotatelog` を使用することで、ドメイン運用中に指定期間やファイルサイズによる切り替えを行うことが可能です。連続運用など WebOTX を停止せずに `server.log` の切り替えを行う必要がある場合に有効です。

設定方法に関しては、[4.1.3 `rotatelog` を利用したログローテーションの設定方法] を参照してください。

#### 3.1.2. webotx\_agent.log

- **概要**

WebOTX ドメインのエージェント内の JVM で出力される全てのメッセージを統合したログです。Log4otx を利用して出力されています。

全てのモジュールのログメッセージが出力されます。モジュールとロガーの対応については、[7.1.1. モジュールおよびロガーの一覧] を参照してください。

- **ログファイルのサイズ**

`log4otx.xml` の Appender 名が FILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション世代数**

`log4otx.xml` の Appender 名が FILELOG の MaxBackupIndex 値を変更することで、ログファイル

世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

設定方法に関しては、[4.1.3.モジュールログレベルの設定方法] を参照してください。

### 3.1.3. webotx\_admin.log

- **概要**

WebOTX ドメインのエージェント内の運用管理機能が出力するログです。Log4otx を利用して出力されています。

以下のモジュールに属するログガーのログメッセージが出力されます。モジュールとログガーの対応については、[7.1.1. モジュールおよびログガーの一覧] を参照してください。

- ◇ admin
- ◇ classloader
- ◇ configuration
- ◇ naming
- ◇ security

- **ログファイルのサイズ**

log4otx.xml の Appender 名が ADMINFILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が ADMINFILELOG の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げた各モジュールのモジュールログレベルを変更してください。設定方法に関しては、[4.1.3.モジュールログレベルの設定方法] を参照してください。

### 3.1.4. webotx\_security.log

- **概要**

WebOTX ドメインのエージェント内のセキュリティに関するログです。Log4otx を利用して出力されています。

以下のモジュールに属するログガーのログメッセージが出力されます。モジュールとログガーの対応については、[7.1.1. モジュールおよびログガーの一覧] を参照してください。

- ◇ security

- **ログサイズ変更方法**

log4otx.xml の Appender 名が SECURITYFILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が SECURITYFILELOG の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げた各モジュールのモジュールログレベルを変更してください。設定方法に関しては、[4.1.

3.モジュールログレベルの設定方法]を参照してください。

### 3.1.5. agent\_access.log

- **概要**

統合運用管理ツールまたは運用管理コマンドからの運用操作履歴に関するログです。Log4otx を利用して出力されています。

以下のモジュールに属するロガーのログメッセージが出力されます。モジュールとロガーの対応については、[7.1.1. モジュールおよびロガーの一覧]を参照してください。

◇ jmx-remote-access

- **設定方法**

既定では出力されません。以下のコマンドを実行して機能を有効にしてください。

また、設定を反映するためにドメインを再起動する必要があります。

```
otxadmin > set server.admin-service.jmx-connector.jmx_connector.access-log-enabled=true
```

- **ログサイズ変更方法**

log4otx.xml の Appender 名が JMXREMOTEACCESSLOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が JMXREMOTEACCESSLOG の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げた各モジュールのモジュールログレベルを変更してください。設定方法に関しては、[4.1.3.モジュールログレベルの設定方法]を参照してください。

## 3.2. CHDL

### 3.2.1. webotx\_chdl.log

- **概要**

XML 高速変換エンジン(CHDL)に関するログです。Log4otx を利用して出力されています。

以下のロガーのログメッセージが出力されます。ロガーについては、[7.1.1. モジュールおよびロガーの一覧]を参照してください。

◇ com.nec.jp.rws.chdl

- **ログサイズ変更方法**

log4otx.xml の Appender 名が CHDLFILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が CHDLFILELOG の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げたロガーのログレベルを変更してください。設定方法に関しては、[4.1.7.Log4otx のログレベルの設定方法]を参照してください。

## 3.3. Deploy

### 3.3.1. webotx\_deploy.log

- **概要**

アプリケーション配備に関するログです。Log4otx を利用して出力されています。

以下のモジュールに属するロガーのログメッセージが出力されます。モジュールとロガーの対応については、[7.1.1. モジュールおよびロガーの一覧] を参照してください。

◇ deployment

- **ログサイズ変更方法**

log4otx.xml の Appender 名が DEPLOYFILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が DEPLOYFILELOG の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げた各モジュールのモジュールログレベルを変更してください。設定方法に関しては、[4.1. 3.モジュールログレベルの設定方法] を参照してください。

## 3.4. EJBコンテナ

### 3.4.1. webotx\_ejbcont.log

- **概要**

EJB コンテナが出力するログです。Log4otx を利用して出力されています。

以下のモジュールに属するロガーのログメッセージが出力されます。モジュールとロガーの対応については、[7.1.1. モジュールおよびロガーの一覧] を参照してください。

◇ ejb-container

- **ログサイズ変更方法**

log4otx.xml の Appender 名が EJBCONTAINERFILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が EJBCONTAINERFILELOG の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げた各モジュールのモジュールログレベルを変更してください。設定方法に関しては、[4.1. 3.モジュールログレベルの設定方法] を参照してください。

## 3.5. JDBC

### 3.5.1. wojdbc.log

- **概要**

JDBC データソースに関するログです。Log4otx を利用して出力されています。

以下のモジュールに属するロガーのログメッセージが出力されます。モジュールとロガーの対応については、[7.1.1. モジュールおよびロガーの一覧] を参照してください。

◇ jdbc-datasource

- **ログサイズ変更方法**

log4otx.xml の Appender 名が WOJDBCFILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が WOJDBCFILELOG の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げた各モジュールのモジュールログレベルを変更してください。設定方法に関しては、[4.1.3.モジュールログレベルの設定方法] を参照してください。

### 3.5.2. wojta.log

- **概要**

JTA に関するログです。Log4otx を利用して出力されています。

以下のモジュールに属するロガーのログメッセージが出力されます。モジュールとロガーの対応については、[7.1.1. モジュールおよびロガーの一覧] を参照してください。

◇ jta

- **ログサイズ変更方法**

log4otx.xml の Appender 名が WOJTAFILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が WOJTAFILELOG の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げた各モジュールのモジュールログレベルを変更してください。設定方法に関しては、[4.1.3.モジュールログレベルの設定方法] を参照してください。

## 3.6. JMS

### 3.6.1. wojms.log

- **概要**

JMS リソースアダプタおよびクライアントに関するログです。Log4otx を利用して出力されています。

以下のモジュールに属するロガーのログメッセージが出力されます。モジュールとロガーの対応については、[7.1.1. モジュールおよびロガーの一覧] を参照してください。

◇ jms-client

- **ログサイズ変更方法**

log4otx.xml の Appender 名が WOJMSFILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が WOJMSFILELOG の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げた各モジュールのモジュールログレベルを変更してください。設定方法に関しては、[4.1.3.モジュールログレベルの設定方法]を参照してください。

### 3.6.2. wojmsservice.log

- **概要**

JMS サーバのログ情報です。Log4otx は利用せず、サービス独自の方法で出力しています。

- **ログサイズ変更方法**

以下のコマンドで変更することができます。デフォルトでは 1048576 バイト(1MB)となっています。

```
otxadmin > set server:jms-service logfileRolloverBytes=<サイズ(バイト)>
```

- **ローテーション設定方法**

ローテーションは上記で設定したファイルサイズを超えた場合、または経過時間で行うことができます。経過時間を設定する場合は、以下のコマンドで、秒単位で指定してください。デフォルトでは1(切り替えなし)が設定されています。

```
otxadmin > set server:jms-service logfileRolloverSecs=<秒>
```

ローテーション数は、デフォルトで9世代となっています。変更する場合、\${INSTANCE\_ROOT}/wojms/instances/wojmsbroker/props/config.properties に以下の定義を追加してください。

```
wojms.log.file.filenum=<世代数>
```

- **ログレベル変更方法**

以下のコマンドでログレベルを変更してください。

```
otxadmin > set server:jms-service loglevel=<ログレベル>
```

設定できるログレベルは以下になります。

NONE、ERROR、WARNING、INFO、DEBUG、DEBUGMED、DEBUGHIGH

### 3.6.3. std.log

- **概要**

JMS サーバの標準出力、標準エラー出力。JMS サーバ起動時のエラーや、スレッドダンプの取得結果を出力するログです。Log4otx は利用せず、サービス独自の方法で出力しています。

- **ログサイズ変更方法**

ログサイズは無制限です。

- **ローテーション設定方法**

JMS サーバが起動するタイミングで std.log.bak にリネームされ、一世代分のバックアップを行います。

- **ログレベル変更方法**

ログレベルはありません。

### 3.6.4. wojmsadmin.log

- **概要**

JMS 管理コマンドや、統合運用管理ツール、運用管理コマンドからの JMS サーバに対する運用管理操作の実行履歴です。Log4otx は利用せず、サービス独自の方法で出力しています。

- **ログサイズ変更方法**

以下のコマンドで変更することができます。デフォルトでは 1048576 バイト(1MB)となっています。

```
otxadmin > set server.jms-service.logfileRolloverBytesAdmin=<サイズ(バイト)>
```

- **ローテーション設定方法**

ローテーションは上記で設定したファイルサイズを超えた場合、または経過時間で行うことができます。経過時間を設定する場合は、以下のコマンドで、秒単位で指定してください。デフォルトでは・1(切り替えなし)が設定されています。

```
otxadmin > set server.jms-service.logfileRolloverSecsAdmin=<秒>
```

ローテーション数は、デフォルトで9世代となっています。変更する場合、`${INSTANCE_ROOT}/wojms/instances/wojmsbroker/props/config.properties` に以下の定義を追加してください。

```
wojms.log.admin.fileenum=<世代数>
```

- **ログレベル変更方法**

ログレベル変更方法

以下のコマンドでログレベルを変更してください。

```
otxadmin > set server.jms-service.loglevel=<ログレベル>
```

設定できるログレベルは以下になります。

NONE、ERROR、WARNING、INFO、DEBUG、DEBUGMED、DEBUGHIGH

### 3.6.5. wojmspacket.log

- **概要**

JMS サーバと JMS クライアント間で送受信される JMS メッセージのパケットの情報です。Log4otx は利用せず、サービス独自の方法で出力しています。

- **ログサイズ変更方法**

以下のコマンドで変更することができます。デフォルトでは 1048576 バイト(1MB)となっています。

```
otxadmin > set server.jms-service.logfileRolloverBytesPacket=<サイズ(バイト)>
```

- **ローテーション設定方法**

ローテーションは上記で設定したファイルサイズを超えた場合、または経過時間で行うことができます。経過時間を設定する場合は、以下のコマンドで、秒単位で指定してください。デフォルトでは・1(切り替えなし)が設定されています。

```
otxadmin > set server.jms-service.logfileRolloverSecsPacket=<秒>
```

ローテーション数は、デフォルトで9世代となっています。変更する場合、`${INSTANCE_ROOT}/wojms/instances/wojmsbroker/props/config.properties` に以下の定義を追加してください。

```
wojms.log.packet.fileenum=<世代数>
```

- **ログレベル変更方法**

以下のコマンドでログレベルを変更してください。

```
otxadmin > set server.jms-service.loglevel=<ログレベル>
```

設定できるログレベルは以下になります。

NONE、ERROR、WARNING、INFO、DEBUG、DEBUGMED、DEBUGHIGH

### 3.6.6. wojmsmessage.log

- **概要**

メッセージライフサイクルの情報です。Log4otx は利用せず、サービス独自の方法で出力しています。JMS メッセージの送信先に対する送信(ADD)、送信先からの削除(REMOVE)に関する情報を出力します。

- **ログサイズ変更方法**

以下のコマンドで変更することができます。デフォルトでは 1048576 バイト(1MB)となっています。

```
otxadmin > set server.jms-service.logfileRolloverBytesMessage=<サイズ(バイト)>
```

- **ローテーション設定方法**

ローテーションは上記で設定したファイルサイズを超えた場合、または経過時間で行うことができます。経過時間を設定する場合は、以下のコマンドで、秒単位で指定してください。デフォルトでは・1(切り替えなし)が設定されています。

```
otxadmin > set server.jms-service.logfileRolloverSecsMessage=<秒>
```

ローテーション数は、デフォルトで9世代となっています。変更する場合、`${INSTANCE_ROOT}/wojms/instances/wojmsbroker/props/config.properties` に以下の定義を追加してください。

```
wojms.log.message.filenum=<世代数>
```

- **ログレベル変更方法**

以下のコマンドでログレベルを変更してください。

```
otxadmin > set server.jms-service.loglevel=<ログレベル>
```

設定できるログレベルは以下になります。

NONE、ERROR、WARNING、INFO、DEBUG、DEBUGMED、DEBUGHIGH

### 3.6.7. wojmserror.log

- **概要**

JMS が出力するエラー情報です。Log4otx は利用せず、サービス独自の方法で出力しています。

- **ログサイズ変更方法**

以下のコマンドで変更することができます。デフォルトでは 1048576 バイト(1MB)となっています。

```
otxadmin > set server.jms-service.logfileRolloverBytesError=<サイズ(バイト)>
```

- **ローテーション設定方法**

ローテーションは上記で設定したファイルサイズを超えた場合、または経過時間で行うことができます。経過時間を設定する場合は、以下のコマンドで、秒単位で指定してください。デフォルトでは・1(切り替えなし)が設定されています。

```
otxadmin > set server.jms-service.logfileRolloverSecsError=<秒>
```

ローテーション数は、デフォルトで9世代となっています。変更する場合、`${INSTANCE_ROOT}/wojms/instances/wojmsbroker/props/config.properties` に以下の定義を追加してください。

```
wojms.log.error.filenum=<世代数>
```

- **ログレベル変更方法**

以下のコマンドでログレベルを変更してください。

```
otxadmin > set server.jms-service.loglevel=<ログレベル>
```

設定できるログレベルは以下になります。

NONE、ERROR、WARNING、INFO、DEBUG、DEBUGMED、DEBUGHIGH

### 3.6.8. wojmssv.pid

- **概要**

JMS サーバのプロセス ID が出力されます。Log4otx は利用せず、サービス独自の方法で出力しています。

- **ログサイズ変更方法**

JMS サーバが起動するタイミングでファイルを切り替えます。

- **ローテーション設定方法**

JMS サーバが起動するタイミングでファイルを切り替えます。

- **ログレベル変更方法**

ログレベルなし

### 3.6.9. wojmsexec.log

- **概要**

JMS サーバの起動、停止時の情報です。Log4otx は利用せず、サービス独自の方法で出力しています。

JMS サーバの起動引数や、終了コードなどを出力します。

- **ログサイズ変更方法**

JMS サーバが起動するタイミングでファイルを切り替えます。

- **ローテーション設定方法**

JMS サーバが起動するタイミングでファイルを切り替えます。

- **ログレベル変更方法**

ログレベルなし

## 3.7. JNDI

### 3.7.1. webotx\_jndisp.log

- **概要**

JNDI サービスに関するログです。Log4otx を利用して出力されています。

以下のモジュールに属するロガーのログメッセージが出力されます。モジュールとロガーの対応については、[7.1.1. モジュールおよびロガーの一覧] を参照してください。

◇ jndi

◇ jndi-service

- **ログサイズ変更方法**

log4otx.xml の Appender 名が JNDISPFILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が JNDISPFILELOG の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げた各モジュールのモジュールログレベルを変更してください。設定方法に関しては、[4.1. 3.モジュールログレベルの設定方法] を参照してください。

## 3.7.2. webotx\_naming.log

- **概要**

JNDI の名前の登録、検索等の操作に関するログです。Log4otx を利用して出力されています。  
以下のモジュールに属するロガーのログメッセージが出力されます。モジュールとロガーの対応については、[7.1.1. モジュールおよびロガーの一覧] を参照してください。

◇ core

- **ログサイズ変更方法**

log4otx.xml の Appender 名が NAMINGFILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が NAMINGFILELOG の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げた各モジュールのモジュールログレベルを変更してください。設定方法に関しては、[4.1.3.モジュールログレベルの設定方法] を参照してください。

または、以下のロガーのログレベルを変更してください。設定方法に関しては、[4.1.7.Log4otx のログレベルの設定方法] を参照してください。

\*com.nec.webotx.enterprise.system.core.naming

## 3.8. Object Broker

### 3.8.1. objava.log

- **概要**

WebOTX Object Broker Java ライブラリに関するログです。Log4otx を利用して出力されています。

以下のモジュールに属するロガーのログメッセージが出力されます。モジュールとロガーの対応については、[7.1.1. モジュールおよびロガーの一覧] を参照してください。

◇ objectbrokerjava-lib

◇ objectbrokerjava-msg

- **ログサイズ変更方法**

log4otx.xml の Appender 名が OSPIJLIBFILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が OSPIJLIBFILELOG の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げた各モジュールのモジュールログレベルを変更してください。設定方法に関しては、[4.1.3.モジュールログレベルの設定方法] を参照してください。

### 3.8.2. webotx\_ospi.log

- **概要**

WebOTX Object Broker に関するログです。Log4otx を利用して出力されています。

以下のモジュールに属するロガーのログメッセージが出力されます。モジュールとロガーの対応については、[7.1.1. モジュールおよびロガーの一覧] を参照してください。

- ◇ oadj
- ◇ objectbroker

- **ログサイズ変更方法**

log4otx.xml の Appender 名が OSPIFILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が OSPIFILELOG の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げた各モジュールのモジュールログレベルを変更してください。設定方法に関しては、[4.1.3.モジュールログレベルの設定方法] を参照してください。

### 3.8.3. ObLog.log

- **概要**

oad、名前サーバ、IR サーバ、corbaloc サーバおよびキャッシュ名前サーバ以外の AP が出力する。Log4otx は利用せず、サービス独自の方法で出力しています。

- **ログサイズ変更方法**

ログファイルのサイズは、デフォルトでは、無制限です。以下のコマンドでファイルサイズを変更することができます。0 を設定するとファイルサイズの上限がなくなります。

```
otxadmin > set server.objectbrokerconfig.LogLimit=<サイズ(KB)>
```

なお、LogLimit の値は ObLog.log、Oad.log、InterfaceRepository.log、corbaloc.log、namesv.log、cnamesv.log、message.log で共通となります。

- **ローテーション設定方法**

上記コマンドで指定したファイルサイズを超えた場合、\_old という拡張子をつけて 1 世代分待避されます。世代数は設定できません。

- **ログレベル変更方法**

以下のコマンドでログレベルを設定することができます。既定値は ERROR です。

```
otxadmin > set server.objectbrokerconfig.LoggingLevel=<ログレベル>
```

設定できるログレベルは以下になります。

NONE、ERROR、WARNING、INFORMATION

### 3.8.4. Oad.log

- **概要**

oad が出力するログ。Log4otx は利用せず、サービス独自の方法で出力しています。

内容は ObLog.log と同じです。

- **ログサイズ変更方法**

ログファイルのサイズは、デフォルトでは、無制限です。以下のコマンドでファイルサイズを変更するこ

とができます。0を設定するとファイルサイズの上限がなくなります。

```
otxadmin > set server.objectbrokerconfig.LogLimit=<サイズ(KB)>
```

なお、LogLimit の値は ObLog.log、Oad.log、InterfaceRepository.log、corbaloc.log、namesv.log、cnamesv.log、message.log で共通となります。

- **ローテーション設定方法**

上記コマンドで指定したファイルサイズを超えた場合、\_old という拡張子をつけて 1 世代分待避されます。世代数は設定できません。

- **ログレベル変更方法**

以下のコマンドでログレベルを設定することができます。既定値は NONE です。

```
otxadmin > set server.objectbrokerservice.oad.OadLoggingLevel=<ログレベル>
```

設定できるログレベルは以下になります。

NONE、ERROR、WARNING、INFORMATION

NONE が設定されているときは LoggingLevel の設定値を使用(既定値)されます。

### 3.8.5. InterfaceRepository.log

- **概要**

IR サーバが出力するログ。Log4otx は利用せず、サービス独自の方法で出力しています。

内容は ObLog.log と同じです。

- **ログサイズ変更方法**

ログファイルのサイズは、デフォルトでは、無制限です。以下のコマンドでファイルサイズを変更することができます。0を設定するとファイルサイズの上限がなくなります。

```
otxadmin > set server.objectbrokerconfig.LogLimit=<サイズ(KB)>
```

なお、LogLimit の値は ObLog.log、Oad.log、InterfaceRepository.log、corbaloc.log、namesv.log、cnamesv.log、message.log で共通となります。

- **ローテーション設定方法**

上記コマンドで指定したファイルサイズを超えた場合、\_old という拡張子をつけて 1 世代分待避されます。世代数は設定できません。

- **ログレベル変更方法**

以下のコマンドでログレベルを設定することができます。既定値は NONE です。

```
otxadmin > set server.objectbrokerservice.irsv.InterfaceRepositoryLoggingLevel =<ログレベル>
```

設定できるログレベルは以下になります。

NONE、ERROR、WARNING、INFORMATION

NONE が設定されているときは LoggingLevel の設定値を使用(既定値)されます。

### 3.8.6. corbaloc.log

- **概要**

corbaloc サーバが出力するログ。Log4otx は利用せず、サービス独自の方法で出力しています。

内容は ObLog.log と同じ。

- **ログサイズ変更方法**

ログファイルのサイズは、デフォルトでは、無制限です。以下のコマンドでファイルサイズを変更することができます。0を設定するとファイルサイズの上限がなくなります。

```
otxadmin > set server.objectbrokerconfig.LogLimit=<サイズ(KB)>
```

なお、LogLimit の値は ObLog.log、Oad.log、InterfaceRepository.log、corbaloc.log、namesv.log、cnamesv.log、message.log で共通となります。

- **ローテーション設定方法**

上記コマンドで指定したファイルサイズを超えた場合、\_old という拡張子をつけて 1 世代分待避されます。世代数は設定できません。

- **ログレベル変更方法**

以下のコマンドでログレベルを設定することができます。既定値は NONE です。

```
otxadmin > set server.objectbrokerservice.corbaloc.CorbalocLoggingLevel=<ログレベル>
```

設定できるログレベルは以下になります。

NONE、ERROR、WARNING、INFORMATION

NONE が設定されているときは LoggingLevel の設定値を使用(既定値)されます。

### 3.8.7. namesv.log

- **概要**

名前サーバが出力するログ。Log4otx は利用せず、サービス独自の方法で出力しています。

名前サーバのエラー情報と ObLog.log と同じ内容が出力される。

- **ログサイズ変更方法**

ログファイルのサイズは、デフォルトでは、無制限です。以下のコマンドでファイルサイズを変更することができます。0を設定するとファイルサイズの上限がなくなります。

```
otxadmin > set server.objectbrokerconfig.LogLimit=<サイズ(KB)>
```

なお、LogLimit の値は ObLog.log、Oad.log、InterfaceRepository.log、corbaloc.log、namesv.log、cnamesv.log、message.log で共通となります。

- **ローテーション設定方法**

上記コマンドで指定したファイルサイズを超えた場合、\_old という拡張子をつけて 1 世代分待避されます。世代数は設定できません。

- **ログレベル変更方法**

以下のコマンドでログレベルを設定することができます。既定値は NONE です。

```
otxadmin > set server.objectbrokerservice.namesv.NameServiceLoggingLevel=<ログレベル>
```

>

設定できるログレベルは以下になります。

NONE、ERROR、WARNING、INFORMATION

NONE が設定されているときは LoggingLevel の設定値を使用(既定値)されます。

### 3.8.8. cnamesv.log

- **概要**

キャッシュ名前サーバが出力するログ。Log4otx は利用せず、サービス独自の方法で出力していません。

キャッシュ名前サーバのエラー情報と ObLog.log と同じ内容が出力される。

- **ログサイズ変更方法**

ログファイルのサイズは、デフォルトでは、無制限です。以下のコマンドでファイルサイズを変更することができます。0 を設定するとファイルサイズの上限がなくなります。

```
otxadmin > set server.objectbrokerconfig.LogLimit=<サイズ(KB)>
```

なお、LogLimit の値は ObLog.log、Oad.log、InterfaceRepository.log、corbaloc.log、namesv.log、cnamesv.log、message.log で共通となります。

- **ローテーション設定方法**

上記コマンドで指定したファイルサイズを超えた場合、\_old という拡張子をつけて 1 世代分待避されます。世代数は設定できません。

- **ログレベル変更方法**

以下のコマンドでログレベルを設定することができます。既定値は NONE です。

```
otxadmin > set server.objectbrokerservice.cnamesv.CacheNameServiceLoggingLevel=<ログレベル>
```

設定できるログレベルは以下になります。

NONE、ERROR、WARNING、INFORMATION

NONE が設定されているときは LoggingLevel の設定値を使用(既定値)されます。

### 3.8.9. message.log

- **概要**

通信毎に通信内容、通信相手、通信量を出力するログ。Log4otx は利用せず、サービス独自の方法で出力しています。

Object Broker C++ライブラリを使用する全アプリケーションが出力。デフォルト設定では出力されません。

- **ログサイズ変更方法**

ログファイルのサイズは、デフォルトでは、無制限です。以下のコマンドでファイルサイズを変更することができます。0 を設定するとファイルサイズの上限がなくなります。

```
otxadmin > set server.objectbrokerconfig.LogLimit=<サイズ(KB)>
```

なお、LogLimit の値は ObLog.log、Oad.log、InterfaceRepository.log、corbaloc.log、namesv.log、cnamesv.log、message.log で共通となります。

- **ローテーション設定方法**

上記コマンドで指定したファイルサイズを超えた場合、\_old という拡張子をつけて 1 世代分待避されます。世代数は設定できません。

- **ログレベル変更方法**

ログレベルはありませんが、出力されるログの内容を変更することができます。

・ログを出力する場合

```
otxadmin > server.objectbrokerconfig.MessagingLog=true
```

・通信内容を message.log に出力させる場合

```
otxadmin > server.objectbrokerconfig.MessageBodyLog=true
```

### 3.8.10. syslog.log

- **概要**

シスログ(Windows のみ)。デフォルトの設定ではログは出力されません。ただし、oad、名前サーバ等の Object Broker サービスのデーモン類については、この設定にかかわらず、syslog に起動時と停止時のメッセージが常に出力されます。Log4otx は利用せず、サービス独自の方法で出力しています。

- **ログサイズ変更方法**

ログファイルのサイズは、デフォルトでは、無制限です。以下のコマンドでファイルサイズを変更することができます。0 を設定するとファイルサイズの上限がなくなります。

```
otxadmin > set server.objectbrokerconfig.LogLimit=<サイズ(KB)>
```

なお、LogLimit の値は ObLog.log、Oad.log、InterfaceRepository.log、corbaloc.log、namesv.log、cnamesv.log、message.log で共通となります。

- **ローテーション設定方法**

上記コマンドで指定したファイルサイズを超えた場合、\_old という拡張子をつけて 1 世代分待避されます。世代数は設定できません。

- **ログレベル変更方法**

ログレベルはありませんが、出力されるログの内容を変更することができます。

・ログを出力する場合

```
otxadmin > server.objectbrokerconfig.Syslog=true
```

・システム運用者向けのログを出力する場合

```
otxadmin > server.objectbrokerconfig.SyslogForOperator=true
```

・アプリケーション開発者向けのログを出力する場合

```
otxadmin > server.objectbrokerconfig.SyslogForDeveloper=true
```

・システム運用者向け、アプリケーション開発者向けの詳細情報があれば、追加で出力する場合

```
otxadmin > server.objectbrokerconfig.SyslogDetail=true
```

### 3.8.11. oadj.log

- **概要**

OadJ の運用時のログを出力。Log4otx は利用せず、サービス独自の方法で出力しています。必要に応じてログファイルを作成します。

- **ログサイズ変更方法**

以下のコマンドでファイルサイズを変更することができます。デフォルトで 100KB が設定されています。

```
otxadmin > set server.objectbrokerservice.oadj.LogFileSize=<サイズ(KB)>
```

- **ローテーション設定方法**

LogFileSize に設定されたファイルサイズを超えた場合、.bak という拡張子をつけて 1 世代分ログが待避されます。待避する世代数を変更することはできません。

- **ログレベル変更方法**

以下のコマンドでログレベルを設定することができます。既定値は ERROR です。

```
otxadmin > set server.objectbrokerservice.oadj.LogLevel=<ログレベル>
```

設定できるログレベルは以下になります。

NO、ERROR、WARN、INFO、DEBUG

## 3.9. TPモニタ

### 3.9.1. webotx\_tpmmgr.log

- **概要**

TP モニタマネージャにおける TP モニタ連携マネージャ)に関するログです。Log4otx を利用して出力しています。

運用アシスタント機能に関するログを出力します。

以下のモジュールに属するロガーのログメッセージが出力されます。モジュールとロガーの対応については、[7.1.1. モジュールおよびロガーの一覧] を参照してください。

- ◇ admin-assistant
- ◇ admListener
- ◇ tpmonitormanager

- **ログサイズ変更方法**

log4otx.xml の Appender 名が TPMMGRFILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が TPMMGRFILELOG の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げた各モジュールのモジュールログレベルを変更してください。設定方法に関しては、[4.1.3.モジュールログレベルの設定方法] を参照してください。

### 3.9.2. \${PGNAME}.\${PID}.log

- **概要**

サーバアプリケーションのトレースログ。Log4otx は利用せず、サービス独自の方法で出力しています。

サーバアプリケーションのトレース情報を出力します。

- **ログサイズ変更方法**

以下のコマンドでログサイズを変更することができます。デフォルト値は 32KB となっています。

```
otxadmin > set tpsystem.applicationGroups.${APGNAME}.processGroups.${PGNAME}.traceFileMaxSize=<サイズ(KB)>
```

- **ローテーション設定方法**

トレースファイルはトレースファイルの最大サイズに達すると、ファイル名に ".bak" を付加してバックアップファイルを作成します。そして、新規にトレースファイルを作成します。1 世代のみで古い bak ファイルは削除されます。プロセスグループが再起動した場合は、<time>が付加され save ディレクトリに退避されます。save ディレクトリ内のファイルは削除されませんので、ディスクフルを回避するために、定期的に手動で削除する必要があります。

- **ログレベル変更方法**

```
otxadmin > set tpsystem.applicationGroups.${APGNAME}.processGroups.${PGNAME}.traceLevel=<レベル>
```

設定できるレベルは以下になります。

- 0 - 0.パニック状態
- 1 - 1.即時に訂正すべきである状態
- 2 - 2.危急の状態
- 3 - 3.一般的なエラー
- 4 - 4.警告メッセージ
- 5 - 5.通知メッセージ
- 6 - 6.情報メッセージ
- 7 - 7.デバッグ時のメッセージ
- 1 - -1.採取しない

### 3.9.3. \${PGNAME}\_sys.\${PID}.log

- **概要**

システムトレースファイル。Log4otx は利用せず、サービス独自の方法で出力しています。

プロセス起動時の情報や CORBA アプリケーションで検出したエラーの情報、Java アプリケーションでスタック情報採取を行ったときに出力します。

- **ログサイズ変更方法**

ログサイズを変更することはできません。デフォルト値の 1MB となります。

- **ローテーション設定方法**

ローテーションされません。プロセスグループ停止時に<time>が付加され save ディレクトリに退避されます。save ディレクトリ内のファイルは削除されませんので、ディスクフルを回避するために、定期的に手動で削除する必要があります。

- **ログレベル変更方法**

```
otxadmin > set tpsystem.applicationGroups.${APGNAME}.processGroups.${PGNAME}.traceLevel=<レベル>
```

設定できるレベルは以下になります。

- 0 - 0.パニック状態
- 1 - 1.即時に訂正すべきである状態
- 2 - 2.危急の状態
- 3 - 3.一般的なエラー
- 4 - 4.警告メッセージ
- 5 - 5.通知メッセージ
- 6 - 6.情報メッセージ
- 7 - 7.デバッグ時のメッセージ
- 1 - -1.採取しない

### 3.9.4. \${PGNAME}\_recovery.\${PID}.log

- **概要**

プロセス異常終了によりプロセスグループが停止した場合に出力します。Log4otx は利用せず、サービス独自の方法で出力しています。

- **ログサイズ変更方法**

ファイルサイズ上限は 1MB です。1MB を超える出力は通常ありません。

- **ローテーション設定方法**

ローテーションできません。プロセスグループ停止時に<time>が付加され save ディレクトリに退避されます。save ディレクトリ内のファイルは削除されませんので、ディスクフルを回避するために、定期的に手動で削除する必要があります。

- **ログレベル変更方法**

```
otxadmin > set tpsystem.applicationGroups.${APGNAME}.processGroups.${PGNAME}.traceLevel=<レベル>
```

設定できるレベルは以下になります。

- 0 - 0.パニック状態
- 1 - 1.即時に訂正すべきである状態
- 2 - 2.危急の状態
- 3 - 3.一般的なエラー
- 4 - 4.警告メッセージ
- 5 - 5.通知メッセージ
- 6 - 6.情報メッセージ
- 7 - 7.デバッグ時のメッセージ
- 1 - -1.採取しない

### 3.9.5. \${APGNAME}\_EXIT\_sys.\${PID}.log

- **概要**

アプリケーション初期プロセスのシステムトレースファイル。Log4otx は利用せず、サービス独自の方法で出力しています。

アプリケーション初期プロセスを使用する場合のみ、プロセス起動時の情報やエラーの情報が出力されます。

- **ログサイズ変更方法**

ファイルサイズ上限は 1MB です。1MB を超える出力は通常ありません。

- **ローテーション設定方法**

ローテーションできません。プロセスグループ停止時に<time>が付加され save ディレクトリに退避されます。save ディレクトリ内のファイルは削除されませんので、ディスクフルを回避するために、定期的に手動で削除する必要があります。

- **ログレベル変更方法**

```
otxadmin > set tpsystem.applicationGroups.${APGNAME}.processGroups.${PGNAME}.tr
```

aceLevel=<レベル>

設定できるレベルは以下になります。

- 0 - 0.パニック状態
- 1 - 1.即時に訂正すべきである状態
- 2 - 2.危急の状態
- 3 - 3.一般的なエラー
- 4 - 4.警告メッセージ
- 5 - 5.通知メッセージ
- 6 - 6.情報メッセージ
- 7 - 7.デバッグ時のメッセージ
- 1 - -1.採取しない

### 3.9.6. \${SYSTEMNAME}\_IIOPLsn.log \${SYSTEMNAME}\_A\_IIOPLsn.log

- **概要**

IIOP リスナ内部トレースファイル。Log4otx は利用せず、サービス独自の方法で出力しています。

- **ログサイズ変更方法**

\${INSTANCE\_ROOT}/config/tpsystem/IIOPLsn.sg.sv を IIOPLsn.sg にファイル名を変更し、以下の定義を追加することでログサイズを変更することができます。以下の定義がされていない、または IIOPLsn.sg ファイルがない場合、デフォルトで 32KB となります。

TraceSize <サイズ(バイト)>

- **ローテーション設定方法**

ログファイルのサイズが設定値(デフォルト 32KB)を超えると新しいログファイルが作成されます。古いログファイルは<ログファイル名.bak>として保存されます。1 世代のみバックアップされます。

- **ログレベル変更方法**

ログレベルはありません。

### 3.9.7. \${SYSTEMNAME}\_woejout<number>.log

- **概要**

イベントジャーナルの採取をユーザが実行した場合のみ作成されます。Log4otx は利用せず、サービス独自の方法で出力しています。

- **ログサイズ変更方法**

変更できません。

- **ローテーション設定方法**

変更できません。

- **ログレベル変更方法**

ログレベルはありません。

## 3.10. Transaction サービス

### 3.10.1. webotx\_ts.log

- **概要**

WebOTX Transaction サービスに関するログです。Log4otx を利用して出力されています。

以下のモジュールに属するロガーのログメッセージが出力されます。モジュールとロガーの対応については、[7.1.1. モジュールおよびロガーの一覧] を参照してください。

◇ transactionservice

- **ログサイズ変更方法**

log4otx.xml の Appender 名が TSFILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が TSFILELOG の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げた各モジュールのモジュールログレベルを変更してください。設定方法に関しては、[4.1.3.モジュールログレベルの設定方法] を参照してください

### 3.10.2. RCS.trc

- **概要**

Transaction サービス関連のログ。Transaction 復旧用プロセス(RCS)のログ。Log4otx は利用せず、サービス独自の方法で出力しています。

- **ログサイズ変更方法**

以下のコマンドで変更することができます。デフォルト値は 1024KB となっています。なお、以下の値は、RCS.trc、LRS.trc 両方で使用されます。

```
otxadmin > set server.transactionservice.trace-size=<サイズ(KB)>
```

- **ローテーション設定方法**

最大サイズを超えると、常にも上書きされます。バックアップを残すことはできません。

- **ログレベル変更方法**

以下のコマンドで設定することができます。既定値は 2 です。

```
otxadmin > set server.transactionservice.trace-level=<レベル>
```

レベルは 1(エラーレベル)~5(デバッグレベル)が設定できます。

### 3.10.3. LRS.trc

- **概要**

Transaction サービス関連のログ。AP 内での Transaction 動作ログ。Log4otx は利用せず、サービス独自の方法で出力しています。

- **ログサイズ変更方法**

以下のコマンドで変更することができます。デフォルト値は 1024KB となっています。なお、以下の値は、RCS.trc、LRS.trc 両方で使用されます。

```
otxadmin > set server.transactionservice.trace-size=<サイズ(KB)>
```

- **ローテーション設定方法**

最大サイズを超えると、常にも上書きされます。バックアップを残すことはできません。

- **ログレベル変更方法**

以下のコマンドで設定することができます。既定値は 2 です。

otxadmin > set server.transactionservice.trace-level=<レベル>

レベルは 1(エラーレベル)~5(デバッグレベル)が設定できます。

### 3.10.4. webotx\_ts\_cpp.log

- **概要**

Transaction サービス関連のログ。Transaction 復旧用プロセス(RCS(C++AP 用))。Log4otx は利用せず、サービス独自の方法で出力しています。

- **ログサイズ変更方法**

起動時と終了時のログのみで、大量に出力されることはありません。また、再起動時に上書きされるため、ログサイズを変更する必要はありません。

- **ローテーション設定方法**

再起動時には上書きされるため、サイズが無尽蔵に増えることはありません。これよりローテーションを行う必要はありません。

- **ログレベル変更方法**

ログレベルはありません。

## 3.11. Webコンテナ

### 3.11.1. webotx\_catalina.log

- **概要**

Web コンテナの基盤部分の動作ログです。Log4otx を利用して出力されています。

以下のロガーのログメッセージが出力されます。ロガーについては、[7.1.1. モジュールおよびロガーの一覧] を参照してください。

◇ org.apache.catalina

◇ tomcat

- **ログサイズ変更方法**

log4otx.xml の Appender 名が CATALINAFILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が CATALINAFILELOG の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げたロガーのログレベルを変更してください。設定方法に関しては、[4.1.7.Log4otx のログレベルの設定方法] を参照してください。

### 3.11.2. webotx\_webcont.log

- **概要**

Webコンテナと接続するドメインの基幹部分の動作ログです。Log4otxを利用して出力されています。

以下のモジュールに属するロガーのログメッセージが出力されます。モジュールとロガーの対応については、[7.1.1. モジュールおよびロガーの一覧] を参照してください。

◇ web-container

- **ログサイズ変更方法**

log4otx.xml の Appender 名が WEBCONTAINERFILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が WEBCONTAINERFILELOG の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げた各モジュールのモジュールログレベルを変更してください。設定方法に関しては、[4.1.3.モジュールログレベルの設定方法] を参照してください。

### 3.11.3. webotx\_container.log

- **概要**

Web コンテナと接続するドメインの基幹部分の動作ログです。Log4otx を利用して出力されています。以下のモジュールに属するロガーのログメッセージが出力されます。モジュールとロガーの対応については、[7.1.1. モジュールおよびロガーの一覧] を参照してください。

- ◇ appclient-container
- ◇ ejb-container
- ◇ web-container

- **ログサイズ変更方法**

log4otx.xml の Appender 名が CONTAINERFILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が CONTAINERFILELOG の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **対応モジュールログレベル**

概要で挙げた各モジュールのモジュールログレベルを変更してください。設定方法に関しては、[4.1.3.モジュールログレベルの設定方法] を参照してください。

### 3.11.4. webotx\_webc\_message.log

- **概要**

Web コンテナが通知する各種メッセージログ。Web コンテナ、および Web アプリケーションの警告や障害、運用管理コンソールのオペレーション結果などを出力です。Log4otx を利用して出力されています。

以下のロガーのログメッセージが出力されます。ロガーについては、[7.1.1. モジュールおよびロガーの一覧] を参照してください。

- ◇ com.nec.webotx.webcontainer.core.MessageNotification

- **ログサイズ変更方法**

log4otx.xml の Appender 名が WEBCMSGFILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が WEBCMSGFILELOG の MaxBackupIndex 値を変更することで、

ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げたロガーのログレベルを変更してください。設定方法に関しては、[4.1.7.Log4otx のログレベルの設定方法] を参照してください。

### 3.11.5. webotx\_webc\_stat\_vm.log

- **概要**

WebOTXドメインのエージェント Java VM の統計ログ(メモリ情報)。運用管理コンソールで「Web Application の統計情報」の採取を指定した場合に出力です。Log4otx を利用して出力されています。以下のロガーのログメッセージが出力されます。ロガーについては、[7.1.1. モジュールおよびロガーの一覧] を参照してください。

◇ com.nec.webotx.webcontainer.statistics.JavaVMStatistics

- **ログサイズ変更方法**

log4otx.xml の Appender 名が WEBCSTATVMFILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が WEBCSTATVMFILELOG の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げたロガーのログレベルを変更してください。設定方法に関しては、[4.1.7.Log4otx のログレベルの設定方法] を参照してください。

### 3.11.6. webotx\_webc\_stat\_webap.log

- **概要**

Web アプリケーションの統計情報。運用管理コンソールで統計を採取し、「JavaVM メモリ情報ログ取得」も ON にした場合に出力です。Log4otx を利用して出力されています。

以下のロガーのログメッセージが出力されます。ロガーについては、[7.1.1. モジュールおよびロガーの一覧] を参照してください。

◇ com.nec.webotx.webcontainer.statistics.WebAPtstatistics

- **ログサイズ変更方法**

log4otx.xml の Appender 名が WEBCSTATAPFILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が WEBCSTATAPFILELOG の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げたロガーのログレベルを変更してください。設定方法に関しては、[4.1.7.Log4otx のログレベルの設定方法] を参照してください。

### 3.11.7. webotx\_struts.log

- **概要**

Struts パッケージの動作ログです。Log4otx を利用して出力されています。

Web 版運用管理コンソールに関するログが出力されます。

以下のロガーのログメッセージが出力されます。ロガーについては、[7.1.1. モジュールおよびロガーの一覧] を参照してください。

◇ org.apache.struts

- **ログサイズ変更方法**

log4otx.xml の Appender 名が STRUTSFILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が STRUTSFILELOG の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げたロガーのログレベルを変更してください。設定方法に関しては、[4.1.7.Log4otx のログレベルの設定方法] を参照してください。

### 3.11.8. mod\_jk.log

- **概要**

WebOTX Web Server 1.3 や Apache HTTP Server 1.3 と連携している時に、Web サーバに組み込んだプラグインモジュールの動作トレースや障害内容を出力します。Log4otx は利用せず、サービス独自の方法で出力しています。

- **ログサイズ変更方法**

rotatelogs を利用した場合は任意に指定可能です。詳しくは[4.2.3 mod\_jk のログのローテーション] を参照してください。

- **ローテーション設定方法**

rotatelogs を利用した場合は任意に指定可能です。%d などで日付の利用が可能です。省略時はマシン起動からの経過時間が付加されます。詳しくは[4.2.3 mod\_jk のログのローテーション]を参照してください。

- **ログレベル変更方法**

`\${INSTANCE\_ROOT}/config/webcontainer/mod\_jk.conf-auto` の JkLogLevel の値で指定することができます。既定値は error です。info,error,debug の値が指定可能です。

### 3.11.9. mod\_jk\_20.log

- **概要**

WebOTX Web Server 2.0 や Apache HTTP Server 2.0 と連携している時に、Web サーバに組み込んだプラグインモジュールの動作トレースや障害内容を出力します。Log4otx は利用せず、サービス独自の方法で出力しています。

- **ログサイズ変更方法**

rotatelogs を利用した場合は任意に指定可能です。詳しくは[4.2.3 mod\_jk のログのローテーション]

を参照してください。

- **ローテーション設定方法**

rotatelogs を利用した場合は任意に指定可能です。%d など で日付の利用が可能です。省略時はマシン起動からの経過時間が付加されます。詳しくは[4.2.3 mod\_jk のログのローテーション]を参照してください。

- **ログレベル変更方法**

`${INSTANCE_ROOT}/config/webcontainer/mod_jk-20.conf-auto` の `JkLogLevel` の値で指定することができます。既定値は `error` です。info、error、debug の値が指定可能です。

### 3.11.10. isapi.log

- **概要**

IIS と連携している時に、Web サーバに組み込んだプラグインモジュールの動作トレースや障害内容を出力。Log4otx は利用せず、サービス独自の方法で出力しています。

- **ログサイズ変更方法**

ログサイズを変更することはできません。ファイルサイズは無制限となり、OS 依存となります。

- **ローテーション設定方法**

ローテーションを行うことはできません。

- **ログレベル変更方法**

`${INSTANCE_ROOT}/config/WebCont/iis_redirect.reg` の `log_level` の定義を変更してください。

```
log_level="<ログレベル>"
```

設定できるレベルは `error`、`warn`、`info`、`debug`、`trace` のいずれかとなります。

設定変更後、`iis_redirect.reg` をダブルクリックし、レジストリに保存します。

設定を反映させるために、IIS を再起動する必要があります。

### 3.11.11. nsapi.log

- **概要**

Sun Java System Web Server (Sun ONE) と連携している時に、Web サーバに組み込んだプラグインモジュールの動作トレースや障害内容を出力します。Log4otx は利用せず、サービス独自の方法で出力しています。

- **ログサイズ変更方法**

ログサイズを変更することはできません。ファイルサイズは無制限となり、OS 依存となります。

- **ローテーション設定方法**

ローテーションを行うことはできません。

- **ログレベル変更方法**

以下のファイルを編集してください。

```
<SunJavaWebServer_InstallDir>/https-ホスト名/config/magnus.conf
```

```
Init fn="jk_init" worker_file="${INSTANCE_ROOT}/config/WebCont/workers.properties"
```

```
worker_mount_file="${INSTANCE_ROOT}/config/WebCont/uriworkermapNS.properties-a
```

```
uto" log_level="<ログレベル>"
```

```
log_file="${INSTANCE_ROOT}/logs/webcontainer/nsapi.log"
```

設定できるログレベルは error、warn、info、debug、trace のいずれかとなります。

設定を反映させるために、Sun Web Server を再起動する必要があります。

### 3.11.12. mod\_jk\_om.log

- **概要**

マルチプロセスモードで WebOTX Web Server 1.3 や Apache HTTP Server 1.3 と連携している時に、Web サーバに組み込んだプラグインモジュールの動作トレースや障害内容を出力します。Log 4otx は利用せず、サービス独自の方法で出力しています。

- **ログサイズ変更方法**

rotatelogs を利用した場合は任意に指定可能です。詳しくは[4.2.3 mod\_jk のログのローテーション]を参照してください。

- **ローテーション設定方法**

rotatelogs を利用した場合は任意に指定可能です。%d などで日付の利用が可能です。省略時はマシン起動からの経過時間が付加されます。詳しくは[4.2.3 mod\_jk のログのローテーション]を参照してください。

- **ログレベル変更方法**

`${INSTANCE_ROOT}/config/webcontainer/mod_jk_om.conf-auto` の JkLogLevel の値で指定することができます。既定値は error です。info、error、debug の値が指定可能です。

### 3.11.13. mod\_jk\_om-20.log

- **概要**

マルチプロセスモードで、WebOTX Web Server 2.0 や Apache HTTP Server 2.0 と連携している時に、Web サーバに組み込んだプラグインモジュールの動作トレースや障害内容を出力します。Log 4otx は利用せず、サービス独自の方法で出力しています。

- **ログサイズ変更方法**

rotatelogs を利用した場合は任意に指定可能です。詳しくは[4.2.3 mod\_jk のログのローテーション]を参照してください。

- **ローテーション設定方法**

rotatelogs を利用した場合は任意に指定可能です。%d などで日付の利用が可能です。省略時はマシン起動からの経過時間が付加されます。詳しくは[4.2.3 mod\_jk のログのローテーション]を参照してください。

- **ログレベル変更方法**

`${INSTANCE_ROOT}/config/webcontainer/mod_jk_om-20.conf-auto` の JkLogLevel の値で指定することができます。既定値は error です。info、error、debug の値が指定可能です。

## 3.12. Webサーバ

### 3.12.1. webotx\_websv.log

- **概要**

WebOTX Web サーバのライフサイクルに関するログです。Log4otx を利用して出力しています。

以下のモジュールに属するロガーのログメッセージが出力されます。モジュールとロガーの対応については、[7.1.1. モジュールおよびロガーの一覧] を参照してください。

- **ログサイズ変更方法**

log4otx.xml の Appender 名が WEBSVFILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が WEBSVFILELOG の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げた各モジュールのモジュールログレベルを変更してください。設定方法に関しては、[4.1.3.モジュールログレベルの設定方法] を参照してください。

### 3.12.2. access\_log(access.log)

- **概要**

ブラウザのアクセス情報を出力するアクセスログです。Log4otx は利用せず、サービス独自の方法で出力しています。

1 行で 1 リクエストに対応します。

- **ログサイズ変更方法**

rotatelogs を利用した場合は任意に指定可能です。詳しくは[4.2.2 Web サーバのログのローテーション]を参照してください。

- **ローテーション設定方法**

rotatelogs を利用した場合は任意に指定可能です。%d などで日付の利用が可能です。省略時はマシン起動からの経過時間が付加されます。詳しくは[4.2.2 Web サーバのログのローテーション]を参照してください。

- **ログレベル変更方法**

ログレベルはありません。

### 3.12.3. error\_log(error.log)

- **概要**

WebOTX Web サーバのエラー情報を出力するエラーログです。Log4otx は利用せず、サービス独自の方法で出力しています。

WebOTX Web サーバで発生したエラー情報等を出力します。

- **ログサイズ変更方法**

rotatelogs を利用した場合は任意に指定可能です。詳しくは[4.2.2 Web サーバのログのローテーション]を参照してください。

- **ローテーション設定方法**

rotatelogs を利用した場合は任意に指定可能です。%d などで日付の利用が可能です。省略時はマシン起動からの経過時間が付加されます。詳しくは[4.2.2 Web サーバのログのローテーション]を参照してください。

- **ログレベル変更方法**

`#{INSTANCE_ROOT}/config/WebServer/httpd.conf` の以下の定義を編集し、Web サーバを再起

動してください。

LogLevel <レベル>

レベルには以下のいずれかの値を指定できます。既定値は warn です。

debug、info、notice、warn、error、crit、alert、emerg

### 3.12.4. ssl\_request\_log(ssl\_request.log)

- **概要**

HTTPS 通信時のブラウザのアクセス情報と、SSL プロトコルのバージョンと暗号化アルゴリズムの情報を出力するアクセスログです。Log4otx は利用せず、サービス独自の方法で出力しています。

1 行で1リクエストに対応します。

- **ログサイズ変更方法**

rotatelogs を利用した場合は任意に指定可能です。詳しくは[4.2.2 Web サーバのログのローテーション]を参照してください。

- **ローテーション設定方法**

rotatelogs を利用した場合は任意に指定可能です。%d などで日付の利用が可能です。省略時はマシン起動からの経過時間が付加されます。詳しくは[4.2.2 Web サーバのログのローテーション]を参照してください。

- **ログレベル変更方法**

ログレベルはありません。

### 3.12.5. server\_access.log

- **概要**

Web コンテナのアクセスログです。リクエストの状況や結果が記録されます。Log4otx を利用して出力しています。

以下のロガーのログメッセージが出力されます。ロガーについては、[7.1.1. モジュールおよびロガーの一覧] を参照してください。

◇ \_\_otxadmin

- **ログサイズ変更方法**

log4otx.xml の Appender 名が ACCESSFILELOGSERVER の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が ACCESSFILELOGSERVER の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げたロガーのログレベルを変更してください。設定方法に関しては、[4.1.7.Log4otx のログレベルの設定方法] を参照してください。

## 3.13. Web Service

### 3.13.1. webservice.log

- **概要**

WebOTX Web サービスに関するログです。Log4otx を利用して出力しています。

以下のモジュールに属するロガーのログメッセージが出力されます。モジュールとロガーの対応については、[7.1.1. モジュールおよびロガーの一覧] を参照してください。

◇ jaxws

- **ログサイズ変更方法**

log4otx.xml の Appender 名が WSFILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が WSFILELOG の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げたモジュールのログレベルを変更してください。設定方法に関しては、[4.1.3.モジュールログレベルの設定方法] を参照してください。

### 3.13.2. wsrn.log

- **概要**

Web サービスの高信頼メッセージング機能に関するログです。Log4otx を利用して出力しています。

以下のロガーのログメッセージが出力されます。ロガーについては、[7.1.1. モジュールおよびロガーの一覧] を参照してください。

◇ com.nec.webotx.webservice.wsrn

- **ログサイズ変更方法**

log4otx.xml の Appender 名が WSRMFILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が WSRMFILELOG の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げたロガーのログレベルを変更してください。設定方法に関しては、[4.1.7.Log4otx のログレベルの設定方法] を参照してください。

## 3.14. Working Domain Coordinator

### 3.14.1. wowdc.log

- **概要**

WebOTX Working Domain Coordinator に関するログです。Log4otx を利用して出力しています。

以下のロガーのログメッセージが出力されます。ロガーについては、[7.1.1. モジュールおよびロガーの一覧] を参照してください。

◇ com.nec.webotx.enterprise.system.wdc

- **ログサイズ変更方法**

log4otx.xml の Appender 名が WDCFILELOG の MaxFileSize 値を変更することで最大ログファイルサイズを変更することができます。デフォルトでは 1MB です。

- **ローテーション設定方法**

log4otx.xml の Appender 名が WDCFILELOG の MaxBackupIndex 値を変更することで、ログファイル世代数を設定できます。デフォルトでは過去 2 世代分バックアップします。

- **ログレベル変更方法**

概要で挙げたロガーのログレベルを変更してください。設定方法に関しては、[4.1.7.Log4otx のログレベルの設定方法] を参照してください。

# 4. 設定値と設定方法について

## 4.1. ログレベルについて

### 4.1.1. WebOTX ASのログレベル

WebOTX Application Server では、ログレベルとして「モジュールログレベル」、「トレースログレベル」、「Log4otx ログレベル」の3つが存在します。これらのログレベルは、利用する Edition によって使用する対象が異なり、また各々の設定による適用範囲も異なるため、ログレベル変更時の目的によって使い分けする必要があります。

以下に、各ログレベルとエディションとの関係を示します。

ログレベル	主な利用目的	Edition 名			
		Web	Standard-J	Standard	Enterprise
モジュールログレベル	運用中の出力レベル変更	○	○	○ ◎	○ ◎
トレースログレベル	プロセスグループ(アプリケーション実行プロセス)に対する運用中の出力レベル一括変更	-	-	◎	◎
Log4otx ログレベル	ロガーに対する出力レベルの初期値を指定	○	○	○ ◎	○ ◎

○ : エージェントプロセスに有効な設定

◎ : プロセスグループに有効な設定

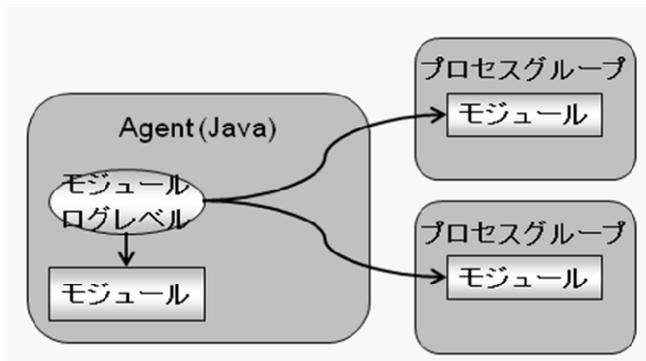
— : 適用の対象外

Web/Standard-J Edition では「モジュールログレベル」と「Log4otx ログレベル」を、Standard/Enterprise Edition ではさらに「トレースログレベル」を使用することでログの出力レベルを設定します。

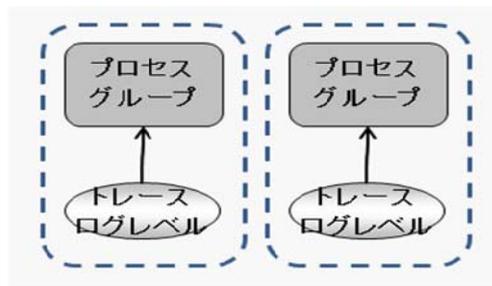
また、WebOTX のロガーは Log4j を拡張した Log4otx で構成されます。モジュールログレベルとトレースログレベルについては、各々が対応する Log4otx のレベルに内部で変換することでロガーに設定されます。

以降では、各ログレベルの利用目的について解説します。詳細は以降の各ログレベルに関する節を参照してください。

- **モジュールログレベル (実行モジュール単位での出力レベル)**  
Java Logging API をベースに、各ロガーを実行モジュール単位にまとめた際のログレベルです。詳細な動作確認や障害調査などの目的で、該当モジュールのみログレベルを変更したい場合に使用します。  
また、Standard / Enterprise Edition では、運用中のモジュールログレベルの設定変更が、全てのプロセスグループのロガーに対しても一括して反映されます。



- **トレースログレベル (プロセスグループ単位での出力レベル)**  
 プロセスグループ毎のロガーのセットに対して割り当てられたログレベルをトレースログレベルと呼びます。特定の業務アプリケーションに対する詳細な動作確認や障害調査などの目的で、該当のプロセスグループのみログレベルを変更したい場合に使用します。  
 トレースログレベルの設定は、プロセスグループ内の全てのロガーに適用されます。



- **Log4otx ログレベル (ログレベルの初期値)**  
 Log4jを基に、WebOTXが独自の拡張を施したログレベルです。WebOTXが内部で使用するLog4otxの定義ファイルにおいて使用されます。各ロガーのログレベルに対して、その初期値を決定する役割を持ちます。  
 ただし、プロセスグループにおいては、この設定値はトレースログレベルの設定により更新される可能性があります。詳しくは、「4.1.9 ログレベルの適用順序」を参照してください。

#### 4.1.2. 各種ログレベルの対応関係について

前節で示した各種ログレベルと、各レベルの対応関係は次のとおりです。

モジュールログレベル (値)	トレースログ レベル	log4otx ログレベル (値)
ALL (INT_MIN)	--	ALL (0)
SEVERE (1000)	0.パニック状態	FATAL (50000)
	1.即時に訂正すべきである状態	
	2.危急の状態	
	3.一般的なエラー	ERROR (40000)

WARNING (900)	4.警告メッセージ	WARN (30000)
SLOGINFO(*) (850)	--	SLOGINFO(*) (25000)
INFO (800)	5.通知メッセージ	INFO (20000)
CONFIG (700)	6.情報メッセージ	CONFIG(*) (10000)
FINE (500)	--	DEBUG (10000)
FINER (400)	--	DETAIL(*) (8000)
FINEST (300)	7.デバッグ時のメッセージ	TRACE (5000)
OFF (INT_MAX)	-1.採取しない	OFF (INT_MAX)

(\*)1) WebOTX 独自のレベルです。log4j については” com.nec.webotx.logging.OTXLogLevel”クラスで拡張、モジュールログレベルについては” com.nec.webotx.logging.OTXJ2SELogLevel.java”クラスで拡張しています。SLOGINFO は、サービスの起動・停止などイベントログ/シスログに表示すべき情報を出力するために、WebOTX が内部で使用している独自のレベルです。そのため、ログレベルの値としては存在しますが、運用管理ツールなどから設定することはできません。

また、ログレベルとイベントログ/シスログとの関係は次のとおりです。

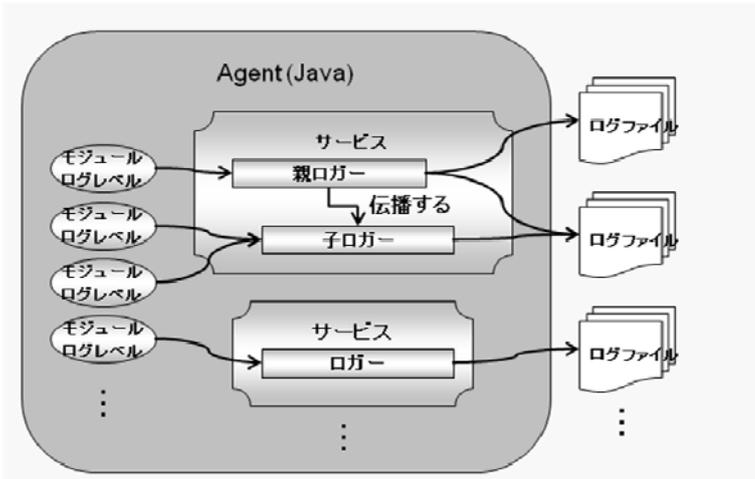
なお、WebOTX では Log4otx のイベントログアペンダ/シスログアペンダの「threshold」を独自に「SLOGINFO」以上に制限しています。そのため、イベントログ、シスログには SLOGINFO 以上の情報のみ出力されます。

log4otx ログレベル (値)	イベントログ との関係	シスログ との関係	説明
FATAL (50000)	エラー	EMERG(0)	システムに重大な影響を与える障害
ERROR (40000)	エラー	ERR(3)	システムを継続運用するのに支障をきたす障害
WARN (30000)	警告	WARNING(4)	システムを継続運用することはできるレベルの障害
SLOGINFO (25000)	情報	INFO(6)	サービスの起動・停止などイベントログとして表示すべき運用状況を示す情報

### 4.1.3. モジュールログレベルの設定

ログのログレベルを実行モジュール単位にまとめた設定値をモジュールログレベルと呼びます。詳細な動作確認や障害調査などの目的で、該当モジュールのみログレベルを変更したい場合には、モジュールログレベルが有効です。

ただし、モジュールログレベルの単位と個々のログは1対1の反映関係にあるは限りません。個々のログはそのパッケージ名(ログドメイン名)の階層構造により親子関係を持つため、1つのモジュールログレベルが複数のログのレベルを変更することがあります。例えば、後述する「root」「server」「core」「tools」といったモジュールログレベルは、操作対象となるログに親子関係があるため、それぞれの子にあたるログにもログレベルの変更が反映されます。これは、個々のモジュール単位にではなく、システム全体で一括してログレベルを変更したい場合などに有効です。ログの親子関係については「7.1 モジュールとログとアベンダの一覧」を参照してください。



Web/Standard-J Edition の場合、モジュールログレベルはエージェントプロセス内で動作するモジュールに対するログの出力レベルを設定します。

Standard/Enterprise Edition の場合、モジュールログレベルの設定によるエージェントプロセス内での動作は Web/Standard-J Edition と同様ですが、Web コンテナや EJB コンテナなど、プロセスグループ上で動作するモジュールに対して、エージェントプロセスでの設定が全てのプロセスグループ上の同じモジュールのログに対しても反映されます。

また、モジュールログレベルの設定値は永続化されますが、ログに設定されているログレベルは永続化されません。そのため、ドメインを再起動すると、「4.1.9 ログレベルの反映順序」での「ドメイン起動時」で解説している順序によって、初期化されたログにログレベルが反映されます。

モジュールログレベルは、ドメイン稼働中に運用管理ツールから設定変更が可能です。設定方法については「4.1.4 モジュールログレベルの設定方法」を参照してください。

モジュールログレベルでは、以下のモジュール毎のログレベルが設定可能です。

モジュール名 (統合運用管理ツールの表示名)	説明	ログレベルの 既定値
root (Root)	システム全体のモジュールに対するログレベルを指定します。	CONFIG
server (Server)	アプリケーションサーバ実行基盤のモジュールに対するログレベルを指定します。	CONFIG
webserver (Web Server)	WebOTX Web Server のモジュールに対するログレベルを指定します。	CONFIG

web-container (Web Container)	Web コンテナのモジュールに対するログレベルを指定します。	CONFIG
jaxrpc (JAXRPC)	JAXRPC のモジュールに対するログレベルを指定します。	CONFIG
jaxws (JAXWS)	JAXWS のモジュールに対するログレベルを指定します。	CONFIG
saaj (SAAJ)	SAAJ のモジュールに対するログレベルを指定します。	CONFIG
jaxr (JAXR)	JAXR のモジュールに対するログレベルを指定します。	CONFIG
ejb-container (EJB Container)	EJB コンテナのモジュールに対するログレベルを指定します。	CONFIG
appclient-container (Application Client Container)	アプリケーションクライアントコンテナのモジュールに対するログレベルを指定します。	CONFIG
jdo (JDO)	JDO モジュールのログレベルを指定します。	CONFIG
classloader (Classloader)	クラスローダのモジュールに対するログレベルを指定します。	CONFIG
core (Server Core)	アプリケーションサーバのコア部分のモジュールに対するログレベルを指定します。	CONFIG
jndi (JNDI)	JNDI のモジュールに対するログレベルを指定します。	CONFIG
jndi-service (JNDI Service)	JNDI サービスのモジュールに対するログレベルを指定します。	CONFIG
jms (JMS Resource)	JMS リソースのモジュールに対するログレベルを指定します。	CONFIG
jms-client (JMS Client)	JMS クライアントのモジュールに対するログレベルを指定します。	CONFIG
jta (JTA)	JTA のモジュールに対するログレベルを指定します。	CONFIG
transaction-service (Transaction サービス)	Transaction サービスのモジュールに対するログレベルを指定します。	CONFIG
jdbc-datasource (JDBC Data Source)	JDBC データソースのモジュールに対するログレベルを指定します。	CONFIG
activation (JavaBeans Activation Framework)	JavaBeans Activation Framework のモジュールに対するログレベルを指定します。	CONFIG

javamail (JavaMail)	JavaMail のモジュールに対するログレベルを指定します。	CONFIG
cobra (CORBA)	CORBA のモジュールに対するログレベルを指定します。	CONFIG
oadj (OAJ)	Object Broker サービス oadj のモジュールに対するログレベルを指定します。	CONFIG
objectbrokerjava-lib (Object Broker Java Library)	Object Broker Java Library のモジュールに対するログレベルを指定します。	CONFIG
objectbrokerjava-msg (Object Broker Java Message)	Object Broker Java Message のモジュールに対するログレベルを指定します。	CONFIG
objectbroker (Object Broker)	Object Broker のモジュールに対するログレベルを指定します。	SEVERE
deployment (Deployment)	配備のためのモジュールに対するログレベルを指定します。	CONFIG
tools (Tools)	実行ツール向けのモジュールに対するログレベルを指定します。	CONFIG
configuration (Configuration)	コンフィグレーション管理のモジュールに対するログレベルを指定します。	CONFIG
admin (Admin)	運用管理全般のモジュールに対するログレベルを指定します。	CONFIG
jmx-remote-access (JMX Remote Access)	JMX リモートアクセスのモジュールに対するログレベルを指定します。	CONFIG
tpmonitormanager (TP Monitor Manager)	TP モニタマネージャのモジュールに対するログレベルを指定します。	CONFIG
nrb-tpmonitormanager (TP Monitor Manager NRB)	TP モニタマネージャ NRB のモジュールに対するログレベルを指定します。	CONFIG
admlistener (Adm Listener)	Adm リスナのモジュールに対するログレベルを指定します。	CONFIG
admin-assistant (Admin assistant)	運用アシスタントのモジュールに対するログレベルを指定します。	CONFIG
security (Security)	セキュリティ管理のモジュールに対するログレベルを指定します。	CONFIG
util (Util)	ユーティリティのモジュールに対するログレベルを指定します。	CONFIG
watchservice (Watch Service)	Watch Service のモジュールに対するログレベルを指定します。	CONFIG
process-conductor	Process Conductor のモジュールに対する	CONFIG

(Process Conductor)	ログレベルを指定します。	
---------------------	--------------	--

#### 4.1.4. モジュールログレベルの設定方法

以下に、統合運用管理ツールと運用管理コマンドを例に、モジュールログレベルの変更方法を示します。

##### ■統合運用管理ツールからログレベルを変更する場合

運用管理ツールから以下の手順で変更を行います。

- ・運用管理ツールを起動しドメインと接続します。
- ・管理対象ツリーより、〈ドメイン名〉→アプリケーションサーバ→ログサービス→モジュールログレベル を選択します。
- ・各コンポーネントのログレベルを変更し、「更新」ボタンを押して設定を反映させます。

##### ■運用管理コマンドからログレベルを変更する場合

運用管理コマンドを起動し set コマンドを使ってログレベルを変更します。

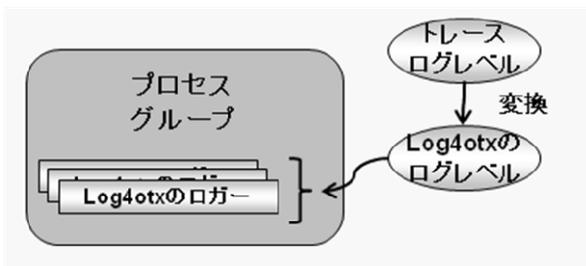
```
otxadmin> login --user admin --password **** --port <管理ポート番号>
otxadmin> get server.log-service.module-log-levels.*
(現在のログレベル設定内容が表示されます。)
otxadmin> set server.log-service.module-log-levels.<モジュール名>=<出力レベル>
(変更後のログレベル設定内容が表示されます。)
otxadmin> exit
```

※〈モジュール名〉には、前節で示した表中の「モジュール名」を指定します。

※〈出力レベル〉には、後述するモジュールログレベルを指定します。

モジュールログレベル値は「4.1.2 各種ログレベルの対応関係について」を参照してください。は「4.1.2 各種ログレベルの対応関係について」を参照してください。

#### 4.1.5. トレースログレベルの設定



トレースログレベルは、プロセスグループ毎にログレベルを変更するための設定です。プロセスグループ内のログ出力はエージェントプロセスと同様、Log4otx を利用して出力されます。そのため、トレースログレベルは内部で Log4otx のログレベル値に変換された後、ロガーに設定されます。

例えば、トレースログレベルを「6.情報メッセージ」に設定した場合、「4.1.2 各種ログレベルの対応関係について」での表に従い、ロガーには Log4otx のログレベルに直した値 (CONFIG) が設定されることになります。

また、トレースログレベルの値は永続化されますが、ロガーに設定されているログレベルは永続化されません。そのため、プロセスグループを再起動すると、「4.1.9 ログレベルの適用順序」の「プロセスグループ起動時」で説明する順序でログレベルが反映されます。個々のロガーにログレベルを設定する場合は、「4.1.7 Log4otx ログレベル」を参照してください。

なお、トレースログレベルの設定は、変更時のプロセスグループの起動状態や、ロガーに現在設定されている Log4otx ログレベル、およびトレースログレベルの変更前と変更後の値を基に、その変更の適用可否が決定されます。具体的には、現在の Log4otx ログレベル、設定の変更前および変更後のトレースログレベルをそれぞれ「4.1.2 各種ログレベルの対応関係について」での表中に対応する Log4otx ログレベルに置き換えた上で、次の条件を満たす場合に設定の変更がロガーに適用されます。

#### 【トレースログレベルの設定適用条件】

※ ロガーに現在設定されている Log4otx ログレベル :  $L$

※ トレースログレベルの変更前の値に対応する Log4otx ログレベル :  $Lt$

※ トレースログレベルの変更後の値に対応する Log4otx ログレベル :  $Lt'$

- プロセスグループ起動処理中(永続化された設定のプロセス再起動に伴う適用条件)

設定変更後のトレースログレベルが既定値の「5.通知メッセージ」以外  
かつ

$L$  の値 が Log4otx ログレベルの SLOGINFO の値 と  $Lt'$  の値 との間に位置する

- プロセスグループ起動完了後(運用中の設定変更に伴うその適用条件)

$L$  の値 が  $Lt$  の値 と  $Lt'$  の値 との間に位置する

それぞれのケースにおいて条件を満たさない場合、変更したトレースログレベルの設定は個々のロガーに対して反映されません。

以下に具体例を記載します。ログレベルの対応関係については、「4.1.2 各種ログレベルの対応関係について」を合わせて参照してください。

#### 例 1) プロセスグループ起動完了後のトレースログレベル反映の例 1

トレースログレベルを「6.情報メッセージ」から「4.警告メッセージ」に変更する場合を例に挙げます。

ロガーに設定されているログレベル値: INFO

トレースログレベルの変更後の値 : 6.情報メッセージ (Log4otx ログレベルの CONFIG 相当)

トレースログレベルの変更前の値 : 4.警告メッセージ (Log4otx ログレベルの WARN 相当)

この場合、以下に示すように、ロガーの現在の値である INFO は変更後の値 CONFIG と変更前の値 WARN の間にあるため、ログレベルは CONFIG に変わります。



### 例 2) プロセスグループ起動完了後のトレースログレベル反映の例 2

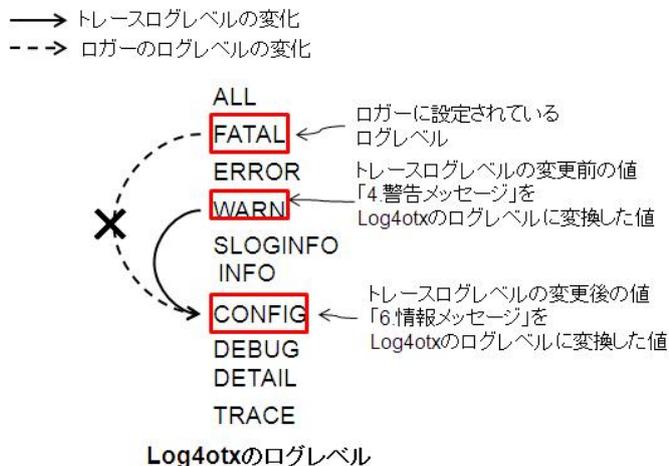
トレースログレベルを「6.情報メッセージ」から「4.警告メッセージ」に変更する場合を例に挙げます。

ログラーに設定されているログレベル値: FATAL

トレースログレベルの変更後の値: 6.情報メッセージ (Log4otx ログレベルの CONFIG 相当)

トレースログレベルの変更前の値: 4.警告メッセージ (Log4otx ログレベルの WARN 相当)

この場合、以下に示すように、ログラーの現在の値である FATAL は変更後の値 CONFIG と変更前の値 WARN の間にないので、ログレベルは FATAL のままになります。



### 例 3) ドメイン起動時およびプロセスグループ起動時のトレースログレベル反映の例 1

トレースログレベルが「7.デバッグ時のメッセージ」で永続化されている場合を例に挙げます。

ログラーに設定されているログレベル値: INFO

トレースログレベルの変更後の値: 7.デバッグ時のメッセージ (Log4otx ログレベルの TRACE 相当)

トレースログレベルの変更前の値: -- (既定値である Log4otx ログレベルの SLOGINFO 相当)

プロセスグループ起動時は、トレースログレベルは一旦既定値である Log4otx ログレベルの SLOGINFO に設定され、その後に永続化された値が適用されます。そのため、「トレースログレベルの変更前の値」は既定値 SLOGINFO になります。また、この例の場合、「7.デバッグ時のメッセージ」で永続化されているため、これが「トレースログレベルの変更後の値」になります。

この場合、以下に示すように、現在の値である INFO は変更後の値 TRACE と変更前の値 SLOGINFO の間にあるので、ログレベルは TRACE になります。



#### 例 4) ドメイン起動時およびプロセスグループ起動時のトレースログレベル反映の例 2

トレースログレベルが「5.通知メッセージ」で永続化されている場合を例に挙げます。

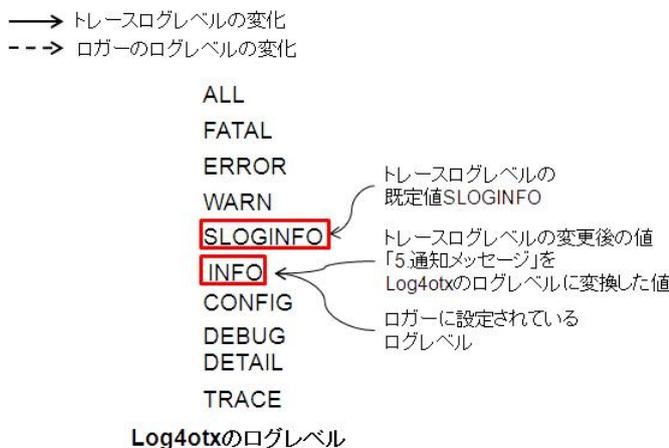
ロガーに設定されているログレベル値: INFO

トレースログレベルの変更後の値: 5.通知メッセージ (Log4otx ログレベルの INFO 相当)

トレースログレベルの変更前の値: -- (既定値である Log4otx ログレベルの SLOGINFO 相当)

例 3 と同様、「トレースログレベルの変更前の値」は既定値 SLOGINFO になります。また、この例の場合、「5.通知メッセージ」で永続化されているため、これが「トレースログレベルの変更後の値」になります。

ここで、「5.通知メッセージ」はトレースログレベルの既定値です。そのため、先の適用条件を満たせず、トレースログレベルの値は反映されません。従って、ログレベルは INFO のままになります。



## 4.1.6. トレースログレベルの設定方法

以下に、統合運用管理ツールと運用管理コマンドを例に、トレースログレベルの変更方法を記載します。

### ■統合運用管理ツールからログレベルを変更する場合

運用管理ツールから以下の手順で変更を行います。

- ・運用管理ツールを起動しドメインと接続します。
- ・管理対象ツリーより、〈ドメイン名〉→TP システム→アプリケーショングループ→〈アプリケーショングループ名〉→プロセスグループ→〈プロセスグループ名〉を選択します。
- ・[トレース設定] タブにある、トレースレベルを変更し、「更新」ボタンを押して設定を反映させます。

### ■運用管理コマンドからログレベルを変更する場合

運用管理コマンドを起動し set コマンドを使ってログレベルを変更します。

```
otxadmin> login --user admin --password ***** --port 6212
otxadmin> get tpsystem.applicationGroups.<アプリケーショングループ名>.processGroups.<プロセスグループ名>.traceLevel
(現在のログレベル設定値が表示されます。)
otxadmin> set tpsystem.applicationGroups.<アプリケーショングループ名>.processGroups.<プロセスグループ名>.traceLevel=<出力レベル>
(変更後のログレベル設定値が表示されます。)
otxadmin> exit
```

※<出力レベル>には、後述するトレースログレベルを指定します。

トレースログレベルのログレベル値は「4.1.2 各種ログレベルの対応関係について」を参照してください。

## 4.1.7. Log4otxログレベル

Log4otx ログレベルは、Log4j を基に、WebOTX が独自の拡張を施したログレベルです。各ロガーに対する初期値を設定するために使用します。Log4j の仕様と同様に、設定ファイルを直接編集することで変更可能です。

## 4.1.8. Log4otxログレベルの設定方法

WebOTX のログ設定はドメイン単位に Log4otx の定義ファイル(log4otx.xml)に定義されています。この XML ファイルを編集することでログの設定を変更することが可能です(通常の log4j と全く同様の手順で行えます)。ただし、ログレベルの変更を反映させるためには、ドメインの再起動が必要です。

- エージェントプロセスで使用する設定ファイル
  - `${INSTANCE_ROOT}/config/log4otx.xml`

なお、Standard/Enterprise Edition では、各プロセスグループ上のロガーはエージェントプロセスの定義フ

ファイルとは別に用意されています。各プロセスグループに対してもエージェントプロセスの場合と同様に設定を変更することが可能です。ただし、ログレベルの変更を反映させるためには、プロセスグループの再起動が必要です。

- プロセスグループで使用する設定ファイル

➤ `${INSTANCE_ROOT}/config/tpssystem/logconf/${APGNAME}-${PGNAME}/log4otx.xml`

Log4otx のログレベル設定値は「4.1.2 各種ログレベルの対応関係について」を参照してください。

#### 4.1.9. ログレベルの適用順序

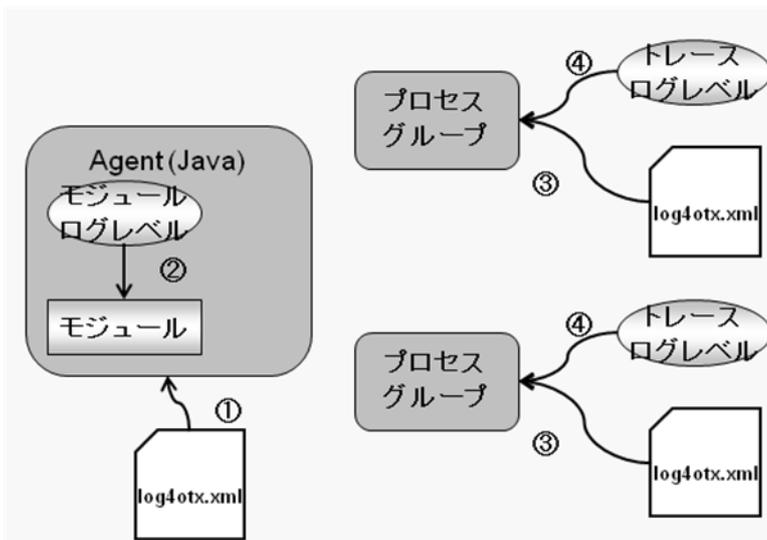
「モジュールログレベル」、「トレースログレベル」、「Log4otx ログレベル」の3つのログレベルは、それぞれ、ログレベルとしてロガーに反映されるタイミングが異なります。また、各々のログレベルは既に設定されている値を上書きすることで反映されるため、最後に設定されたログレベルが最も優先されます。本節では、その適用順序について整理します。適用順序を把握した上で、適切に設定を行ってください。

- ドメイン起動処理時(永続化されたログレベル設定の適用順序)

エージェントプロセスでは、その起動処理中は「Log4otx ログレベル→モジュールログレベル」の順でログレベルの設定が適用されます。

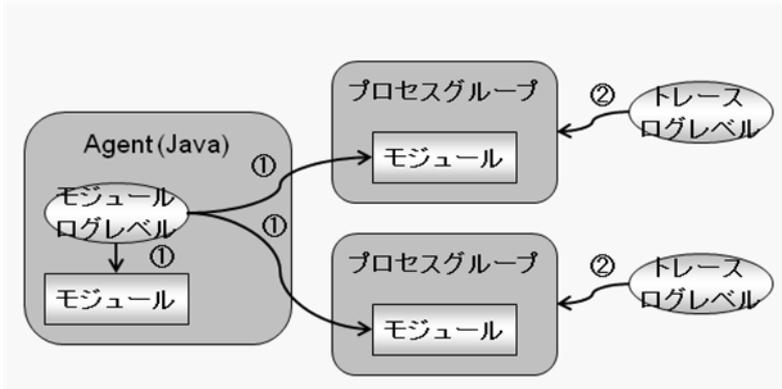
また、Standard/Enterprise Edition の場合、プロセスグループでは「Log4otx ログレベル→トレースログレベル」の順でログレベルの設定が適用されます。

ただし、いずれの場合も例外として「web-container(Web Container)」モジュールに属するロガーについては「Log4otx ログレベル」の設定は適用されません。



- ① エージェントプロセスの各ロガーに Log4otx ログレベル(log4otx.xml での設定)が適用される。
- ② 永続化されたモジュールログレベルの設定値がエージェントプロセスの各ロガーに適用される。  
＜以降は Standard/Enterprise Edition 利用時の流れ＞
- ③ プロセスグループの各ロガーに Log4otx ログレベル(log4otx.xml での設定)が適用される。  
※web-container(Web Container) は Log4otx ログレベルが適用されません。
- ④ 永続化されたトレースログレベルの設定がプロセスグループの各ロガーに反映される。

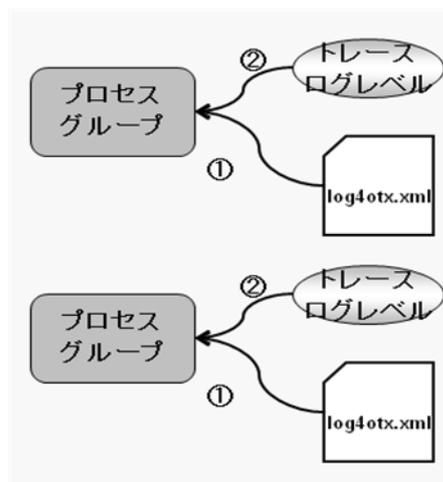
- ドメイン稼働中(運用中の設定変更に伴う反映順序)



- ① モジュールログレベルを変更すると、その設定がエージェントおよび全てのプロセスグループのログに反映される。
- ② Standard/Enterprise Edition の場合、トレースログレベルを変更すると、その設定が対象となるプロセスグループのログに反映される。

この時、①→②の順で設定変更を行った場合には、プロセスグループでは①の変更が②の変更で上書きされるため、②の設定が優先されます。

- プロセスグループ起動時(永続化された設定の適用順序)



- ① プロセスグループの各ログに Log4otx ログレベル(log4otx.xml での設定)が適用される。  
※web-container(Web Container) は Log4otx ログレベルが適用されません。
- ② 永続化されたトレースログレベルの設定がプロセスグループの各ログに反映される。

## 4.2. Log4otxのローテーションの設定方法

### 4.2.1. ログファイルの最大サイズの変更

ログレベルの場合と同様に、Log4otx の定義を変更することで設定を行います。

- エージェントプロセスで出力されるログの場合
  - `${INSTANCE_ROOT}/config/log4otx.xml`
- プロセスグループで出力されるログの場合

➤ `${INSTANCE_ROOT}/config/tpsystem/logconf/${APGNAME}-${PGNAME}/log4otx.xml`

log4otx.xml 内の appender 要素の MaxFileSize プロパティの値を変更します。

既定では以下のように設定(1MB)されています。

```
<param name="MaxFileSize" value="1MB"/>
```

例: ログファイルの最大サイズを 2MB に変更

```
<param name="MaxFileSize" value="2MB"/>
```

## 4.2.2. ログの世代数の変更

ログレベルの場合と同様に、以下の Log4otx の定義を変更することで設定を行います。

- エージェントプロセスで出力されるログの場合

➤ `${INSTANCE_ROOT}/config/log4otx.xml`

- プロセスグループで出力されるログの場合

➤ `${INSTANCE_ROOT}/config/tpsystem/logconf/${APGNAME}-${PGNAME}/log4otx.xml`

log4otx.xml の appender 要素の MaxBackupIndex の値を変更します。

既定では以下のような設定(ログを 2 世代残す)になっています。

```
<param name="MaxBackupIndex" value="2" />
```

例: ログを3世代残す場合

```
<param name="MaxBackupIndex" value="3" />
```

## 4.3. rotateloggsを利用したログローテーションの設定方法

### 4.3.1. server.logのローテーション

server.log などではログのローテーション機能がありませんが、WebOTX Web Server (2.0)に付属している rotateloggs (Apache に付属するツール)を利用することで、指定期間やファイルサイズによる切り替えを行うことが可能です。

通常は WebOTX を停止したタイミングでログを退避(削除)する必要がありますが、連続運用など WebOTX を停止せずに server.log の切り替えを行う必要がある場合に有効です。ただし、rotateloggs では、切り替後のファイルを自動で削除する機能はないため、定期的に古いファイルを削除する必要があります。

rotateloggs のコマンド形式

```
${INSTALL_ROOT}/WebServer2/bin/rotateloggs [-l] logfile [ rotationtime [ offset ] ] [ filesizeM ]
```

\* サイズを指定する場合は、数字の後に「M」の文字を指定します。

注意事項: rotateloggs を利用して server.log をローテーションさせる方法は UNIX 環境でのみ有効です。

1. `${INSTALL_ROOT}/lib/launcher.xml` の編集

launcher.xml の以下の箇所の redirectoutput の値を false に変更します。

・変更前

```
<launch classname=" com.nec.webotx.enterprise.server.WebOTXMain"
  requiretools="true"
  print="{print}"
  debug="{jdb}"
  usesystemin="false"
  waitforchild="{wait}"
  redirectoutput="true"
```

```
output="{com.nec.webotx.instanceRoot}{file.separator}logs{file.separator}server.log"
appendoutput="true"
```

・変更後

```
<launch classname=" com.nec.webotx.enterprise.server.WebOTXMain"
  requiretools="true"
  print="{print}"
  debug="{jdb}"
  usesystemin="false"
  waitforchild="{wait}"
  redirectoutput="false"
```

```
output="{com.nec.webotx.instanceRoot}{file.separator}logs{file.separator}server.log"
appendoutput="true"
```

## 2. \${INSTANCE\_ROOT}/bin/startserv の編集

startserv の最終行を編集します。

・変更前

```
~--DCATALINA_OPTS="{CATALINA_OPTS}" -Dwait="true" start "{@}" >>
"${INSTANCE_ROOT}/logs/server.log" 2>&1 &
```

・変更後

```
~--DCATALINA_OPTS="{CATALINA_OPTS}" -Dwait="true" start "{@}" 2>&1
|/opt/WebOTX/WebServer2/bin/rotatelogs "${INSTANCE_ROOT}/logs/server.log" 86400 &
```

## 4.3.2. Webサーバのログのログローテーション

Web サーバが出力するログのローテーション方法については、マニュアルの以下の章を参照してください。

[運用編]-[WebOTX Web サーバ運用ガイド]-[4.6 ログファイルのローテーション]

## 4.3.3. mod\_jkのログのローテーション

Web サーバと Web コンテナとの連携モジュールである mod\_jk が出力するログファイルのローテーション方法について説明します。

はじめに、mod\_jk の設定ファイル名について説明します。mod\_jk の設定ファイルは使用する Web サーバと WebOTX の動作モードによって以下のように変わります。設定ファイルは \${INSTANCE\_ROOT}/config/WebCont 以下にあります。

使用 Web サーバ	WebOTX 動作モード	mod_jk 設定ファイル名
------------	--------------	----------------

WebOTX Web サーバ 1.3	シングルプロセスモード	mod_jk.conf-auto
WebOTX Web サーバ 1.3	マルチプロセスモード	mod_jk_om.conf-auto
WebOTX Web サーバ 2.0	シングルプロセスモード	mod_jk-20.conf-auto
WebOTX Web サーバ 2.0	マルチプロセスモード	mod_jk_om-20.conf-auto

本章では WebOTX インストールするフォルダは既定値(C:/WebOTX)、ドメイン名は domain1、WebOTX Web サーバ 2.0 とマルチプロセスモードの組み合わせを例に手順を説明いたします。組み合わせがことなる場合も手順は同じですので、ファイル名、パス名を適宜読み替えてください。

以下の手順を、ドメインが停止した状態で行ってください。

1. mod\_jk 設定ファイルのリネーム  
mod\_jk\_om-20.conf-auto をコピーし、mod\_jk\_om-20.conf-load を作成します。
2. mod\_jk\_om-20.conf-load の編集  
以下の例では、一日(86400 秒)ごとにローテーションさせます。適宜数値は変更してください。また、ファイルサイズでローテーションさせる場合は、数値の後に M をつけることで実現することができます。例えば、10MB 単位にローテーションさせる場合は 10M と記述します。

Windows の場合

・変更前

JkLogFile "C:/WebOTX/domains/domain1/logs/webcontainer/mod\_jk\_om-20.log"

・変更後

JkLogFile "|C:/WebOTX/WebServer2/bin/rotatelogs.exe  
C:/WebOTX/domains/domain1/logs/webcontainer/mod\_jk\_om-20.log 86400"

UNIX の場合

・変更前

JkLogFile "/opt/WebOTX/domains/domain1/logs/webcontainer/mod\_jk\_om-20.log"

・変更後

JkLogFile "|/opt/WebOTX/WebServer2/bin/rotatelogs  
/opt/WebOTX/domains/domain1/logs/webcontainer/mod\_jk\_om-20.log 86400"

3. httpd.conf の編集  
\${INSTANCE\_ROOT}/config/WebServer/httpd.conf を編集し、2 で作成した mod\_jk\_om-20.conf-load を読み込むように編集します。

Windows の場合

・変更前

include "C:/WebOTX/domains/domain1/config/WebCont/mod\_jk\_om-20.conf-auto"

・変更後

include "C:/WebOTX/domains/domain1/config/WebCont/mod\_jk\_om-20.conf-load"

UNIX の場合

・変更前

include "/opt/WebOTX/domains/domain1/config/WebCont/mod\_jk\_om-20.conf-auto"

・変更後

include "/opt/WebOTX/domains/domain1/config/WebCont/mod\_jk\_om-20.conf-load"

本手順を行うと、mod\_jk\_om-20.log.<数字>というファイル名でログが出力されます。数字が小さいものは過

去ログとなりますので、削除することができます。

## 4.4. Log4Jで利用できるJavaシステムプロパティ

### 4.4.1. Log4Jで利用できるJavaシステムプロパティ

Standard、Enterprise Edition を利用している場合は、次の Java システムプロパティが利用できます。

これらの Java システムプロパティを log4j のプロパティに指定することにより、プロセス ID の取得等が可能になります。設定を反映させるために、エージェントプロセスの場合はドメインの再起動、プロセスグループの場合はプロセスグループの再起動が必要です。

出力情報	Java システムプロパティ名
ドメイン名	domain.name
System 名	jp.co.nec.WebOTX.SystemName
アプリケーショングループ名	jp.co.nec.WebOTX.APGroupName
プロセスグループ名	jp.co.nec.WebOTX.ProcessGroupName
プロセス ID	jp.co.nec.WebOTX.ProcessID

(注意) 上記の Java システムプロパティに指定する値は変更できません。

#### 設定例

プロセスグループから出力するログファイルにプロセス ID を付加する例を示します。

`${INSTANCE_ROOT}/config/tpsystem/logconf/<アプリケーショングループ名>-<プロセスグループ名>/log4otx.xml` に以下のような定義を追記してください。

```
<appender name="SAMPLE" class="com.nec.webotx.org.apache.log4j.RollingFileAppender">
  <param name="File"
    value="${com.nec.webotx.instanceRoot}/logs/tpsystem/apg/pg/sample_${jp.co.nec.WebOTX.ProcessID}.log" />
  <param name="MaxFileSize" value="1MB" />
  <param name="MaxBackupIndex" value="2" />
  <layout class="com.nec.webotx.org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%d %-5p %c - %m [%t]%n" />
  </layout>
</appender>
```

次に、対象とするロガーの出力先を「WOTRACE」から「SAMPLE」に変更してください。

```
<logger name="com.nec.webotx">
  <level class="com.nec.webotx.logging.OTXLogLevel" value="INFO" />
  <appender-ref ref="SAMPLE" />
</logger>
```

最後に設定を反映させるために、プロセスグループを再起動してください。

# 5. サーバAPのトレースログについて

## 5.1. Standard/Enterprise EditionにおけるサーバAPのログ

Standard/Enterprise Edition におけるサーバアプリケーションに関するログについて説明します。サーバアプリケーションの動作に問題がある場合に参照すべきログについて説明します。

WebOTX にはシステムやプロセスの稼動状況を示すために様々なトレースファイルが存在します。何か異常が起こったときは以下に説明するファイルを参照してください。

この表の上位 4 つのファイルで、ほとんどの場合に対処可能です。トレースレベルを最大にすると通常運用に支障が出る場合もあります。最大よりは1つないし2つほど下げて運用してください。デバッグ時は常に最大にしてください。これらのファイルについてはファイルの読み方とエラー番号一覧(別のセクションにあるものもあります)を示します。

また、下位 2 つは上位 4 つでも判らない時に採取するログです。通常運用には耐えられないほどのログが出ますので、デバッグ時や開発元からの依頼があった時のみ採取願います。下位 2 つのファイルに関しては採取方法のみ示します。

トレース種別	トレース内容
サーバアプリケーショントレース(Java)	WebOTX のトレースとユーザトレース(TPSUserTrace())をマージして出力します。
サーバアプリケーショントレース(C++)	WebOTX のトレースとユーザトレース(TPSUserTrace())をマージして出力します。
システムトレース	プロセス起動時の情報や CORBA アプリケーションで検出したエラーの情報が出力します。
イベントログ(Windows)、システムログ(UNIX)	WebOTX 内部で異常が発生した時に異常の内容を出力します。
サーバアプリケーション内部トレース	サーバアプリケーションの下位層(スレッド管理やメッセージ管理など)の状況出力します。
IIOP リスナ内部トレース	クライアントからの要求を管理する部分と下位層(ソケット管理やメッセージ管理など)の状況出力します。

### 5.1.1. サーバアプリケーショントレース採取

サーバアプリケーションのログについて説明します。

ファイル名	ディレクトリ	サイズ	説明
-------	--------	-----	----

<p>&lt;pgname&gt;.&lt;pid&gt;.log</p>	<pre> \${ INSTANCE_ROOT }  /logs/tpsystem/ &lt;apgname&gt;/&lt;pgname&gt; </pre>	<p>ユーザ 指定値</p>	<p>サーバアプリケーショントレースを採取するには統合運用管理ツールで設定する必要があります。</p> <p>統合運用管理ツールにてプロセスグループのアイコンを選択し、[トレース設定]タブをクリックして[トレース設定]ページを表示してください。このページにある[トレースレベル]が、[出力しない]になっていないことを確認してください。次に、トレースファイルの最大サイズを確認してください(既定値 32 KBytes)。トレースファイル名を確認してください。最後にトレースレベルを確認してください(既定値 5)。</p> <p>トレースファイルはトレースファイルの最大サイズに達すると、ファイル名に “.bak” を付加してバックアップファイルを作成します。そして、新規にトレースファイルを作成します。バックアップは一回分しか残しません。</p> <p>WebOTX 以外のアプリケーションからサーバアプリケーショントレースにログを出力しても、サイズ算定に含まれません。最大サイズのチェックは出力毎であるため、指定された最大サイズとバックアップされたファイルのサイズは、必ず同じになるわけではありません。</p> <p>サーバアプリケーション中からトレースファイルにトレースを出力するには TPSUserTrace() を使用してください。Java の場合は <code>java.lang.System.out.println()</code> を用いても出力できます。C++ の場合は <code>printf()</code> を用いても出力できますが、トレースファイルのサイズの最大値以上に出力されることがあります。</p> <p>トレースレベルが 5(LOG_NOTICE) の場合、WebOTX としてエラー以外は何も出力しません。</p>
<p>&lt;pgname&gt;_sys.&lt;pid&gt;.log</p>	<pre> \${ INSTANCE_ROOT }  /logs/tpsystem/ &lt;apgname&gt;/&lt;pgname&gt; </pre>	<p>1MB</p>	<p>システムトレースファイルです。</p> <p>プロセス起動時の情報や CORBA アプリケーションで検出したエラーの情報が出力されます。Java アプリケーションでスタック情報採取を行ったときにも出力されます。上記以外の機会に出力されることはありません。</p>
<p>&lt;apgname&gt;_EXIT_sys.&lt;pid&gt;.log</p>	<pre> \${ INSTANCE_ROOT }  /logs/tpsystem/ &lt;apgname&gt;/&lt;apgname&gt;_EXIT </pre>	<p>1MB</p>	<p>アプリケーション初期プロセスのシステムトレースファイルです。アプリケーション初期プロセスを使用する場合のみ出力されます。プロセス起動時の情報やエラーの情報が出力されます。</p>
<p>&lt;pgname&gt;_recovery.&lt;pid&gt;.log</p>	<pre> \${ INSTANCE_ROOT }  /logs/tpsystem/ </pre>	<p>1MB</p>	<p>プロセス異常終了によりプロセスグループが停止した場合に出力されます。</p>

	<apgname>/<pgname>		
<pgname>.<time>.<pid>.log <pgname>_sys.<time>.<pid>.<pgname>_recovery.<time>.<pid>.log	<pre> \${INSTANCE_ROOT} } /logs/tpsystem/ &lt;apgname&gt;/&lt;pgname&gt; /save </pre>	ユーザー指定値	<p>プロセス終了時及びプロセスグループ起動時に、使用していないサーバアプリケーショントレースファイルを save ディレクトリに退避します。save ディレクトリに退避されたログはプロセス異常終了時の障害解析等で必要になる場合がありますが、任意のタイミングで save ディレクトリを退避あるいは削除することができます。ディスクサイズを節約したい場合は定期的に save ディレクトリを削除してください。なお、save ディレクトリに退避されていないログは運用中のプロセスのログなので削除しないでください。</p>
<apgname>_EXIT_sys.<time>.<pid>.log	<pre> \${INSTANCE_ROOT} } /logs/tpsystem/ &lt;apgname&gt;_EXIT/save </pre>	1MB	<p>アプリケーション初期プロセスのプロセス終了時及びアプリケーショングループ起動時に、使用していないシステムトレースファイルを save ディレクトリに退避します。save ディレクトリに退避されたログはプロセス異常終了時の障害解析等で必要になる場合がありますが、任意のタイミングで save ディレクトリを退避あるいは削除することができます。ディスクサイズを節約したい場合は定期的に save ディレクトリを削除してください。なお、save ディレクトリに退避されていないログは運用中のプロセスのログなので削除しないでください。</p>

### 5.1.2. トレースの読み方(Java)

サーバアプリケーションログは以下のような形式になっています。

ログレベル 1~3	yyyy/mm/dd HH:MM:SS.sss TTT: Error: メッセージ
ログレベル 4~7	yyyy/mm/dd HH:MM:SS.sss TTT: メッセージ

yyyy 西暦    HH 時(00~23)    sss    ミリ秒    xxxxxxxx エラーID(無い場合もあります)  
mm 月    MM 分(00~59)    TTT    論理スレッド番号  
dd 日    SS 秒(00~59)    メッセージ    メッセージ本文

WebOTX のサーバアプリケーションが出力するログの中に Error が出力される場合は何か異常が起きています。主に初期化時にエラーが出ることが多いのでここでは初期化時に発生するエラー一覧を載せます。これ以外のエラーが出ている場合は開発元に連絡してください。

エラーメッセージ	原因	対処
Can't get catalog directory.	環境変数「TPM_CAT_DIR」の取得に失敗しました。	メモリが不足している可能性があります。OS 再起動してください。それでも解決しない場合は、可能なら WebOTX を再インストールしてください。
Can't get application server name.	環境変数「TPM_AP_CLASS」の取得に失敗しました。	メモリ不足の可能性があります。マシンのメモリ使用量を確認してください。
Can't get TP_HOME.	環境変数「TP_HOME」の取得に失敗しました。	WebOTX のインストール情報が壊れている可能性があります。可能なら再インストールしてください。
WG2 file is invalid.	コンポーネント定義ファイル(WG2 ファイル)が不正です。	コンポーネント定義ファイル(WG2 ファイル)が壊れている可能性があります。可能なら再インストールしてください。
Can't read wc2 file.	コンポーネント定義ファイル(WC2)ファイルの読み込みに失敗しました。	コンポーネント定義ファイル(Wc2 ファイル)が壊れている可能性があります。可能なら WebOTX を再インストールしてください。
Cannot initialize Winsock.	WinSock の初期化に失敗しました。	WSock32.dll のバージョン(1.1 以上が利用可能か)を確認してください。
Can't access naming mutex object.	名前サーバアクセス用 Mutex の生成に失敗	メモリ不足の可能性があります。マシンのメモリ使用量を確認してください。
Failed in acquiring the port number of the asynchronous message.	サービスのポートに、IOP リスナ(アライブチェック)用ポートが設定されていません。	/etc/services に "webotx-mess" という名前のサービスを追加してください。
Cannot load ObjectManagerComponent.	WebOTX のライブラリのロードに失敗しました。	WebOTX のインストール情報が壊れている可能性があります。可能なら再インストールしてください。
Can't get mutex object for VISConnector.	VIS コネクタ制御用の Mutex オブジェクトの生成ができませんでした。	メモリ不足の可能性があります。マシンのメモリ使用量を確認してください。UNIX の場合、共有メモリが破壊された恐れがあります。マシン再起動が必要かもしれません。
Get classpath failed.	wo_jars.cnf から CLASSPATH の生成に失敗しました。	WebOTX あるいは Object Broker その他のコンポーネントのインストール情報が壊れている可能性があります。可能なら再インストールしてください。
set PATH failed.	PATH 環境変数の設定に失敗しました。	メモリ不足の可能性があります。マシンのメモリ使用量を確認してください。UNIX の場合、共有メモリが破壊された恐れがあります。マシン再起動が必要かもしれません。
load library error. (jvm.dll) : errno=XX, msg=XX	Java VM のロードに失敗しました。	JDK または JRE のインストール情報が壊れている可能性があります。JDK または JRE を再インストールしてください。
find function error. (JNI_CreateJavaVM) : errno=XX, msg=XX	Java VM の初期化関数を見つけられませんでした。	Java VM の DLL が壊れています。
Can't alloc main object.	Java オブジェクトの生成に失敗しました。	WOTXMAIN.jar が壊れているか改ざんされている恐れがあります。またはメモリが不足している可能性があります。OS 再起動してください。それでも解決しない場合は、可能なら WebOTX を再インストールしてください。
Can't create server object.	サーバオブジェクトの生成ができませんでした。	サーバオブジェクトのコンストラクタ内で例外が発生する要因が無いか調べてください。
CORBANameServer.bind() failed.	名前サーバへのオブジェクトのバインドに	名前サーバが正しく稼働しているか、

	失敗しました。	orbmanag.exe(Object Broker のツール)で確認してください。
Catch OMWatchComException.	オブジェクトのバインド時・アンバインド時に、Watch サーバが停止している可能性があります。	Watch サーバプロセスの存在確認と、Watch サーバのログを確認してください。
Catch OMWatchServerException.	オブジェクトのバインド時・アンバインド時に、Watch サーバ内部で例外しました。	Watch サーバプロセスの存在確認と、Watch サーバのログを確認してください。
OTXSCur.TransactionService.init() is error.	WebOTX Transaction サービスの初期化に失敗しました。	WebOTX Transaction サービスのログを確認して対処してください。
APPProlog abnormally done.	アプリケーションの初期化に失敗しました。	上記エラーが出た場合に出力されません。この前のエラーメッセージを見て対処してください。
Thread does not have initialized(ret=XX).	スレッドの初期化に失敗しました。	上記エラーが出た場合に出力されません。この前のエラーメッセージを見て対処してください。
AbortExit(exitkey=XX, endkey=XX).	例外が発生しました(開始)	例外が発生したので、後処理を開始しました。
AbortExit(exitkey=XX, endkey=XX)(finished).	例外が発生しました(終了)	例外の後処理が完了しました。
Disconnecting terminal(XXXXXX).	クライアントとのセッションが切れました(開始)	CreateServerObject()を呼び出した後、ReleaseServerObject()を呼び出す前にクライアントアプリケーションが終了したと思われる。クライアントアプリケーション側で ReleaseServerObject を呼ばずに終了する箇所があれば、通常のクライアントアプリケーション終了時にも出力される可能性があります。また、このメッセージは対象とするクライアントアプリケーションが使用しているはアプリケーショングループ内の全プロセスに通知されます。
Disconnecting terminal(XXXXXX)(finished).	クライアントとのセッションが切れました(開始)	クライアントとのセッションが切れた処理が完了しました。
The following TX execution is unfinished	このエラーメッセージに記述されたオペレーションの実行途中でプロセスが終了しました。	イベントログ(syslog)を確認し、原因を判別してください。該当オペレーションに問題がないか確認してください。

### 5.1.3. トレースの読み方(C++)

サーバアプリケーションログは以下のような形式になっています。

ログレベル 1~3	yyyy/mm/dd HH:MM:SS.sss TTT: Error: メッセージ
ログレベル 4~7	yyyy/mm/dd HH:MM:SS.sss TTT: メッセージ

yyyy 西暦    HH 時(00~23)    sss    ミリ秒    xxxxxxxx エラーID(無い場合もあります)

mm 月    MM 分(00~59)    TTT    論理スレッド番号

dd 日    SS 秒(00~59)    メッセージ    メッセージ本文

WebOTX のサーバアプリケーションが出力するログの中に Error が出力される場合は何か異常が起きています。主に初期化時にエラーが出ることが多いのでここでは初期化時に発生するエラー一覧を載せません。これ以外のエラーが出ている場合は開発元に連絡してください。

エラーメッセージ	原因	対処
Can't get catalog directory.	環境変数「TPM_CAT_DIR」の取得に失敗しました。	メモリが不足している可能性があります。OS再起動してください。それでも解決しない場合は、可能なら WebOTX を再インストールしてください。
Can't get application server name.	環境変数「TPM_AP_CLASS」の取得に失敗しました。	メモリ不足の可能性があります。マシンのメモリ使用量を確認してください。
Can't get TP_HOME.	環境変数「TP_HOME」の取得に失敗しました。	WebOTX のインストール情報が壊れている可能性があります。可能なら再インストールしてください。
WG2 file is invalid.	コンポーネント定義ファイル(WG2 ファイル)が不正です。	コンポーネント定義ファイル(WG2 ファイル)が壊れている可能性があります。可能なら再インストールしてください。
Can't read wc2 file.	コンポーネント定義ファイル(Wc2)の読み込みに失敗しました。	コンポーネント定義ファイル(Wc2 ファイル)が壊れている可能性があります。可能なら再インストールしてください。
Cannot initialize Winsock.	WinSock の初期化に失敗しました。	WSock32.dll のバージョン(1.1 以上が利用可能か)を確認してください。
Can't access naming mutex object.	名前サーバアクセス用 Mutex の生成に失敗しました。	メモリ不足の可能性があります。マシンのメモリ使用量を確認してください。
Failed in acquiring the port number of the asynchronous message.	サービスのポートに、IIOP リスナ(アライブチェック)用ポートが設定されていません。	/etc/services に"webotx-mess" という名前のサービスを追加してください。
Cannot load ObjectManagerComponent.	WebOTX のライブラリのロードに失敗しました。	WebOTX のインストール情報が壊れている可能性があります。可能なら再インストールしてください。
Cannot load dll = XX.	サーバコンポーネントのロードに失敗しました。	サーバコンポーネントが不正か、コンポーネントが正常に登録できていません。また、サーバコンポーネントと、それがリンクしているライブラリとの間に不整合がある場合も考えられます。サーバコンポーネント、使用している共有コンポーネントを再登録してみてください。それでも解決しない場合は、コンポーネントをリビルドして再登録してください。
exception: CreateRegidentObject()	常駐オブジェクトの生成に失敗しました。	常駐オブジェクトのコンストラクタ内で例外が発生する要因が無いか調べてください。メモリ不足の可能性もあります。
Cannot Find Component Init Function XX.	コンポーネント初期化関数が見つかりません。	設定されているコンポーネント初期化関数の名前を確認してください。
WOServantCallback is not set .	WOServantCallback が set されていません。	コンポーネント初期化関数から、WebOTX::WOComponent::add()を使用してコンポーネント内の全インターフェースの情報を登録するようにコンポーネントを再作成してください。
Can't get function address for create object.	サーバオブジェクト生成関数が見つかりません。	IDL コンパイルからやり直してください。xxx_Entry.C(.cpp)を含めているか確認してください。
Can't create object.	サーバオブジェクトの生成ができませんでした。	サーバオブジェクトのコンストラクタ内で例外が発生する要因が無いか調べてください。メモリ不足の可能性もあります。
Fail binding. error=XX	名前サーバへのオブジェクトのバインドに失敗しました。	名前サーバが正しく稼働しているか、orbmanag.exe(Object Broker のツール)で確認してください。
Fail unbinding. error=XX	名前サーバへのオブジェクトのアンバインドに失敗しました。	名前サーバが正しく稼働しているか、orbmanag.exe(Object Broker のツール)で確認してください。

Catch OMWatchComException.	オブジェクトのバインド時・アンバインド時に、Watch サーバが停止している可能性があります。	Watch サーバプロセスの存在確認と、Watch サーバのログを確認してください。
Catch OMWatchServerException.	オブジェクトのバインド時・アンバインド時に、Watch サーバ内部で例外しました。	Watch サーバプロセスの存在確認と、Watch サーバのログを確認してください。
Can't set property header(process shared memory is exception).	共有プロパティのプロパティアクセスにて共有メモリにアクセスできませんでした。	メモリ不足の可能性があります。マシンのメモリ使用量を確認してください。UNIX の場合、共有メモリが破壊された恐れがあります。マシン再起動が必要かもしれません。
Can't get property header(process shared memory is exception).	共有プロパティのプロパティアクセスにて共有メモリにアクセスできませんでした。	メモリ不足の可能性があります。マシンのメモリ使用量を確認してください。UNIX の場合、共有メモリが破壊された恐れがあります。マシン再起動が必要かもしれません。
WebOTX TransactionService is not installed.	WebOTX Transaction サービスがインストールされてないようです。	PATH 環境変数を確認してください。可能なら WebOTX Transaction サービスを再インストールしてください。
TransactionService::init() is FALSE.	WebOTX Transaction サービスの初期化に失敗しました。	WebOTX Transaction サービスのログを確認して対処してください。
Unable to tx_register().	WebOTX Transaction サービスのデータベース名の設定が失敗しました。	WebOTX Transaction サービスのログを確認して対処してください。
APPProlog abnormally done.	アプリケーションの初期化に失敗しました。	上記エラーが出た場合に出力されます。この前のエラーメッセージを見て対処してください。
Thread does not have initialized(ret=XX).	スレッドの初期化に失敗しました。	上記エラーが出た場合に出力されます。この前のエラーメッセージを見て対処してください。
AbortExit(exitkey=XX, endkey=XX)(finished).	例外が発生しました(終了)	例外の後処理が完了しました。
Disconnecting terminal(XXXXXX).	クライアントアプリケーションとのソケットコネクションが切れました	CreateServerObject()を呼び出した後、ReleaseServerObject()を呼び出す前にクライアントアプリケーションが終了したと思われる。クライアントアプリケーション側でReleaseServerObject を呼ばずに終了する箇所があれば、通常のクライアントアプリケーション終了時にも出力される可能性があります。また、このメッセージは対象とするクライアントアプリケーションが使用しているはアプリケーショングループ内の全プロセスに通知されます。
Disconnecting terminal(XXXXXX)(finished).	クライアントアプリケーションとのソケットコネクションが切れました(終了)	クライアントアプリケーションとのソケットコネクション切断時の後処理が完了しました。
The following TX execution is unfinished.	このエラーメッセージに記述されたオペレーションの実行途中でプロセスが終了しました。	イベントログ(syslog)を確認し、原因を判別してください。該当オペレーションに問題がないか確認してください。

#### 5.1.4. サーバアプリケーション内部トレースの採取方法

サーバアプリケーショントレース(Java)および サーバアプリケーショントレース(C++)では主にオブジェクト管理部分のトレースを出力します。

これより低レベルの情報(スレッド管理、メッセージ管理、プロセス監視など)を表示する方法があります。かなりの量のトレースを出力するため、プロセスが起動しないときや、明らかに動作がおかしいとき、開発元に調査依頼を行う時などのみ採取すべきです。

以下にサーバアプリケーション内部トレースの採取方法を示します。

- ・ 統合運用管理ツールでプロセスグループを選択します。
- ・ [コマンドライン引数]タブを選択して、以下のように入力します。

-OltTpIlibTrace

- ・ 次に、`$(INSTANCE_ROOT)/config/tpsystem/プロセスグループ名.ped` (環境定義ファイル)を notepad や vi などのエディタで開きます。
- ・ 先ほど統合運用管理ツールで入力した`-OltTpplibTrace`を探してください(`ARGS -OltTpplibTrace`という行があるはずです)。この行の次に以下の二行を追加してください。この例では、`/tmp/servertrc.txt` に内部トレースを出力する為の設定です。Windows では `C:%temp%servertrc.txt` などに置き換えてください。

```
OUTPUTDEV /tmp/servertrc.txt
ERRORDEV /tmp/servertrc.txt
```

これで、`/tmp/servertrc.txt` に内部ログの初めの部分、続きはサーバアプリケーショントレースに出力されま  
す。ファイルに何か出力されていることを確認した後、まとめて開発元に送付してください。

ファイルの採取が終わりましたら、環境定義ファイルを元に戻してください。コマンドライン引数の  
`-OltTpplibTrace` も削除してください。

### 5.1.5. IIOPリスナ内部トレースの採取方法

IIOP リスナは普段はクライアントからの要求を適切なサーバアプリケーションに振り分ける処理を行います。もし、クライアントからの通信が行えなかったり、サーバアプリケーションに正しく要求が届かなかつたりする場合は、IIOP リスナのトレースを採取して開発元に送付してください。

これも、かなりの量のトレースを出力しますので、普段は採取しないようにしてください。

IIOP リスナは二つのトレースファイルから構成されます。一つは通信管理部分。もう一つは振り分け部分になります。

#### 通信管理部分のトレース

- ・ `$(INSTANCE_ROOT)/config/tpsystem/iiop.sg` を notepad か vi などのエディタで開いてください。ファイル中の説明を読んでトレース採取の有無、トレースファイル名、ファイルサイズを設定してください。このログは開発環境での障害発生時に採取を依頼する場合があります。通常の運用ではユーザは意識する必要はありません。

#### 振り分け部分のトレース

- ・ `$(INSTANCE_ROOT)/config/tpsystem/IIOPLsn.sg.sv` を `IIOPLsn.sg` に変更してください。これで、`$(INSTANCE_ROOT)/logs/tpsystem/システム名_IIOPLsn.log` に振り分け部分のトレースを出力します。ファイルサイズを変更する場合は、上記設定ファイル `IIOPLsn.sg` をファイル中の説明を読んで変更してください。このログは障害発生時に採取を依頼する場合があります。通常の運用ではユーザは意識する必要はありません。ログファイルのサイズが 32KB を超えると新しいログファイルが作成されます。古いログファイルは<ログファイル名.bak>として保存されます。ログは 2 世代で、世代数の変更はできません。

以上の設定が終わったら該当する TP システムを再起動してください。上記ファイルにファイルに何か出力されていることを確認した後、まとめて開発元に送付してください。

ファイルの採取が終わりましたら、全て元に戻してください。

## 6. 注意・制限事項について

### 6.1. 連続運用時のディスクフルについて

連続運用されるシステムを構築する場合、ローテートされないログファイルなどは、ファイルの容量増加によってディスクフルを起こす危険があります。ローテート設定可能なログについては設定の変更を、そうでないログファイルについては、定期的にログファイルを退避するなどして、ディスクフルが起きないように運用してください。ディスクフルの危険性があるログファイルについては「7.3 ディスクフルの危険性一覧」を参照してください。

### 6.2. アプリケーションがLog4Jを利用する場合の注意点

WebOTX 上で動作する Java アプリケーションにおいて、log4j を利用する場合の注意点について説明します。

#### 6.2.1. WebOTXにバンドルされているLog4Jを利用する場合

war ファイルに log4j を含めず、`#{INSTALL_ROOT}/lib/log4j.jar` を利用してログ出力を行う方法について説明します。

##### ・Web、Standard-J Edition の場合

アプリケーションで利用するログ定義は、`#{INSTANCE_ROOT}/config/log4j.xml` に追記してください。このとき、log4j.xml にデフォルトで定義されている内容は削除しないようにしてください。

##### ・Standard、Enterprise Edition の場合

プロセスグループ上で動作させるアプリケーションで利用するログ定義は、`#{INSTANCE_ROOT}/config/tpsystem/logconf/#{APGNAME}-#{PGNAME}/log4j.xml` に追記してください。このとき、log4j.xml でデフォルトとして定義されている内容は削除しないようにしてください。

#### 6.2.2. Webアプリケーションに含まれるLog4Jを利用する場合

WebOTX がバンドルしている log4j を利用せず、アプリケーションに含まれる log4j を利用する方法について説明します。アプリケーションが利用する log4j 定義ファイルは、WEB-INF/classes 以下に配置してください。Web アプリケーションのログ出力を行うには以下の手順を行ってください。

1. WebOTX に設定されている log4j.configuration の削除  
WebOTX で設定されている log4j.configuration 定義を削除します。

Windows 環境では以下のコマンドを実行してください。

```
otxadmin > delete-jvm-options  
-Dlog4j.configuration=file¥¥:////#{com.nec.webotx.instanceRoot}#{file.separator}config#{file.separator}log4j.xml
```

UNIX 環境では以下のコマンドを実行してください。

```
otxadmin > delete-jvm-options  
-Dlog4j.configuration=file¥¥:////#{com.nec.webotx.instanceRoot}#{file.separator}config#{file.separator}log4j.xml
```

Web アプリケーション内で log4j.properties を使用する場合は、`#{INSTANCE_ROOT}/config/log4j.xml` を削除してください。

2. war ファイルの配備  
war ファイルを WebOTX に配備してください。  
otxadmin > deploy #{WEBAPNAME}

3. nec-web.xml の確認  
war ファイルを配備すると、nec-web.xml が自動作成されます。nec-web.xml は以下にあります。  
`#{INSTANCE_ROOT}/applications/j2ee-modules/#{WEBAPNAME}/WEB-INF/nec-web.xml`  
nec-web.xml で `delegate=false` となっていることを確認してください。Servlet 2.4 仕様の WebAP を配備した場合は `delegate=true` となっているので、true の場合は、nec-web.xml を以下のように変更します。  
`<class-loader delegate="false"/>`  
変更後、配備前の Web アプリケーションの WEB-INF 以下に nec-web.xml をコピーし、war ファイルを再作成してください。
4. war ファイルの再配備  
2 で war ファイルを再作成した場合は、WebOTX に再配備してください。  
`otxadmin > deploy --force #{WEBAPNAME}`
5. ドメイン停止  
ドメインを停止してください。  
`otxadmin stop-domain domain1`
6. WebOTX にバンドルしている log4j を削除  
WebOTX にバンドルしている log4j を削除してください。  
`#{AS_INSTALL}/lib/log4j.jar` を削除してください(※)。
7. ドメイン起動  
ドメインを起動してください。  
`otxadmin start-domain domain1`

※ 統合運用管理ツールを使用している場合は、統合運用管理ツールの起動に log4j.jar が必要になります。そのため、WebOTX が `#{AS_INSTALL}/lib/log4j.jar` を優先してロードしてしまわないように、log4j.jar を削除せず WebOTX のインストールディレクトリ以外の場所に配置し、統合運用管理ツールからのみ参照できるように `#{AS_INSTALL}/bin/otxadmingui.bat` または `otxadmingui.sh` を以下のように変更してください。

#### ・Windows(otxadmin.bat)

##### 変更前

```
set CLASSPATH="%AS_INSTALL%\lib\admingui.jar";"%AS_INSTALL%\lib\log4j.jar";~::~
```

##### 変更後

```
set CLASSPATH="%AS_INSTALL%\lib\admingui.jar";"<log4j.jar を配置したディレクトリ>\log4j.jar";~::~
```

#### ・UNIX(otxadmingui.sh)

##### 変更前

```
CPATH=$AS_ICU_LIB/admingui.jar:$AS_ICU_LIB/log4j.jar:~::~
```

##### 変更後

```
CPATH=$AS_ICU_LIB/admingui.jar:<log4j.jar を配置したディレクトリ>/log4j.jar:~::~
```

## 6.2.3. マルチプロセスモードでWebアプリケーションのログ出力を行う場合

Standard、Enterprise Edition をマルチプロセスモードで動作させる場合の Web アプリケーションのログ出力の手順について説明します。

マルチプロセスモードで実行する場合、WebAP はプロセスグループ上で動作します。プロセスグループは、多重度を設定することができるため、ファイルにログ出力を行う場合、複数のプロセスが同じファイルに書き込みを行い、メッセージの欠損やローテート失敗によるメッセージの全損等の問題が発生する可能性

があります。そのため、以下のいずれかの手順で問題を回避する必要があります。

- Web アプリケーションのログをサーバアプリケーショントレースログに出力  
アプリケーションのログをサーバアプリケーショントレースログに出力するには、標準出力を利用します。プロセスグループ上で標準出力を行うと、以下のパスにあるサーバアプリケーショントレースログに出力されます。

```
${INSTANCE_ROOT}/logs/tpsystem/${APGNAME}/${PGNAME}/${PGNAME}.${PID}.log
```

- Web アプリケーションのログを個別のファイルに出力  
Web アプリケーションのログを個別のファイルに出力する場合、出力するログファイル名に、Web アプリケーションが動作するプロセスのプロセス ID を付与することで、多重度を設定した場合でもプロセス単位にログを出力することができます。

ログ出力の手順について説明します。

1. log4j.configuration の設定  
プロセスグループにはデフォルトで log4j.configuration が設定されています。デフォルトで定義しているファイルは以下となります。使用しない場合は、定義をプロセスグループから削除してください。

```
${INSTANCE_ROOT}/config/tpsystem/logconf/${APGNAME}-${PGNAME}/log4j.xml
```

2. log4j 定義ファイルの編集  
使用する log4j 定義ファイル(log4j.xml, log4j.properties 等)の編集を行います。
  - Web アプリケーションのログをサーバアプリケーショントレースログに出力させる場合  
Web アプリケーションが使用する Appender として、org.apache.log4j.ConsoleAppender 等の標準出力へ出力するアペンダを利用することで、サーバアプリケーショントレースログに Web アプリケーションのログを出力することができます。
  - Web アプリケーションのログを個別のファイルに出力する場合  
Web アプリケーションのログを出力するファイル名に、\${jp.co.nec.WebOTX.ProcessID}を使用することで、ファイル名にアプリケーションが動作するプロセスのプロセス ID を付与することができます。そのため、多重度を設定した場合でもプロセスごとにログファイルを分離することができます。  
例: プロセスグループのプロセス ID が 1234、ログファイル名を app\_\${jp.co.nec.WebOTX.ProcessID}.log とした場合、主力されるログファイル名は app\_1234.log となります。
3. プロセスグループの再起動  
プロセスグループを再起動することで、変更した log4j 定義ファイルの内容が反映されます。  
otxadmin > stop-pg --apgroup \${APGNAME} \${PGNAME}  
otxadmin > start-pg --apgroup \${APGNAME} \${PGNAME}

(注意) ログ定義ファイルについて

プロセスグループ上で WebAP を動作させる場合、Web アプリケーションが利用するログ定義ファイルを、同一プロセスグループ上に配備されたアプリケーションごとに分けて利用することはできません。プロセスグループで同一の log4j 定義ファイルを使用してください。

# 7. 参照

## 7.1. モジュールとロガーとアペンドの一覧

WebOTX が既定で設定しているモジュール、ロガーおよびアペンドについて説明します。

モジュール名	ロガー名	使用するアペンド
Admin	com.nec.webotx.enterprise.system.tools.admin	<a href="#">ADMINFILELOG</a>
Admin assistant	com.nec.webotx.enterprise.system.WOAdminAssistant	<a href="#">TPMMGRFILELOG</a>
Adm Listener	com.nec.webotx.enterprise.system.admListener	<a href="#">TPMMGRFILELOG</a>
Application Client Container	com.nec.webotx.enterprise.system.container.applclient	<a href="#">CONTAINERFILELOG</a>
Classloader	com.nec.webotx.enterprise.system.core.classloading	<a href="#">ADMINFILELOG</a>
Configuration	com.nec.webotx.enterprise.system.core.config	<a href="#">ADMINFILELOG</a>
CORBA	com.nec.webotx.enterprise.resource.corba	<a href="#">FILELOG</a>
Deployment	com.nec.webotx.enterprise.system.tools.deployment	<a href="#">DEPLOYFILELOG</a>
EJB Container	com.nec.webotx.enterprise.system.container.ejb	<a href="#">EJBCONTAINERFILELOG</a>
JavaBeans Activation Framework	com.nec.webotx.enterprise.system.activation	<a href="#">FILELOG</a>
JavaMail	com.nec.webotx.enterprise.resource.javamail	<a href="#">FILELOG</a>
JAXR	com.nec.webotx.enterprise.system.webservices.registry	<a href="#">FILELOG</a>
JAXRPC	com.nec.webotx.enterprise.system.webservices.rpc	<a href="#">FILELOG</a>
JAXWS	com.nec.webotx.enterprise.system.webservices.ws	<a href="#">WSFILELOG</a>
JDBC Data Source	com.nec.wojdbc	<a href="#">WOJDBCFILELOG</a>
JDO	com.nec.webotx.enterprise.resource.jdo	<a href="#">FILELOG</a>
JMS Client	com.nec.wojms	<a href="#">WOJMSFILELOG</a>
JMS Resource	com.nec.webotx.enterprise.resource.jms	<a href="#">FILELOG</a>
JMX Remote Access	com.nec.webotx.enterprise.system.remote_access	<a href="#">JMXREMOTEACCESSLOG</a>
JNDI	com.nec.webotx.enterprise.system.core.naming	<a href="#">NAMINGFILELOG</a>
JNDI Service	com.nec.webotx.enterprise.system.jndisp	<a href="#">JNDISPFILELOG</a>
JTA	com.nec.wojta	<a href="#">WOJTAFILELOG</a>
OADJ	com.nec.webotx.enterprise.system.ospij	<a href="#">OSPIFILELOG</a>
Object Broker	com.nec.webotx.enterprise.system.objectbroker	<a href="#">OSPIFILELOG</a>
Object Broker Java Library	com.nec.objectbrokerjava.lib	<a href="#">OSPIJLIBFILELOG</a>
Object Broker Java Message	com.nec.objectbrokerjava.msg	<a href="#">OSPIJLIBFILELOG</a>
Process Conductor	jp.co.nec.ebiz.bpel.engine.PROCESS_ENGINE_LOGGER	PROCESSENGINELOG
Root	com.nec.webotx.enterprise	<a href="#">FILELOG</a>
SAAJ	com.nec.webotx.enterprise.system.webservices.saaj	<a href="#">FILELOG</a>

Security	com.nec.webotx.enterprise.system.core.security	<a href="#">SECURITYFILELOG</a>	
Server	com.nec.webotx.enterprise.system	<a href="#">FILELOG</a>	
Server Core	com.nec.webotx.enterprise.system.core	<a href="#">FILELOG</a>	
Tools	com.nec.webotx.enterprise.system.tools	<a href="#">FILELOG</a>	
TPMonitor Manager	com.nec.webotx.enterprise.system.TPMonitorManager	<a href="#">TPMMGRFILELOG</a>	
Transaction Service	com.nec.webotx.enterprise.system.transactionservice	<a href="#">TSFILELOG</a>	
Util	com.nec.webotx.enterprise.system.util	<a href="#">ADMINFILELOG</a>	
Watch Service	com.nec.webotx.enterprise.system.watchservice	<a href="#">FILELOG</a>	
Web Container	com.nec.webotx.enterprise.system.container.web	<a href="#">WEBCONTAINERFILELOG</a>	
Web Server	com.nec.webotx.enterprise.system.webserver	<a href="#">WEBSVFILELOG</a>	

## 7.2. アペンダー一覧

WebOTX が既定で設定しているアペンダについて説明します。

### CONSLOG

コンソールに出力するためのアペンダです。通常は使用しません。

パラメータ		既定値	説明
appender	class	com.nec.webotx.org.apache.log4j.ConsoleAppender	アペンダのクラス名
	layout	com.nec.webotx.org.apache.log4j.PatternLayout	パターンレイアウトクラス
	ConversionPattern	"%d %-5p %c - %m [%t]%n"	レイアウトパターン

### EVENTLOG(Windows環境のみ有効)

イベントログに出力するためのアペンダです。EVENTLOGアペンダを利用した場合、SLOGINFOレベル以上のメッセージのみ出力されます。

パラメータ		既定値	説明
appender	class	com.nec.webotx.logging.OTXNTEventLogAppender	アペンダのクラス名
	Source	WebOTX AS	イベントログソース名
layout	class	com.nec.webotx.org.apache.log4j.PatternLayout	パターンレイアウトクラス
	ConversionPattern	"%m (%c)"	レイアウトパターン

### SYSLOG(UNIX環境のみ有効)

syslogに出力するためのアペンダです。SYSLOGアペンダを利用した場合、SLOGINFOレベル以上のメッセージのみ出力されます。なおLinux環境においてはシスログ出力を有効にするためにはsyslogdに'-r' オプション

をつけてください。'-r' オプションがないと出力されません。

パラメータ		既定値	説明
appender	class	com.nec.webotx.logging. OTXSyslogAppender	アペンダのクラス名
	SyslogHost	localhost	ホスト名
	Facility	user	ファンリティ
layout	class	org.apache.log4j. PatternLayout	パターンレイアウトクラス
	ConversionPattern	"%m (%c)"	レイアウトパターン

## NOTIFICATIONLOG

重要ログメッセージをJMX Notificationとして通知を行なうためのアペンダです。この通知は運用管理ツールのイベントビューに表示されます。

パラメータ		既定値	説明
appender	class	com.nec.webotx.logging. OTXNotificationLogAppender	アペンダのクラス名
layout	class	org.apache.log4j. PatternLayout	パターンレイアウトクラス
	ConversionPattern	"%m (%c)"	レイアウトパターン

## XXXXFILELOG

ログファイルに出力するためのアペンダです。

パラメータ		既定値	説明
appender	class	org.apache.log4j.RollingFileAppender	アペンダのクラス名
	threshold	WARN	出カレベル 以下のアペンダ UDDIFILELOG
		INFO	出カレベル 以下のアペンダ STRUTSFILELOG

TOMCATFILELOG

WEBCSTATVMFILELOG

WEBCSTATAPFILELOG

			WSFILELOG AXISFILELOG
	File	「 <a href="#">各アペンダに対応するファイル名一覧</a> 」を参照してください。	出力ファイル名
	MaxFileSize	1MB	ファイルサイズ上限
	MaxBackupIndex	2	ログファイルをバックアップする数
layout	class	“%c - %m [%t]”	パターンレイアウトクラス
	ConversionPattern		レイアウトパターン

### JMXREMOTEACCESSLOG

リモートアクセスログを出力するためのアペンダです。この通知は`${INSTANCE_ROOT}/logs/agent/`に `agent_access.log` として出力されます。

パラメータ		既定値	説明
appender	class	com.nec.webotx.org.apache.log4j.RollingFileAppender	アペンダのクラス名
	class	com.nec.webotx.org.apache.log4j.PatternLayout	パターンレイアウトクラス
layout	ConversionPattern	“%d %m %n”	レイアウトパターン

### 各アペンダに対応するファイル名一覧

アペンダ名	出力ファイル名	ロガー名
ACCESSFILELOGSERVER	server_access.log	server _otxadmin
ADMINFILELOG	webotx_admin.log	com.nec.webotx.enterprise.system.util com.nec.webotx.enterprise.system.tools.admin com.nec.webotx.enterprise.system.core com.nec.webotx.enterprise.admin org.apache.commons
BPLOG	bizbp_engine.log	jp.co.nec.ebiz.bpel org.springframework org.apache.wsif
CATALINAFILELOG	webotx_catalina.log	org.apache.catalina tomcat
CHDLFILELOG	webotx_chdl.log	com.nec.jp.rws.chdl
CONTAINERFILELOG	webotx_container.log	com.nec.webotx.enterprise.

		system.container	
DEPLOYFILELOG	webotx_deploy.log	com.nec.webotx.enterprise.system. tools.deployment	
EJBCONTAINERFILELOG	webotx_ejbcont.log	com.nec.webotx.enterprise. system.container.ejb	
ESBDBLOG	-	jbi.messaging.melog.db	
ESBFILELOG	webotx_esb.log	com.nec.webotx.jbi	
ESBFILEDUMPLOG	Dump[ID].log	jbi.messaging.melog.file.dump	
ESBFILETRACELOG	me_trace.log	jbi.messaging.melog.file.trace	
EVENTLOG	-	com.nec.webotx	
FILELOG	webotx_agent.log	com.nec.webotx org.apache.tomcat org.apache.coyote org.apache.jasper org.apache.jk javax	
JNDISPFILELOG	webotx_jndisp.log	com.nec.webotx.enterprise.system.jndisp	
JMXREMOTEACCESS	agent_access.log	com.nec.webotx.enterprise.system.remote_access	
NAMINGFILELOG	webotx_naming.log	com.nec.webotx.enterprise.system.core.naming	
OSPIFILELOG	webotx_ospi.log	com.nec.webotx.enterprise.system.ospij com.nec.webotx.enterprise.system.objectbroker	
OSPIJLIBFILELOG	objava.log	com.nec.objectbrokerjava objectbrokerjava.lib objectbrokerjava.msg	
PROCESSENGINELOG	bizbp_processinstance.log	jp.co.nec.ebiz.bpel.engine. PROCESS_ENGINE_LOGGER	
SECURITYFILELOG	webotx_security.log	com.nec.webotx.enterprise.system.core.security	
SIPSTACKFILELOG	webotx_sipstack.log	com.nec.webotx.sipstack.core.LogWriter	
SIPSTACKTRCLOG	webotx_sipstack.trc	com.nec.webotx.sipstack.javax.sip.stack.ServerLog	
STDOUT	server.log (※1)	org.apache.commons org.apache.struts com.opensymphony.oscache java.sql.Connection com.ibatis	

		org.apache.axis httpclient.wire	
	\$(PGNAME).\$(PID).log (※2)	org.apache.commons org.apache.axis com.opensymphony.oscache java.sql.Connection com.ibatis httpclient.wire	
STRUTSFILELOG	webotx_struts.log	org.apache.struts	
SYSLOG	-	com.nec.webotx	
TPMMGRFILELOG	webotx_tpmmgr.log	com.nec.webotx.enterprise.system.TPMonitorManager com.nec.webotx.enterprise.system.admListener com.nec.webotx.enterprise.system.WOAdminAssistant	
TSFILELOG	webotx_ts.log	com.nec.webotx.enterprise.system.transactionservice	
UDDIFILELOG	webotx_uddi.log	com.nec.jp.uddi	
WEBCONTAINERFILELOG	webotx_webcont.log	com.nec.webotx.enterprise.system.container.web	
WEBCMSGFILELOG	webotx_webc_message.log	com.nec.webotx.webcontainer.core.MessageNotification	
WEBCSTATAPFILELOG	webotx_webc_stat_webap.log	com.nec.webotx.webcontainer.statistics.WebAPStatistics	
WEBCSTATVMFILELOG	webotx_webc_stat_vm.log	com.nec.webotx.webcontainer.statistics.JavaVMStatistics	
WEBSVFILELOG	webotx_websv.log	com.nec.webotx.enterprise.system.webserver	
WDCFILELOG	wowdc.log	com.nec.webotx.enterprise.system.wdc	
WOJDBCFILELOG	wojdbc.log	com.nec.wojdbc	
WOJMSFILELOG	wojms.log	com.nec.wojms.ra com.nec.wojms.client	
WOJTAFILELOG	wojta.log	com.nec.wojta	
WOTRACE (※3)	\$(PGNAME).\$(PID).log	javax com.nec.webotx com.nec.wojdbc com.nec.wojta com.nec.objectbrokerjava com.nec.objectbrokerjava.lib com.nec.objectbrokerjava.msg com.nec.wojms.ra	

		com.nec.wojms.client org.apache.tomcat org.apache.coyote org.apache.jasper org.apache.inside org.apache.jk org.apache.catalina org.apache.commons tomcat server com.nec.jp.rws.chdl com.nec.webotx.webservice.wsrn jbi.messaging.melog.file.trace jbi.messaging.melog.file.dump jbi.messaging.melog.db	
WSFILELOG	webservice.log	com.nec.webotx.webservice org.apache.axis org.apache.ws com.nec.webotx.enterprise.system.webservices.ws	
WSRMFILELOG	wsrn.log	com.nec.webotx.webservice.wsrn	

(※1)エージェントプロセスの標準出力は「server.log」にリダイレクトされています。そのため、STDOUT アペンダに出力した内容は「server.log」へ出力されます。

(※2) プロセスグループの標準出力は「\${PGNAME}.\${PID}.log」にリダイレクトされています。そのため、STDOUT アペンダに出力した内容は「\${PGNAME}.\${PID}.log」へ出力されます。

(※3)WOTRACE アペンダは WebOTX AS 向けに独自に作成されたアペンダで、プロセスグループでのみ使用可能なアペンダです。

### 7.3. ディスクフルの危険性一覧

連続運用時にログの容量増加に伴うディスクフルの危険性があるログファイルを以下の表に示します。

ログファイル名	対象 Edition				備考
	Web	Standard-J	Standard	Enterprise	
server.log	○	○	○	○	(*5)
ObLog.log	○	○	○	○	(*6)
Oad.log	○	○	○	○	(*6)
Interface Repository.log	○(*1)	○(*1)	○(*1)	○(*1)	(*6)

corbaloc.log	○(*1)	○(*1)	○(*1)	○(*1)	(*6)
namesv.log	○	○	○	○	(*6)
cnamesv.log	-	-	○(*2)	○	(*6)
message.log	○	○	○	○	(*6)
syslog.log	○	○	○	○	(*6)
mod_jk.log (*3)	○	○	○	○	(*5)
mod_jk-20.log	○	○	○	○	(*5)
isapi.log	○	○	○	○	
nsapi.log	○	○	○	○	
mod_jk_om.log	-	-	○	○	(*5)
mod_jk_om-20.log	-	-	○	○	(*5)
access_log(access.log)	○(*3)	○(*3)	○(*3)	○(*3)	(*5)
error_log(error.log) (*6)	○(*3)	○(*3)	○(*3)	○(*3)	(*5)
ssl_request_log(ssl_request.log)	○(*4)	○(*4)	○(*4)	○(*4)	(*5)
std.log	-	○	○	○	(*7)

(\*1) デフォルトでは起動しません。

(\*2) WebOTX Cluster と組み合わせた場合のみ出力されます。

(\*3) インストール時に「WebOTX Web サーバ(Apache)」を選択した場合のみ出力されます。

(\*4) HTTPS 使用時に出力されます。

(\*5) rotatelogs を利用することで回避可能です。

詳細は「4.3 rotatelogs を利用したログローテーションの設定方法」を参照してください。

(\*6) 設定により回避可能です。詳しくは 3.8 節の各ログファイルの項目を参照してください。

(\*7) スレッドダンプ取得回数に依存します。