



BEA

WebLogic Server

インターナショナルライゼーション ガイド

BEA WebLogic Server 6.1
マニュアルの日付：2002 年 6 月 24 日

著作権

Copyright © 2002 BEA Systems, Inc. All Rights Reserved.

限定的権利条項

本ソフトウェアおよびマニュアルは、BEA Systems, Inc. 又は日本ビー・イー・イー・システムズ株式会社（以下、「BEA」といいます）の使用許諾契約に基づいて提供され、その内容に同意する場合にのみ使用することができ、同契約の条項通りにのみ使用またはコピーすることができます。同契約で明示的に許可されている以外の方法で同ソフトウェアをコピーすることは法律に違反します。このマニュアルの一部または全部を、BEA からの書面による事前の同意なしに、複写、複製、翻訳、あるいはいかなる電子媒体または機械可読形式への変換も行うことはできません。

米国政府による使用、複製もしくは開示は、BEA の使用許諾契約、および FAR 52.227-19 の「Commercial Computer Software-Restricted Rights」条項のサブパラグラフ (c)(1)、DFARS 252.227-7013 の「Rights in Technical Data and Computer Software」条項のサブパラグラフ (c)(1)(ii)、NASA FAR 補遺 16-52.227-86 の「Commercial Computer Software--Licensing」条項のサブパラグラフ (d)、もしくはそれらと同等の条項で定める制限の対象となります。

このマニュアルに記載されている内容は予告なく変更されることがあり、また BEA による責務を意味するものではありません。本ソフトウェアおよびマニュアルは「現状のまま」提供され、商品性や特定用途への適合性を始めとする（ただし、これらには限定されない）いかなる種類の保証も与えません。さらに、BEA は、正当性、正確さ、信頼性などについて、本ソフトウェアまたはマニュアルの使用もしくは使用結果に関していかなる確約、保証、あるいは表明も行いません。

商標または登録商標

BEA、Jolt、Tuxedo、および WebLogic は BEA Systems, Inc. の登録商標です。BEA Builder、BEA Campaign Manager for WebLogic、BEA eLink、BEA Manager、BEA WebLogic Collaborate、BEA WebLogic Commerce Server、BEA WebLogic E-Business Platform、BEA WebLogic Enterprise、BEA WebLogic Integration、BEA WebLogic Personalization Server、BEA WebLogic Process Integrator、BEA WebLogic Server、E-Business Control Center、How Business Becomes E-Business、Liquid Data、Operating System for the Internet、および Portal FrameWork は、BEA Systems, Inc. の商標です。

その他の商標はすべて、関係各社がその権利を有します。

インターナショナルライゼーション ガイド

マニュアルの版数	日付	ソフトウェアのバージョン
なし	2002 年 6 月 24 日	BEA WebLogic Server バージョン 6.1

目次

このマニュアルの内容

対象読者.....	viii
e-docs Web サイト.....	viii
このマニュアルの印刷方法.....	viii
関連情報.....	ix
サポート情報.....	ix
表記規則.....	x

1. WebLogic Server のインターナショナルライゼーションの概要

ログ メッセージについて.....	1-1
ローカライゼーションについて.....	1-2
インターナショナルライゼーションのための Java インタフェース.....	1-3

2. BEA WebLogic Server でのメッセージ カタログの使い方

メッセージ カタログの種類.....	2-2
メッセージ引数の挿入.....	2-2
メッセージ カタログの名前の選択.....	2-3
メッセージ カタログの階層.....	2-4
メッセージ カタログのフォーマット.....	2-4
ログ メッセージ カタログの要素.....	2-5
message_catalog.....	2-5
log_message.....	2-6
ログ メッセージ カタログの他の要素.....	2-7
ログ メッセージ カタログの構文.....	2-8
シンプル テキスト メッセージ カタログのフォーマット.....	2-9
message_catalog.....	2-9
message.....	2-10
messagebody.....	2-11

シンプル テキスト カタログの例	2-12
ロケール固有のカタログのフォーマット	2-12
3. BEA WebLogic Server のインターナショナルライゼーション インターフェイスとローカライゼーション インタフェースの 使い方	
インターナショナルライゼーション インタフェースの使い方	3-1
ローカライゼーション インタフェースの使い方	3-2
4. BEA WebLogic Server インターナショナルライゼーション ツールおよびユーティリティの使い方	
メッセージ エディタの使い方	4-1
メッセージ エディタの起動	4-2
既存のカタログの編集	4-4
新規カタログの作成	4-4
新規ログ メッセージの入力	4-5
新規シンプル テキスト メッセージの入力	4-6
ログ メッセージの検索	4-6
シンプル テキスト メッセージの検索	4-7
カタログのすべてのメッセージの表示	4-7
カタログ リストからの編集用メッセージの選択	4-8
複数のカタログのすべてのメッセージの表示	4-8
表示されている任意のカタログからの編集用メッセージの選択	4-8
既存のメッセージの編集	4-9
インターナショナルライゼーション ユーティリティの使い方	4-9
i18ngen	4-9
構文	4-10
オプション	4-10
l10ngen	4-11
構文	4-11
オプション	4-12
CatInfo	4-12
構文	4-12
オプション	4-13
5. ローカライゼーションを考慮したコードのインスツルメン	

ト

メッセージのロギング	5-1
新規カタログの作成	5-2
新規メッセージの作成	5-3
メッセージの修正	5-3

A. BEA WebLogic Server 用の Localizer クラスのリファレンス

ルックアップ プロパティ	A-3
--------------------	-----

索引



このマニュアルの内容

このマニュアルでは、インターナショナルライゼーションとローカライゼーションについて定義し、WebLogic Server で提供されるテンプレートおよびツールを使用して、ロケール固有のメッセージ カタログを作成または編集する方法について説明します。

このマニュアルの内容は以下のとおりです。

- [第 1 章「WebLogic Server のインターナショナルライゼーションの概要」](#)では、インターナショナルライゼーションとローカライゼーションに必要なプロセスの概要について説明します。
- [第 2 章「BEA WebLogic Server でのメッセージ カタログの使い方」](#)では、メッセージ カタログの種類と、メッセージ定義、要素、および引数について説明します。
- [第 3 章「BEA WebLogic Server のインターナショナルライゼーション インタフェースとローカライゼーション インタフェースの使い方」](#)では、インタフェースの使い方の概要について説明します。
- [第 4 章「BEA WebLogic Server インターナショナルライゼーション ツールおよびユーティリティの使い方」](#)では、WLS に付属のインターナショナルライゼーション ツールの使い方について説明します。
- [第 5 章「ローカライゼーションを考慮したコードのインストールメント」](#)では、コード インストールメンテーションを使用して、新規のカタログを作成し、メッセージをロギング、作成、および変更する方法について説明します。
- [付録 A「BEA WebLogic Server 用の Localizer クラスのリファレンス」](#)

対象読者

このマニュアルは、ロケール固有の管理のために、WLS 配布キットに含まれるメッセージカタログをインターナショナルライズまたはローカライズする必要があるアプリケーション開発者を主な対象としています。WebLogic Server プラットフォームおよび Java プログラミングに読者が精通していることを前提として書かれています。

e-docs Web サイト

BEA 製品のドキュメントは、BEA の Web サイトで入手できます。BEA のホームページで [製品のドキュメント] をクリックするか、または WebLogic Server 製品ドキュメント ページ (<http://edocs.beasys.co.jp/e-docs/wls61>) を直接表示してください。

このマニュアルの印刷方法

Web ブラウザの [ファイル | 印刷] オプションを使用すると、Web ブラウザからこのマニュアルを一度に 1 章ずつ印刷できます。

このマニュアルの PDF 版は、Web サイトで入手できます。PDF を Adobe Acrobat Reader で開くと、マニュアルの全体 (または一部分) を書籍の形式で印刷できます。PDF を表示するには、WebLogic Server ドキュメントのホームページを開き、[ドキュメントのダウンロード] をクリックして、印刷するマニュアルを選択します。

Adobe Acrobat Reader は、Adobe の Web サイト (<http://www.adobe.co.jp/>) で無料で入手できます。

関連情報

以下の WebLogic Server のマニュアルには、idltojava コンパイラの使い方、および WLE システムにおける Java CORBA アプリケーションの実装方法に関する情報が含まれています。

Java IDL および Java CORBA アプリケーションの一般情報については、以下のソースを参照してください。

- OMG Web サイト (<http://www.omg.org/>)
- Sun Microsystems, Inc. の Java サイト (<http://java.sun.com/>)

CORBA と分散オブジェクト コンピューティング、トランザクション処理、および Java の詳細については、<http://edocs.beasys.co.jp/e-docs/> の参考文献を参照してください。

サポート情報

BEA のドキュメントに関するユーザからのフィードバックは弊社にとって非常に重要です。質問や意見などがあれば、電子メールで docsupport-jp@bea.com までお送りください。寄せられた意見については、WebLogic Server のドキュメントを作成および改訂する BEA の専門の担当者が直に目を通します。

電子メールのメッセージには、ご使用のソフトウェアの名前とバージョン、およびドキュメントのタイトルと日付をお書き添えください。本バージョンの BEA WebLogic Server について不明な点がある場合、または BEA WebLogic Server のインストールおよび動作に問題がある場合は、BEA WebSUPPORT (www.bea.com) を通じて BEA カスタマ サポートまでお問い合わせください。カスタマ サポートへの連絡方法については、製品パッケージに同梱されているカスタマ サポート カードにも記載されています。

カスタマ サポートでは以下の情報をお尋ねしますので、お問い合わせの際はあらかじめご用意ください。

- お名前、電子メール アドレス、電話番号、ファクス番号
- 会社の名前と住所

- お使いの機種とコード番号
- 製品の名前とバージョン
- 問題の状況と表示されるエラー メッセージの内容

表記規則

このマニュアルでは、全体を通して以下の表記規則が使用されています。

表記法	適用
太字	用語集で定義されている用語を示す。
{ Ctrl } + { Tab }	複数のキーを同時に押すことを示す。
<i>斜体</i>	強調または書籍のタイトルを示す。
等幅テキスト	コード サンプル、コマンドとそのオプション、データ構造体とそのメンバー、データ型、ディレクトリ、およびファイル名とその拡張子を示す。等幅テキストはキーボードから入力するテキストも示す。 例： <pre>#include <iostream.h> void main () the pointer psz chmod u+w * \tux\data\ap .doc tux.doc BITMAP float</pre>
太字の等幅 テキスト	コード内の重要な箇所を示す。 例： <pre>void commit ()</pre>

表記法	適用
斜体の等幅テキスト	コード内の変数を示す。 例： <code>String expr</code>
すべて大文字のテキスト	デバイス名、環境変数、および論理演算子を示す。 例： LPT1 SIGNON OR
{ }	構文の中で複数の選択肢を示す。実際には、この括弧は入力しない。
[]	構文の中で任意指定の項目を示す。実際には、この括弧は入力しない。 例： <code>buildobjclient [-v] [-o name] [-f file-list]...[-l file-list]...</code>
	構文の中で相互に排他的な選択肢を区切る。実際には、この記号は入力しない。
...	コマンドラインで以下のいずれかを示す。 <ul style="list-style-type: none"> ◆ 引数を複数回繰り返すことができる ◆ 任意指定の引数が省略されている ◆ パラメータや値などの情報を追加入力できる 実際には、この省略符号は入力しない。 例： <code>buildobjclient [-v] [-o name] [-f file-list]...[-l file-list]...</code>
.	コード サンプルまたは構文で項目が省略されていることを示す。実際には、この省略符号は入力しない。

1 WebLogic Server のインターナショナルライゼーションの概要

インターナショナルライゼーション (I18N) とは、さまざまな地域で適切に動作するようにソフトウェアを準備することです。ローカライゼーション (L10N) とは、実行時にロケール固有の言語および構成を使用することです。このガイドでは、テキストデータの表示に関連する情報だけを提供します。

ログメッセージには、ログファイルに書き込まれるデータが含まれています。この極めて動的なデータには、アプリケーションやシステムの現在の状態に特定の情報が含まれています。ローカライズされたログメッセージカタログのテキストと結合すれば、このデータはユーザの言語でエラー状態を説明する、適切にフォーマットされたローカライズ済みのメッセージになります。

この章の内容は以下のとおりです。

- [ログメッセージについて](#)
- [ローカライゼーションについて](#)
- [インターナショナルライゼーションのための Java インタフェース](#)

ログメッセージについて

インターナショナルライズされたメッセージを作成するには、コードの変更や再コンパイルの必要なくさまざまなロケールに簡単に変換できるように、すべてのメッセージ文字列をメッセージカタログに外部化する必要があります。アプリケーション コードでは、現在のロケールに従ってコードとカタログ内のメッセージ文字列を結合し、ローカライズされたメッセージをログファイルに出力するロギングメソッドに実行時値を提供します。

メッセージ カタログ ファイル内で、各メッセージにはユニークなメッセージ ID とエラーに固有のメッセージ テキストが割り当てられます。原則的に、メッセージは、サポート担当者が簡単に見つけられるように、システム内の 1 つの場所からログに記録します。

インターナショナルライズされたメッセージは、以下のようにして作成されます。

1. メッセージ カタログでメッセージを定義します。詳細については、「[BEA WebLogic Server インターナショナルライゼーション ツールおよびユーティリティの使い方](#)」を参照してください。

カタログのエントリでは、メッセージ テキストだけでなく、メッセージが格納する実行時値の型と配置に関する情報も指定します。

2. `i18ngen` を実行します。i18ngen では、メッセージ カタログの有効性が検証され、カタログごとの `Logger` (Java クラス) が生成されます。生成されたクラスでは、メッセージ カタログのエントリで指定された情報に従ってメッセージごとのメソッドが定義されます。
3. 特定のメッセージをログに記録する必要がある場合、アプリケーション コードでは `Logger` から適切なメソッドを呼び出し、実行時値を引数として渡さなければなりません。
4. `Logger` のメソッドでは、実行時値とカタログ内のメッセージ文字列が結合され、WebLogic Server のログ ファイルにローカライズされたエントリが作成されます。

ロギング サブシステムの概要やログ メッセージの要素の説明を含めた詳細については、『[管理者ガイド](#)』の「[ログ メッセージを使用した WebLogic Server の管理](#)」を参照してください。

ローカライゼーションについて

ローカライゼーションでは、言語だけでなく、照合、日付と時刻の書式、通貨の書式、および文字コードも対象となります。WebLogic Server のエラー ログに記録されるメッセージはローカライズできます。

WebLogic Server のインターナショナルライゼーションでは、以下の 2 種類のデータのローカライゼーションがサポートされます。

- **ログメッセージ** - ログメッセージはサーバ ログに書き込まれる通知メッセージであり、適切なメッセージ引数がメッセージの定義で指定されている場合はエラー メッセージも含まれる
- **シンプル テキスト** - サーバで表示しなければならないログメッセージや例外以外のテキスト（ユーティリティからの出力など）、シンプル テキストの例としてはヘルプメッセージ、GUI ラベル、エラー メッセージなどがある

インターナショナルライゼーションのための Java インタフェース

Java インターナショナルライゼーション インタフェースを使用する場合は、Java Developer's Kit (JDK) に含まれている以下のインタフェースについて理解しておく必要があります。

<code>java.util.Locale</code>	地理的、政治的、または文化的に区別される特定の地域を表す
<code>java.util.ResourceBundle</code>	ロケール固有のオブジェクトが格納されるコンテナ
<code>java.text.MessageFormat</code>	言語に依存しない方法で連結されたメッセージを生成する手段

インターナショナルライゼーション カタログは、次のディレクトリに XML ファイルとして格納します。

```
weblogic\msgcat
```

ローカライゼーション カタログは、`weblogic\msgcat` のサブディレクトリに XML ファイルとして格納します（「メッセージ カタログの階層」を参照）。これらの XML カタログは、ビルド プロセスでクラスにコンパイルされます。生成されたクラスのメソッドは、実行時にメッセージをログに記録するために使用されるオブジェクトです。

メッセージカタログは、テキストメッセージのコレクションの説明が格納される XML ファイルです。各メッセージには、ユニークな識別子が付けられます。基本ロケールのカタログは、`msgcat.dtd` の文書型定義 (DTD) に基づいて定義します。基本カタログのさまざまなローカライゼーションを提供するカタログは、ロケールに基づく名前 (たとえば日本の場合は `weblogic\msgcat\ja`) の付いた `msgcat` のサブディレクトリで定義し、`l10n_msgcat.dtd` を使用します。

2 BEA WebLogic Server でのメッセージ カタログの使い方

インターナショナル化するテキストはすべてメッセージ カタログで定義します。各メッセージ カタログでは、ログ メッセージまたはシンプル テキストのコレクションが定義されます。

メッセージ カタログは、文書型定義 (DTD) で定義された XML ファイルであり、テキスト メッセージを定義します。各カタログでは、システムの一部で使用されるログ メッセージまたはシンプル テキストが定義されます。ユーザは、ロギングのすべての必要条件に対応する 1 つのログ メッセージ カタログを作成するか、サブシステムまたは Java パッケージごとに細かく分けてカタログを作成することができます。BEA では、後者の方法を推奨しています。複数のサブシステムを使用すると、ログを表示するときに特定の部分に焦点を当てることができます。

シンプル テキストのカタログの場合は、インターナショナル化されるユーティリティごとに 1 つのカタログを作成することをお勧めします。「[BEA WebLogic Server インターナショナルイゼーション ツールおよびユーティリティの使い方](#)」で説明されているツールを使用すれば、特定サイト用のメッセージ カタログを作成することができます。

すべてのメッセージと例外は、デフォルトの最上位カタログで定義する必要があります。WLS は、カタログのコレクションと共に配布されます。それらのカタログは、weblogic.jar ファイルの \weblogic\msgcat の中に配置されています。

この章では、メッセージ カタログに関連する以下のトピックについて説明します。

- [メッセージ カタログの種類](#)
- [メッセージ 引数の挿入](#)
- [メッセージ カタログの名前の選択](#)
- [メッセージ カタログの階層](#)

- [メッセージ カタログのフォーマット](#)

メッセージ カタログの種類

weblogic\msgcat ディレクトリには、メッセージ カタログの構文を記述する DTD も格納されています。WLS 配布キットには、以下の 2 つの DTD が含まれています。

- msgcat.dtd - 最上位のデフォルト カタログの構文を記述します。このカタログにはロケール固有のカタログにはない情報（メッセージの重要度など）が含まれます。
- l10n_msgcat.dtd - ロケール固有のカタログの構文を記述します。

このディレクトリには、最上位のメッセージ カタログおよびロケール固有のメッセージ カタログを作成するために使用できるテンプレートもあります。

メッセージ引数の挿入

java.text.MessageFormat で規定されているとおり、ログ メッセージのすべてのセクションではメッセージ引数を挿入できます。メッセージでは、0 ~ 9 で番号付けされた 10 個までの引数をサポートできます。それらの引数のサブセットを、メッセージ定義のどのテキスト セクションにでも挿入できます。メッセージ引数は開発時にメッセージの定義に挿入され、実行時においてメッセージがログに記録されるときに適切なメッセージ コンテンツに置き換えられます。

ある XML ログ メッセージ定義から抜粋した次の部分は、メッセージ引数の使い方を示しています。引数の番号は、*method* 属性で指定された引数の 1 つと対応していなければなりません。具体的には、{0} は最初の引数と、{1} は 2 番目の引数と対応している必要があります。

```
<messagebody>Unable to open file, {0}.</messagebody>
  <messagedetail>
    File, {0} does not exist.The server will restore the file
    contents from {1}, resulting in the use of default values
    for all future requests.
  </messagedetail>
```

```
<cause>The file was deleted</cause>
<action>
If this error repeats then investigate unauthorized access to
the file system.
</action>
```

次に、上のメッセージの `method` 属性の例を示します。

```
-method="logNoFile(String name, String path)"
```

このメッセージでは、2つの引数 {0} と {1} が使用されます。

- {0} はメッセージ本文で使用されます。
- 両方ともメッセージの詳細で使用されます。
- どちらも `<cause>` セクションまたは `<action>` セクションでは使用されません。

また、引数は文字列であるか、文字列として表現可能である必要があります。数値データは {n,number} として表されます。日付は {n,date} のかたちでサポートされます。ログメッセージには、重要度レベルを割り当てなければなりません。ログメッセージは、`method` 属性で定義されているとおりに `Logger` のメソッドによって生成されます。

メッセージ カタログの名前の選択

メッセージ カタログ ファイルの名前 (.xml 拡張子なし) は、`Localizer` と `Logger` のクラス名を生成するために使用されます。したがって、名前は慎重に選択する必要があります。たとえば、メッセージ カタログの名前は以下の条件を満たしていなければなりません。

- 対象パッケージのどのクラスの名前とも衝突しない
- クラス名で使用できない文字が含まれていない
- クラス名の命名規約に従っている

たとえば、カタログの名前が `Xyz.xml` の場合、生成されるクラスの名前は `XyzLogLocalizer` および `XyzLogger` になります。

メッセージ ID は、通常は先頭に 0 が付く 6 文字の文字列です。一部のインタフェースでは整数表現もサポートされています。

パッケージ名は、特定のカタログが存在するサブシステムの名前と一致している必要があります。

ログの Localizer 「クラス」は、実際には ResourceBundle プロパティ ファイルです。

メッセージ カタログの階層

メッセージ カタログでは、複数のロケールまたは言語がサポートされています。特定のメッセージ カタログには、1 つのデフォルト バージョン（最上位カタログ）と、サポートされているロケールごとに 1 つのカタログが存在します。たとえば、mycat.xml という最上位カタログと ..\ja\mycat.xml というそのカタログの日本語版を使用するかもしれません。通常、最上位カタログは英語ですが、インストールされている WLS のカタログ以外のカタログで英語は必ずしも必要ではありません。

java.util.Locale のドキュメントで定義されているように、ロケールの指定（ja など）にも階層があります。簡単に言うと、ロケールでは言語、国、およびバリエーションを指定できます。言語が基本であり、言語は国コードで拡張できます。たとえば、en/US はアメリカ英語を示します。関連するカタログの名前は ..\en\US\mycat.xml になります。バリエーションはベンダまたはブラウザに固有であり、言語または国で定義された複数のロケールの間で細かな違い（照合順序など）を持たせるために使用します。

メッセージ カタログのフォーマット

最上位とロケール固有のカタログ ファイルでは、カタログ フォーマットがわずかに異なります。ロケール固有のカタログは、最上位バージョンで定義されているテキストの翻訳だけを備えています。さらに、ログ メッセージ カタログはシンプル テキスト カタログとは異なる方法で定義されます。それらの違いについては、以下の節で説明します。

- [ログ メッセージ カタログの要素](#)
- [シンプル テキスト メッセージ カタログのフォーマット](#)

- [ロケール固有のカタログのフォーマット](#)

ログ メッセージ カタログの要素

この節では、ログ メッセージ カタログの以下の要素に関するリファレンス情報を提供します。

- [message_catalog](#)
- [log_message](#)
- [ログ メッセージ カタログの他の要素](#)

message_catalog

次の表では、message_catalog 要素で定義できる属性について説明します。

属性	説明
i18n_package	省略可能。このカタログの生成された Logger クラスを配置する Java パッケージ。クラスの名前はカタログ ファイルの名前に基づいて付けられる。たとえばカタログの名前が mycat.xml の場合、生成される Logger クラスの名前は <code>i18n_package.mycatLogger.class</code> になる。デフォルト値は <code>weblogic.i18n</code> 。
l10n_package	省略可能。このカタログの生成された LogLocalizer プロパティを配置する Java パッケージ。クラスの名前はカタログ ファイルの名前に基づいて付けられる。たとえばカタログの名前が mycat.xml の場合は、 <code>l10n_package.mycatLogLocalizer.properties</code> という名前のプロパティ ファイルが生成される。デフォルト値は <code>weblogic.i18n</code> 。
subsystem	必須。このカタログと関連付けられているサブシステムを識別する頭字語。サブシステムの名前はエラー ログに含まれ、メッセージの分離を目的として使用される。

属性	説明
version	必須。使用される msgcat.dtd のバージョンを指定する。書式は <i>n.n</i> であり、たとえば <code>version="1.0"</code> のように定義する。
baseid	省略可能。このカタログで使用される最低のメッセージ ID を指定する。構文は 0 ~ 9 を 6 つ並べたもの。WLS カタログのデフォルト値は 000000、ユーザ カタログのデフォルト値は 500000。
endid	省略可能。このカタログで使用される最高のメッセージ ID を指定する。構文は 0 ~ 9 を 6 つ並べたもの。WLS カタログのデフォルト値は 499999、ユーザ カタログのデフォルト値は 999999。

log_message

次の表では、log_message 要素で定義できる属性について説明します。

属性	説明
messageid	必須。このログ メッセージのユニークな識別子。識別子は、すべてのカタログにわたってユニークでなければならない。値は、baseid 属性と endid 属性で定義された範囲で指定する必要がある。これは message_catalog の子要素である。
datelastchanged	省略可能。このメッセージの修正の管理に便利な日付と時刻のスタンプ。日付はカタログで動作するユーティリティによって提供される。構文は次のとおり。 <code>Long.toString(new Date().getTime());</code>
severity	必須。ログ メッセージの重要度。「debug」 _⌘ 、「info」 _⌘ 、「warning」 _⌘ 、「notice」 _⌘ 、「error」 _⌘ 、「critical」 _⌘ 、「alert」 _⌘ 、または「emergency」のいずれかを指定する。ユーザ カタログで使用できるのは「debug」 _⌘ 、「info」 _⌘ 、「warning」 _⌘ 、および「error」 _⌘ だけ。

属性	説明
stacktrace	<p>省略可能。Throwable の引数としてスタック トレースを生成するかどうかを示す。true または false を指定できる。デフォルト値は true。値が true の場合はトレースが生成される。構文は次のとおり。</p> <pre>stacktrace="true"</pre>
method	<p>必須。このメッセージをログに記録するためのメソッド シグネチャ。実際には 2 つのメソッドが提供される。ここで指定されるメソッドと Throwable 引数が追加された同様のメソッド。</p> <p>構文は標準の Java メソッド シグネチャから修飾子、セミコロン、および拡張子を除いたもの。引数の型には Java プリミティブまたは Java クラスを使用できる。クラスは java.lang にない場合は完全修飾でなければならない。また、クラスは java.text.MessageFormat の規約に準拠していなければならない。たいいていのクラスは便利な toString() メソッドを備えている。</p> <p>引数には有効であればどのような名前でも指定できるが、argn (n は 0 ~ 9) の規約に従っていなければならない。指定できる引数は 10 個まで。各 argn について、「ログ メッセージ カタログの他の要素」で説明されているテキスト要素に少なくとも 1 つの対応するプレースホルダが存在しなければならない。プレースホルダの形式は {n}、{n,number}、または {n,date}。</p>

ログ メッセージ カタログの他の要素

要素	説明
messagebody	<p>必須。このメッセージの短い説明を示す文字列。この要素では 0 個以上のプレースホルダ {n} を使用できる。プレースホルダは、ログ メッセージがローカライズされる時に適切な引数で置き換えられる。これは log_message の子要素である。</p>

要素	説明
messagedetail	必須。イベントの詳しい説明を示す文字列。この要素では 0 個以上のプレースホルダ {n} を使用できる。プレースホルダは、ログメッセージがローカライズされるときに適切な引数で置き換えられる。これは log_message の子要素である。
cause	必須。問題の根本的な原因を説明する文字列。この要素では 0 個以上のプレースホルダ {n} を使用できる。プレースホルダは、ログメッセージがローカライズされるときに適切な引数で置き換えられる。これは log_message の子要素である。
action	必須。適切な解決策を説明する文字列。この要素では 0 個以上のプレースホルダ {n} を使用できる。プレースホルダは、ログメッセージがローカライズされるときに適切な引数で置き換えられる。これは log_message の子要素である。

ログメッセージ カタログの構文

次の例は、1 つのログメッセージを備えるログメッセージ カタログ MyUtilLog.xml を示しています。

```
<?xml version="1.0"?>
<!DOCTYPE message_catalog PUBLIC "weblogic-message-catalog-dtd"
    "http://www.weblogic.com/msgcat.dtd">
<message_catalog
  l10n_package="programs.utils"
  i18n_package="programs.utils"
  subsystem="MYUTIL"
  version="1.0"
  baseid="600000"
  endid="600100"
  <log_message
    messageid="600001"
    severity="warning"
    method="logNoAuthorization(String arg0, java.util.Date arg1,
      int arg2)"
    <messagebody>
      Could not open file, {0} on {1,date} after {2,number} attempts.
    </messagebody>
    <messagedetail>
      The configuration for this application will be defaulted to
      factory settings. Custom configuration information resides
      in file, {0}, created on {1,date}, but is not readable.
    </messagedetail>
    <cause>
```



```

The user is not authorized to use custom configurations. Custom
configuration information resides in file, {0}, created on
{1,date}, but is not readable.The attempt has been logged to
the security log.
</cause>
<action>
  The user needs to gain appropriate authorization or learn to
  live with the default settings.
</action>
</log_message>
</message_catalog>

```

シンプルテキスト メッセージ カタログのフォーマット

この節では、ログ メッセージ カタログの以下の要素に関するリファレンス情報を提供します。

- [message_catalog](#)
- [message](#)
- [messagebody](#)

message_catalog

次の表では、`message_catalog` 要素で定義できる属性について説明します。

属性	説明
<code>l10n_package</code>	省略可能。このカタログの生成された <code>LogLocalizer</code> プロパティを配置する Java パッケージ。クラスの名前はカタログ ファイルの名前に基づいて付けられる。カタログ ファイルの名前が <code>mycat.xml</code> の場合は、 <code>l10n_package.mycatLogLocalizer.properties</code> という名前のプロパティ ファイルが生成される。デフォルト値は「 <code>weblogic.i18n</code> 」。
<code>subsystem</code>	必須。このカタログと関連付けられているサブシステムを識別する頭字語。サブシステムはエラー ログに含まれ、メッセージの分離を目的として使用される。

属性	説明
version	必須。使用される msgcat.dtd のバージョンを指定する。「1.0」以上でなければならない。

message

次の表では、message 要素で定義できる属性について説明します。

属性	説明
messageid	必須。このログ メッセージのユニークな識別子 (英数字の文字列)。識別子はこのカタログのコンテキストの中でユニークであれば良い。message は message_catalog の子要素である。
datelastchanged	省略可能。このメッセージの修正の管理に便利な日付と時刻のスタンプ。

属性	説明
method	<p>省略可能。このメッセージをフォーマットするためのメソッドシグネチャ。</p> <p>構文は標準の Java メソッド シグネチャから戻り値の型、修飾子、セミコロン、および拡張子を除いたもの。戻り値の型は常に <code>String</code>。引数の型には Java プリミティブまたは Java クラスを使用できる。クラスは <code>java.lang</code> にはない場合は完全修飾でなければならない。また、クラスは <code>java.text.MessageFormat</code> の規約に準拠していなければならない。たいていのクラス引数は便利な <code>toString()</code> メソッドを備えており、対応する <code>MessageFormat</code> プレースホルダは文字列でなければならない(つまり <code>{n}</code> の形式)。引数には有効であればどのような名前でも指定できる。指定できる引数は 10 個まで。</p> <p>各引数について、次に説明する <code>messagebody</code> 要素に少なくとも 1 つの対応するプレースホルダが存在しなくてはならない。プレースホルダの形式は <code>{n}</code>、<code>{n,number}</code>、または <code>{n,date}</code>。</p> <p>例：</p> <pre>method="getNoAuthorization (String filename, java.util.Date creDate)"</pre> <p>この例は、次のように <code>TextFormatter</code> クラスのメソッドになる。</p> <pre>public String getNoAuthorization (String filename, java.util.Date creDate)</pre>

messagebody

メッセージと関連付けられたテキストです。

使い方 必須です。この要素では、0 個以上のプレースホルダ `{n}` を使用できます。プレースホルダは、ログメッセージがローカライズされるときに適切な引数で置き換えられます。`messagebody` は `message` の子要素です。

シンプル テキスト カタログの例

次の例は、1つのテキスト定義を備えるシンプル テキスト カタログ `MyUtilLabels.xml` を示しています。

```
<?xml version="1.0"?>
<!DOCTYPE message_catalog PUBLIC "weblogic-message-catalog-dtd"
    "http://www.weblogic.com/msgcat.dtd">
<message_catalog>
  l10n_package="programs.utils"
  i18n_package="programs.utils"
  subsystem="MYUTIL"
  version="1.0"
  <message>
    messageid="FileMenuTitle"
    <messagebody>
      File
    </messagebody>
  </message>
</message_catalog>
```

ロケール固有のカタログのフォーマット

ロケール固有のカタログは、最上位カタログのサブセットです。ロケール固有のカタログは、それらが表すロケールに基づいた名前のサブディレクトリに格納されます。ロケール固有のカタログで有効な要素と属性は以下のとおりです。

- `locale_message_catalog` 要素 - 最上位の定義の `message_catalog` と同じです。
- `message` 要素 - 最上位の定義の `message` と同じです。
- `messageid` 属性 - 最上位カタログに適用される同じルールに従います。関連する最上位カタログの `messageid` と一致しなければなりません。
- `messagebody`、`messagedetail`、`cause`、`action` - 最上位カタログに適用される同じルールに従います。

たとえば、`MyUtilLabels.xml` のメッセージのフランス語版は次のディレクトリに格納されます。

```
...\msgcat\fr\MyUtilLabels.xml
```

翻訳されたメッセージは次のようになります。

```
<?xml version="1.0"?>
<!DOCTYPE message_catalog PUBLIC
```

```
"weblogic-locale-message-catalog-dtd"
"http://www.weblogic.com/l10n_msgcat.dtd">
<locale_message_catalog
  l10n_package="programs.utils"
  i18n_package="programs.utils"
  subsystem="MYUTIL"
  version="1.0">
  <message>
    <messageid="FileMenuTitle">
      <messagebody> Fichier </messagebody>
    </message>
  </locale_message_catalog>
```

messagebody、messagedetail、cause、および action の各要素でテキストを入力するときには、有効な UTF-8 文字を生成するツールを使用するとともに、適切なキーボードのマッピングがインストールされている必要があります。MessageLocalizer ユーティリティはそのようなツールの 1 つです。

3 BEA WebLogic Server のインターナショナルライゼーション インタフェースとローカライゼーション インタフェースの使い方

この章では、インターナショナルライゼーションおよびローカライゼーションに使用するインタフェースについて説明します。

- [インターナショナルライゼーション インタフェースの使い方](#)
- [ローカライゼーション インタフェースの使い方](#)

インターナショナルライゼーション インタフェースの使い方

シンプルテキストベースのユーティリティのインターナショナルライゼーションは、それらのユーティリティが `Localizer` を使用してテキスト データにアクセスするように指定することで実現します。ただし、ロギングが目的の場合は、英語のテキストをログに書き込む従来の手段の代わりに、生成された `Logger` クラスを使用する必要があります。たとえば、`i18ngen` では、カタログ `xyz.xml` に対して適切なパッケージにクラス `xyzLogger` が生成されます。

もう1つ例を挙げると、たとえば `MyUtilLog.xml` カタログが使用される場合は、クラス `programs.utils.MyUtilLogger.class` が生成されます。カタログで定義されているログメッセージごとに、このクラスでは `method` 属性で定義されているとおりに静的なパブリック メソッドが格納されます。

ローカライゼーション インタフェースの使い方

TextFormatter クラスは、シンプル メッセージ カタログごとに生成されます。TextFormatter クラスには、カタログのローカライズおよびフォーマットされたテキストにアクセスするために使用する静的なメソッドがあります。このクラスは、メッセージ本文、プレースホルダ、および MessageFormat とのインタフェースを処理するコンビニエンス クラスです。フォーマット用のメソッドは、各メッセージ定義の method 属性を通じて指定されます。たとえば、メッセージの定義に属性 `method=getErrorNumber(int err)` が含まれている場合、生成される TextFormatter クラスは次のようになります。

```
package my.text;
public class xyzTextFormatter {
    public static String getErrorNumber(int err) {
        . . .
    }
}
```

上のメソッドを使用することは、次のコード例を実行することと同じです。

```
import my.text.xyzTextFormatter
. . .

xyzTextFormatter xyzL10n = new xyzTextFormatter();
System.out.println(xyzL10n.getErrorNumber(someVal));
```

この場合、someVal 引数が適切に挿入された状態で現在のロケールのメッセージテキストが出力されます。

2つのコンストラクタが用意されています。デフォルトのコンストラクタは、Java 仮想マシン (JVM) のデフォルトのロケールで使用します。もう1つのコンストラクタでは、JVM 全体のロケールを変更することなく別のロケールを指定できます。

4 BEA WebLogic Server インターナショナルライゼーション ツールおよびユーティリティの使い方

この章の内容は以下のとおりです。

- [メッセージ エディタの使い方](#)
- [インターナショナルライゼーション ユーティリティの使い方](#)

注意： カタログ定義のテキストでは読みやすいように書式設定文字（改行など）が使用される場合がありますが、それらの文字はパーサでは維持されません。テキスト データは、1 行の文字列に標準化されます。先頭と末尾のホワイト スペースはすべて削除されます。埋め込まれている改行は、単語の区切りを維持するために必要に応じてスペースに置き換えられません。タブはそのまま残されます。

メッセージ エディタの使い方

メッセージ エディタは、XML メッセージ カタログを編集するための GUI です。メッセージ エディタでは、XML カタログの作成と読み書きを行います。必要に応じて、XML カタログはテキスト エディタで直に編集することもできます。メッセージ エディタで可能な作業は以下のとおりです。

- XML メッセージ カタログを作成する
- メッセージを作成および編集する
- 1 つのカタログのすべてのメッセージを表示する
- 複数のカタログのすべてのメッセージを同時に表示する
- メッセージを検索する

■ カタログ エントリで XML の有効性を検証する

メッセージ エディタで作成または使用されている最中のカタログは、コンテキスト カタログと呼びます。

カタログ パーサでは、特殊文字の参照が認識され、それらは表 4-1 で示されているように変換されます。メッセージ エディタでは、特殊文字が認識され、文字参照を使用してそれらが書き換えられます。パーサでは、他の文字参照は認識されません。

表 4-1 特殊文字の参照

文字	文字コード
@	&#x26;
<	&#x27;
>	&#x27;
'	&#x27;
"	&#x27;

メッセージ エディタの起動

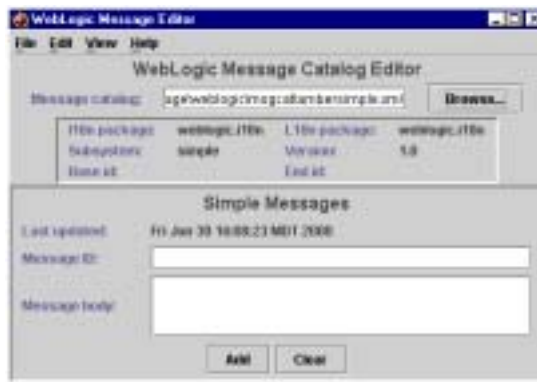
メッセージ エディタを起動するには、次のように入力します。

```
java weblogic.MsgEditor
```

ログ メッセージ用のメイン メッセージ エディタが次のように表示されます。



シンプルメッセージカタログを表示する場合、メッセージエディタは次のように表示されます。



既存のカタログの編集

既存のカタログを編集するには

- [Message Catalog] フィールドに絶対パス名を入力するか、または [Browse] ボタンをクリックして既存のカタログにアクセスします。

カタログは、`$WL_HOME\samples\examples\i18n\msgcat` というディレクトリで見つけることができます。カタログが見つかったら、そのパッケージ、サブシステム、バージョン、および開始 ID と終了 ID (ある場合) が表示され、そのカタログが他のすべてのアクションの実行対象であるカタログ コンテキストになります。この時点で、カタログでは、新しいメッセージの入力、既存のメッセージの編集、メッセージの検索、またはすべてのメッセージの表示を行うことができます。

新規カタログの作成

新規カタログを作成するには、次の手順を行います。

1. メイン メニュー バーから [File] メニューを選択します。
2. [New Catalog] を選択します。

次のように、[Create new catalog] ダイアログ ボックスが表示されます。



3. [Message Catalog] フィールドに絶対パス名を入力するか、または [Browse] ボタンをクリックして WebLogic カタログ ディレクトリ (`$WL_HOME\samples\examples\i18n\msgcat`) にアクセスします。

4. 新規カタログの名前を入力します。名前には、拡張子として `xml` を付ける必要があります。
5. ドロップダウン リストを使用して、カタログが**ログ メッセージ カタログ**または**シンプル テキスト メッセージ カタログ**のどちらであるのかを示します。ログ メッセージ カタログである場合は、**[Base ID]** フィールドと **[End ID]** フィールドが表示されます。シンプル テキスト メッセージ カタログの場合は、これらのフィールドは表示されません。
6. 残りのフィールドに値を入力して、**[Create Catalog]** をクリックします。ダイアログ ボックスが閉じ、作成したカタログがメッセージ エディタのメイン ウィンドウにコンテキスト カタログとして表示されます。

新規ログ メッセージの入力

メッセージを追加すると、カタログ全体がただちにディスクに書き込まれます。ログ カタログに新規メッセージを入力するには、次の手順を行います。

1. メッセージ ID を入力するか、または **[Get next ID]** ボタンをクリックしてコンテキスト カタログ内でユニークな次の ID を取得します。
2. 括弧と引数も含めて適切なメソッドを入力します。次に例を示します。

```
logNoAuthorization(String arg0, java.util.Date arg1,  
    int arg2) logNote()
```

3. リストから重要度を選択します。
4. メッセージの本文、詳細、原因、およびアクションを入力します。パラメータは `{n}` で示します。次に例を示します。

```
User {0} tried to access this on {1} but has no authority to  
do so.{2} lashes with a keyboard with coke spilled on it.
```

5. **[Add]** をクリックします。

メッセージが追加されると、カタログ全体がただちにディスクに書き込まれます。

新規シンプル テキスト メッセージの入力

メッセージを追加すると、カタログ全体がただちにディスクに書き込まれます。

1. メッセージの ID を英数字で入力します。

この番号はカタログ内でユニークでなければなりません。

2. メッセージ本文を入力します。

3. **[Add]** をクリックします。

メッセージが追加されると、カタログ全体がただちにディスクに書き込まれます。

ログ メッセージの検索

1. コンテキスト カatalogがログ カatalogであることを確認します。
2. メイン メニュー バーから **[Edit]** を選択します。
3. **[Search]** を選択して、検索ダイアログ ボックスを表示します。



4. 検索に必要なすべての情報を入力し、**[Find first]** または **[Find next]** をクリックします。

メッセージの検索は、フィールドの情報をつなぎ合わせて実行されます。メッセージ ID とメソッド名は完全一致が検索されますが、テキストに関してはすべてのテキスト フィールドで部分一致が検索されます。

一致するメッセージが見つかると、そのメッセージがメッセージ エディタのメイン ウィンドウに表示されます。表示されたメッセージは編集が可能です。

シンプル テキスト メッセージの検索

コンテキスト カatalogがシンプル テキスト メッセージ カatalogであることを確認します。

1. メイン メニュー バーから [Edit] を選択します。
2. [Search] を選択して、検索ダイアログ ボックスを表示します。



3. 検索に必要なすべての情報を入力し、[Find first] または [Find next] をクリックします。

メッセージの検索は、フィールドの情報をつなぎ合わせて実行されます。メッセージ ID は完全一致が検索されますが、テキストに関しては部分一致が検索されます。

一致するメッセージが見つかると、そのメッセージがメッセージ エディタのメイン ウィンドウに表示されます。表示されたメッセージは編集が可能です。

カatalogのすべてのメッセージの表示

カatalogのすべてのメッセージを表示するには、メイン メニュー バーから [View] を選択します。

現在のカatalog コンテキストのすべてのメッセージが、次のように [Message Viewer] ウィンドウに表形式で表示されます。



The screenshot shows a window titled "Message Viewer: JMS.xml" with a sub-header "Catalog: JMS.xml" and "Log Messages". Below this is a table with the following data:

Message Id	Method	Severity	Body	Detail	Action	Cause
040000	logInfoMsg(String arg0)	info	{0}	Uncatalogued info...	No action required.	Uncatalogued inform...
040001	logErrorMsg(String arg0)	error	{0}	Uncatalogued erro...	No action required.	Uncatalogued error...
040002	logDebugMsg(String arg0)	debug	{0}	Uncatalogued deb...	No action required.	Debugging

カタログ リストからの編集用メッセージの選択

[Message Viewer] ウィンドウで表示されているメッセージのいずれかをクリックすると、そのメッセージがメッセージ エディタのメイン ウィンドウに表示され、編集することができます。

複数のカタログのすべてのメッセージの表示

現在のカタログ コンテキストのメッセージを表示している状態で、新しいカタログを参照することによってコンテキストを変更すると、変更前のカタログの表示が画面に残ったまま新しいカタログが表示されます。この操作を繰り返すと、必要な（または画面に適切に表示できる）数のカタログを同時に表示できます。

表示されている任意のカタログからの編集用メッセージの選択

任意のビューで適当な行をクリックすると、カタログ コンテキストがその表示されたカタログに切り替わり、そのカタログのメッセージがメッセージ エディタのメイン ウィンドウに表示されます。

既存のメッセージの編集

メッセージを追加すると、カタログ全体がただちにディスクに書き込まれます。

1. 検索ダイアログ ボックスを使用するか、メッセージ ビューワーで行をクリックして、編集するメッセージを見つけます。
2. メッセージ エディタのメイン ウィンドウでフィールドを編集します。
3. [Update] をクリックします。

メッセージが追加されると、カタログ全体がただちにディスクに書き込まれます。

インターナショナルライゼーション ユーティリティの使い方

WebLogic Server では、以下の 3 つのインターナショナルライゼーション ユーティリティが用意されています。

- [i18ngen](#) - メッセージ カタログのパーサ
- [l10ngen](#) - ロケール固有のメッセージ カタログのパーサ
- [CatInfo](#) - 組み込まれているログ メッセージのリストを生成するユーティリティ

i18ngen

i18ngen ユーティリティでは、メッセージ カタログ (XML ファイル) が解析されて、ログ メッセージのテキストをローカライズするためのクラスが生成されます。i18ngen は、最上位のメッセージ カタログを Java クラスおよびプロパティ ファイルにコンパイルするために使用します。

構文

```
java weblogic.il8ngen [options] files
```

オプション

オプション	定義
<code>-d targetdirectory</code>	生成された Java ファイルをこの対象に送信する ユーザ カタログ プロパティは <code>-d</code> で指定された対象ディレクトリと相対的な <code>il8n_user.properties</code> に配置される。
<code>-n</code>	解析し、有効性を検証するが、クラスは生成しない。
<code>-keepgenerated</code>	生成された Java ソースを維持する。
<code>-ignore</code>	エラーを無視する。
<code>-il8n</code>	インターナショナルライザ (Logger など) を生成する。
<code>-l10n</code>	Localizer (LogLocalizer など) を生成する。
<code>-compile</code>	生成されたソースをコンパイルする。
<code>-nobuild</code>	解析と有効性の検証のみを実行する。
<code>files</code>	このファイル リストのファイルとディレクトリを処理する。

`il8ngen` ユーティリティでは、`il8n_user.properties` ファイルが作成または更新されます。

オプション `il8ngen -il8n` では、インターナショナルライズされたメッセージのロギングをサポートするインターナショナルライザ ソース (`*Logger.java` など) が作成されます。

オプション `i18ngen -l10n` では、各メッセージ カタログで定義されている各メッセージにアクセスできるようにするローカライザソース（プロパティリソースのまとめり）が作成されます。それらのプロパティは、メッセージをローカライズするためにローカライゼーション ユーティリティによって使用されません。

通常は、コンパイルの途中でエラーが検出されると、クラス ファイルやプロパティ ファイルが作成されません。`i18ngen` は異常な終了ステータスで終了します。

対象ディレクトリ（`-d` オプション）では、ソース ファイルが生成されるルートディレクトリを指定します。ファイルは、メッセージ カタログの `i18n_package` 値および `l10n_package` 値に基づいて適切なディレクトリに配置されます。デフォルトの対象ディレクトリはカレント ディレクトリです。

`-compile` オプションでは、現在の `CLASSPATH` を使用して、生成された Java ファイルがコンパイルされます。生成されたクラスは、`-d` オプションで識別されるディレクトリに配置されます。

`i18ngen` ユーティリティでは、コマンドラインで指定されたすべてのファイルが処理されます。ディレクトリが指定されている場合は、そのディレクトリのすべての XML ファイルが処理されます。すべてのファイルの名前には、XML のサフィックスが付いている必要があります。すべてのファイルは、`msgcat.dtd` の構文に準拠していなければなりません。`i18ngen` では、完全修飾された名前（Java ソース）のリストが `stdout` ログに出力されます（実際に生成されたファイルに関して）。エラーや警告などのメッセージはすべて `stderr` に送信されません。

I10ngen

`l10ngen` ユーティリティでは、最上位カタログの下位に位置するディレクトリ内のローカル固有のカタログが処理されます。

構文

```
java -classpath <l10n_Classpath> weblogic.i18ntools.l10ngen
[options] filelist
```

<libn_Classpath> には、<WebLogic Home>/lib/weblogic.jar が含まれていなければなりません。

オプション

オプション	定義
<code>-d targetdirectory</code>	プロパティを配置するディレクトリ。デフォルト値はカレントディレクトリ。
<code>-language code</code>	言語コード。デフォルト値は all。
<code>-country code</code>	国コード。デフォルト値は all。
<code>-variant code</code>	バリエーションコード。デフォルト値は all。
<code>filelist</code>	処理するメッセージカタログのディレクトリとファイル。カレントディレクトリを基準に指定する。ロケール固有のディレクトリではなく最上位のディレクトリを指定する。

CatInfo

このユーティリティでは、組み込まれているログメッセージのリストが生成されます。デフォルトの CatInfo では、現在組み込まれているすべてのログメッセージの ID とメッセージ本文が順番に表示されます。

構文

```
java weblogic.il8ntools.CatInfo [options]
```

オプション

注意: すべてのオプションは 1 文字に短縮できます。

オプション	定義
<code>-id nnnnnn</code>	<code>nnnnnn</code> はメッセージ ID を表す。 <code>-id</code> オプションは特定のメッセージを指定するために使用する。
<code>-subsystem identifier</code>	サブシステムの識別子。 <code>-subsystem</code> オプションでは指定されたサブシステムに一致するメッセージだけが出力される。
<code>-detail</code>	詳細なリストを要求する。 <code>-detail</code> オプションではバージョン、重要度、サブシステム、メッセージの詳細、原因、およびアクションといった情報も要求される。
<code>-help</code>	ヘルプ情報を提供する。

注意: メッセージの詳細なリストをファイルにエクスポートするには、以下の構文を使用します。

```
java weblogic.i18ntools.CatInfo -detail > Errors.txt
```

5 ローカライゼーションを考慮したコードのインスツルメント

この章では、以下のコード インスツルメンテーションについて説明します。

- [メッセージのロギング](#)
- [新規カタログの作成](#)
- [新規メッセージの作成](#)
- [メッセージの修正](#)

メッセージのロギング

次のサンプル コードは、`Stuff.xml` というログ メッセージを示しています。

```
<?xml version="1.0"?>
<!DOCTYPE message_catalog SYSTEM "msgcat.dtd">
<message_catalog
  i18n_package=weblogic.stuff
  subsystem=STF
  baseid="500000">
  <logmessage
    severity=warning
    messageid=501234
    method="logNoFile(String name)"
    <messagebody>
      Unable to open file, {0}
    </messagebody>
    <messagedetail>
      The file, {0}, does not exist. In its absence default
      values will be used and a new version of the file will
      be created automatically.
    </messagedetail>
    <cause>
      The file was deleted.
    </cause>
    <action>
      If this error repeats then investigate unauthorized
      access to the file system. Consider stiffening the
```

```
        permissions on the file and directory.
    </action>
</logmessage>
</message_catalog>
```

次のサンプルコードは、weblogic.stuff パッケージでどのようにメッセージ 501234 がログに記録されるのかを示しています。

weblogic.stuff パッケージのコードでは、次のようにメッセージ 501234 がログに記録されます。

```
package weblogic.stuff;
import weblogic.stuff.StuffLogger;
import java.io.FileInputStream;
. . .
try {
    FileInputStream fis = new FileInputStream(myfile);
    . . .
}
catch (FileNotFoundException fnfe) {
    StuffLogger.logNoFile(myfile);
}
```

新規カタログの作成

カタログを作成するには、メッセージ エディタまたはテキスト エディタを使用してファイルを作成します。メッセージ エディタの使い方については、[第 4 章「BEA WebLogic Server インターナショナルライゼーション ツールおよびユーティリティの使い方」](#)を参照してください。

カタログは XML ファイルであるため、そのフォーマットは msgcat.dtd で指定された構文に準拠している必要があります。ログ メッセージのテンプレートは、logmessage_template に用意されています。作成したカタログは、i18ngen で処理する必要があります。詳細については、[第 4 章「BEA WebLogic Server インターナショナルライゼーション ツールおよびユーティリティの使い方」](#)の「[新規カタログの作成](#)」を参照してください。

新規カタログを作成するときには、以下の事項に注意する必要があります。

1. ファイルの名前は Java クラスの命名規約に従います。
2. メッセージは番号順に追加します。
3. すべてのメッセージの ID は開始 ID よりも大きくなければなりません。

4. 開始 ID はカタログだけではなく `Msgids` ファイルでも指定します。

作成したカタログは、`i18ngen` で処理する必要があります。詳細については、第 4 章「BEA WebLogic Server インターナショナルライゼーション ツールおよびユーティリティの使い方」の「[新規カタログの作成](#)」を参照してください。

管理を容易にするには、すべてのカタログを同じソース ディレクトリに配置します。

新規メッセージの作成

新しいメッセージの定義は、最後に定義されているメッセージよりも 1 つ大きいメッセージ ID を使用してカタログの最後に追加します。カタログ内でユニークな `method` 名を選択します。新しい ID には、`Msgids` ファイルでそのカタログに指定されている範囲の ID を使用します。ID がすべて使用されている場合などで範囲内の ID を使用できない場合は、新しいカタログを作成するか、または `endid` 属性を使用して範囲を拡張する必要があります。

変更を行ったカタログは、`i18ngen` で処理する必要があります。詳細については、第 4 章の「[新規ログメッセージの入力](#)」を参照してください。

メッセージの修正

通常、既存のメッセージでは、メッセージ本文、メッセージの詳細、原因、およびアクション以外のデータは変更しません。このような制限が必要なのは、他のフィールドで値を変更すると、そのフィールドのコードが正常に機能しなくなる恐れがあるからです（不正な数または型の引数を渡してしまうなど）。

カタログの修正が終わった後は、`i18ngen` コマンドを実行してカタログを再処理します。

5 ローカライゼーションを考慮したコードのインストルメント

A BEA WebLogic Server 用の Localizer クラスのリファレンス

Localizer は、出力用にテキストをローカライズするためにアプリケーションやサーバコードによって使用されるクラスです。i18ngen ユーティリティでは、メッセージカタログの内容に基づいて Localizer クラスが作成されます。

Localizer クラスは、カタログファイルごとに 1 つ生成されます。クラスの名前は、カタログ名 (.xml 拡張子はユーティリティによって削除される) の後に LogLocalizer を付けたものです。カタログ ejb.xml の Localizer クラスは、ejbLogLocalizer です。

Localizer は、java.util.ListResourceBundle クラスを拡張したものです。ローカライゼーションデータのアクセスを容易にするために、4 つのメソッドが追加されています。それらのメソッドは以下のとおりです。

<code>public Object getObject(String key, String id)</code>	メッセージ「id」の「key」要素に対応するローカライゼーションテキストを返す。
---	--

<code>public Object getObject(String key, int id)</code>	メッセージ「id」の「key」要素に対応するローカライゼーションテキストを返す。
--	--

<code>public String getString(String key, String id)</code>	メッセージ「id」の「key」要素に対応するローカライゼーションテキストを返す。
---	--

<code>public String getString(String key, int id)</code>	メッセージ「id」の「key」要素に対応するローカライゼーションテキストを返す。
--	--

指定できるキーの値は、メッセージの属性に対応しています。それらは以下のとおりです。

- `Localizer.SEVERITY`
- `Localizer.MESSAGE_ID`
- `Localizer.MESSAGE_BODY`
- `Localizer.MESSAGE_DEATIL`
- `Localizer.MESSAGE_DETAIL`
- `Localizer.CAUSE`
- `Localizer.ACTION`

`Localizer.SEVERITY` キーの場合を除き、`Localizer` によって返されるローカライゼーション データは整数型のオブジェクトを返す `String` オブジェクトです。返される重要度の値は以下のとおりです。

- `weblogic.logging.severities.EMERGENCY`
- `weblogic.logging.severities.ALERT`
- `weblogic.logging.severities.CRITICAL`
- `weblogic.logging.severities.ERROR`
- `weblogic.logging.severities.WARNING`
- `weblogic.logging.severities.NOTICE`
- `weblogic.logging.severities.INFO`
- `weblogic.logging.severities.DEBUG`

返される特定の文字列は、メッセージ カタログで定義されます。

`get*()` メソッドの「key」引数は、定義のどの要素を返すのかを識別します。受け入れられる値は、`Localizer` クラスの定義で定義されます。返されるテキストは、`java.text.MessageFormat.format()` を通じてさらに拡張できます。当然、それはメッセージ本文、メッセージの詳細、原因、およびアクションの場合に限られます（これらはすべてローカライズ可能）。他の要素（メッセージ ID、重要度、およびサブシステム）はローカライズ可能ではなく、`MessageFormat` で処理を加える必要もありません。

ルックアップ プロパティ

メッセージの適切な Localizer を取得するために、L10nLookup クラスが用意されています。L10nLookup は、システムの起動時に次のプロパティ ファイルからロードされる Property クラスの拡張です。

```
\weblogic\msgcat\i18n.properties
```

このプロパティ ファイルは、i18ngen によって作成されます。ルックアップ ファイルのプロパティのフォーマットは次のとおりです。

```
nnnnnn=subsystem:Localizer class
```

この行の引数は次のように定義されています。

- *nnnnnn* はメッセージ ID を表します。
- *subsystem* は関連するサブシステムを表します。
- *Localizer class* は生成された Localizer クラスの名前を表します。

たとえばメッセージ 001234 は、次のルックアップ ファイルのプロパティに基づいて、weblogic.i18n.ejbLogLocalizer クラスからの EJB サブシステム メッセージ ID として識別されます。

```
001234=EJB:weblogic.i18n.ejbLogLocalizer
```


索引

い

印刷、製品のマニュアル 1-viii
インターナショナルライズされたメッセージ、作成 1-1
インターナショナルライゼーション 1-1
インターナショナルライゼーション、インタフェース 3-1
インターナショナルライゼーション ユーティリティ 4-9

か

カスタマ サポート情報 1-ix
カタログ、内容の表示 4-7
関連情報 1-ix

さ

サポート
技術情報 1-ix

し

新規カタログの作成 5-2
新規メッセージの作成 5-3
シンプル テキスト メッセージ、検索 4-7
シンプル テキスト メッセージ、新規入力 4-6

ま

マニュアル、入手先 1-viii

め

メッセージ エディタ 4-1
メッセージ エディタ、起動 4-2
メッセージ エディタ、新規カタログの作成 4-4
メッセージ エディタ、編集 4-4, 4-9
メッセージ カタログ、階層 2-4
メッセージ カタログ、種類 2-2
メッセージ カタログ、新規ログ メッセージ 4-5
メッセージ カタログ、シンプル テキスト 2-9
メッセージ カタログ、フォーマット 2-4
メッセージ カタログ、命名 2-3
メッセージの修正 5-3
メッセージのロギング 5-1
メッセージ引数 2-2

ゆ

ユーティリティ、catinfo 4-12
ユーティリティ、i18ngen 4-9
ユーティリティ、i10ngen 4-11

ろ

ローカライゼーション 1-1, 1-2
ローカライゼーション、インタフェース 3-2
ローカライゼーション、シンプル テキスト 1-3
ローカライゼーション、ログ メッセージ 1-3

ログ メッセージ 1-1
ログ メッセージ、検索 4-6
ログ メッセージ カタログ、構文 2-8
ログ メッセージ カタログ、要素 2-5
ロケール固有のカタログ 2-12