



BEA WebLogic Integration™

File Plug-in ユーザーズ ガイド

著作権

Copyright © 2003, 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・エー・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 Systems, Inc. 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 Commerce Server、BEA WebLogic Enterprise、BEA WebLogic Enterprise Platform、BEA WebLogic Express、BEA WebLogic Integration、BEA WebLogic Personalization Server、BEA WebLogic Platform、BEA WebLogic Portal、BEA WebLogic Server、BEA WebLogic Workshop および How Business Becomes E-Business は、BEA Systems, Inc の商標です。

その他の商標はすべて、関係各社が著作権を有します。

File Plug-in ユーザーズ ガイド

パート番号	日付	ソフトウェアのバージョン
なし	2003年2月	7.0 SP2

目次

このマニュアルの内容

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

1. File Plug-in の使用方法

File Plug-in について.....	1-1
File Plug-in の機能.....	1-1
File Plug-in かファイルアダプタかの選択.....	1-2
対象読者.....	1-3
File Plug-in とプラグインフレームワーク.....	1-3
File Plug-in の使用方法.....	1-4
File Plug-in アクションの概要.....	1-4
ワークフローの式.....	1-7
ファイルの読み込み.....	1-8
テンプレート定義.....	1-9
ファイルへの書き込み.....	1-10
ファイルの削除.....	1-11
ファイル検出時にワークフローを再開.....	1-12
ファイル作成の検出時にワークフローを開始.....	1-14
FTPを使用したファイルの取得と転送.....	1-16
移行したドメインのための File Plug-in のコンフィグレーション.....	1-18
BPM データベース テーブルの更新.....	1-18
単一サーバドメインへの移行.....	1-19
クラスタドメインへの移行.....	1-19

2. File Plug-in の使用例

ワークフローの設定.....	2-1
----------------	-----

ワークフローのアクティブ化.....	2-8
--------------------	-----

索引

このマニュアルの内容

このマニュアルでは、**File Plug-in** の使用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「**File Plug-in** の使用方法」では、**File Plug-in** の使用方法を説明します。
- 第 2 章「**File Plug-in** の使用例」では、**File Plug-in** の使用例として、ファイルをリモート ファイル システムに **FTP** で転送する手順をステップごとに説明します。

対象読者

このマニュアルは、ファイル システムと他のアプリケーション間のクライアント インタフェースを開発するワークフロー設計者およびシステム インテグレータを対象としています。

e-docs Web サイト

BEA 製品のドキュメントは、BEA Systems, Inc. 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「e-docs」という製品ドキュメント ページ (<http://edocs.beasys.co.jp/e-docs/index.html>) 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 ファイルずつ印刷できます。

このマニュアルの PDF 版は、WebLogic Integration の 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 Integration ドキュメントのホーム ページを開き、[PDF 版] ボタンをクリックして、印刷するマニュアルを選択します。

Adobe Acrobat Reader がない場合は、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 Integration ドキュメントには、この製品の使用に関連した情報が含まれています。

- *WebLogic Integration BPM ユーザーズ ガイド*
- *WebLogic Integration Studio ユーザーズ ガイド*

サポート情報

WebLogic Integration 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@beasys.com までお送りください。寄せられた意見については、WebLogic Integration 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 WebLogic Integration ドキュメントのリリース番号をお書き添えください。

本バージョンの **WebLogic Integration Worklist** について不明な点がある場合、または **Worklist** のインストールおよび動作に問題がある場合は、**BEA WebSupport** (<http://websupport.bea.com/custsupp>) 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
[Ctrl] + [Tab]	複数のキーを同時に押すことを示す。
斜体	強調または書籍のタイトルを示す。

表記法	適用
等幅テキスト	<p>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。</p> <p><i>例</i></p> <pre>#include <iostream.h> void main () the pointer psz chmod u+w * \tux\data\ap .doc tux.doc BITMAP float</pre>
斜体の等幅テキスト	<p>コード内の変数を示す。</p> <p><i>例</i></p> <pre>String expr</pre>
すべて大文字のテキスト	<p>デバイス名、環境変数、および論理演算子を示す。</p> <p><i>例</i></p> <pre>LPT1 SIGNON OR</pre>
{ }	<p>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。</p>
[]	<p>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。</p> <p><i>例</i></p> <pre>buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...</pre>
	<p>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。</p>

表記法	適用
...	<p>コマンドラインで以下のいずれかを示す。</p> <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。 <p>実際には、この省略記号は入力しない。</p> <p><i>例</i></p> <pre>buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...</pre>
.	<p>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。</p> <p>実際には、この省略記号は入力しない。</p>



1 File Plug-in の使用方法

File Plug-in は、BEA WebLogic Integration Business Process Management (BPM) の機能を拡張し、WebLogic Integration BPM Studio に追加機能を提供します。特に、File Plug-in には、ローカルとリモートのファイル システム間でファイル処理およびファイル転送を行うための基本アクションが用意されています。アクションは、ビジネスプロセス管理機能で実行する作業の基本単位であり、ワークフローとタスクの動作を定義します。

この章の内容は以下のとおりです。

- File Plug-in について
- File Plug-in の使用方法
- 移行したドメインのための File Plug-in のコンフィグレーション

File Plug-in について

この節の内容は以下のとおりです。

- File Plug-in の機能
- File Plug-in かファイル アダプタかの選択
- 対象読者
- File Plug-in とプラグイン フレームワーク

File Plug-in の機能

File Plug-in の機能は以下のとおりです。

- ファイルを読み込み、ファイル内容をワークフロー変数に割り当てる。

- ワークフロー変数の内容をファイルに書き込む。
- ファイルを削除する。
- ファイルの作成を契機としてワークフローを開始または再開する。
- FTP 経由でリモート ファイル システムとのファイル転送を行う。

File Plug-in を使用する場合、以下の点に注意する必要があります。

- **File Plug-In** にはファイル ロック サービスがありません。複数のスレッドから同じファイルを同時に変更することは避けてください。
- **File Plug-in** アクションはトランザクション対応ではなく、トランザクション対応のファイル オペレーションを自動的にロールバックする組み込みのメカニズムはありません。ファイル オペレーションのロールバックを実行するには、例外ハンドラでの調整アクションを使用します。例外ハンドラには、コミットとロールバック処理のパスがあります。例外ハンドラの詳細については、『*WebLogic Integration Studio ユーザーズ ガイド*』の「ワークフロー例外の処理」を参照してください。
- **File Plug-in** での読み込みおよび書き込みオペレーションに使用する **BinaryData** 型とファイル変換機能は、**Data Integration** プラグインによって提供されます。**Data Integration** プラグインは自動的に **WebLogic Integration** サーバにロードされます。

File Plug-in かファイル アダプタかの選択

通常、**File Plug-in** は、ファイルの読み込み、書き込み、削除や、ローカルシステムとリモート システム間でのファイル転送などの基本ファイル アクションに使用します。より高度な機能が必要な場合は、**WebLogic Adapter for File** を使用します。**WebLogic Adapter for File** では、追加ファイル オペレーションを実行し、Excel スプレッドシートの内容および **CSV (Comma Separated Variable)** フォーマットの読み込みと書き込みもできます。**File System Adaptor** の詳細については、「**BEA WebLogic Adapter for File**」を参照してください。

対象読者

このマニュアルは、ファイルシステムと他のアプリケーション間のクライアント インタフェースを開発するワークフロー設計者およびシステム インテグレータを対象としています。このマニュアルを読む前提として、ワークフロー設計とワークフロー テンプレート、ファイル転送プロトコル (FTP)、および **WebLogic Integration Studio** について十分に理解していることが必要です。さらに、**Web** 技術に関する知識、および **Microsoft Windows** と **UNIX** システムに関する一般知識も必要です。

ワークフローまたは **WebLogic Integration Studio** に関して前提となる知識が十分でない場合は、以下のマニュアルを参照してください。

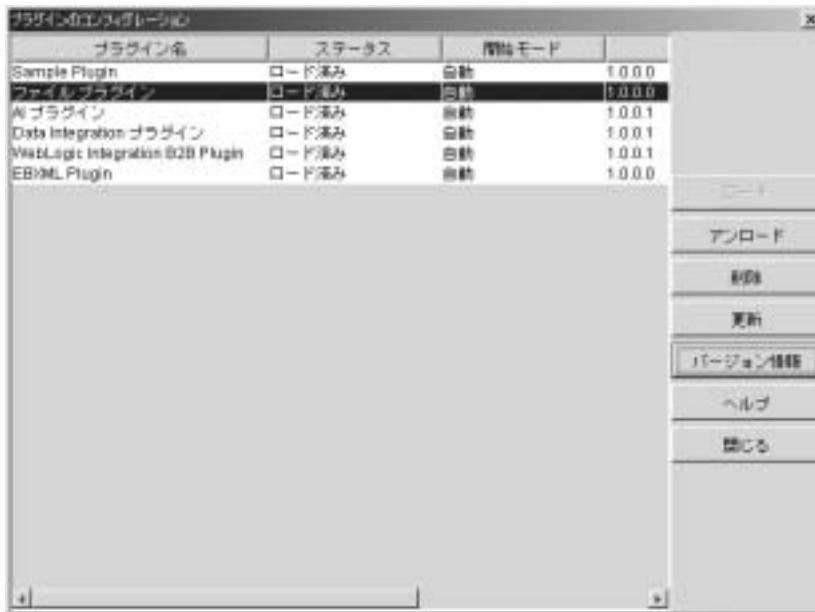
- 『*WebLogic Integration Studio ユーザーズ ガイド*』
(<http://edocs.beasys.co.jp/e-docs/wli/docs70/studio/index.htm>)
- 『*WebLogic Integration BPM ユーザーズ ガイド*』
(<http://edocs.beasys.co.jp/e-docs/wli/docs70/bpmtutor/index.htm>)

File Plug-in とプラグイン フレームワーク

他のプラグインと同様、File Plug-in は BPM 機能を拡張し、BPM Plug-in 仕様に準拠しています。詳細については、『*WebLogic Integration BPM プラグイン プログラミング ガイド*』

(<http://edocs.beasys.co.jp/e-docs/wli/docs70/devplug/index.htm>) を参照してください。[プラグインのコンフィグレーション] ウィンドウには、図 1-1 に示すように、プラグインと各プラグインのステータスが表示されます。

図 1-1 [プラグインのコンフィグレーション] ウィンドウ



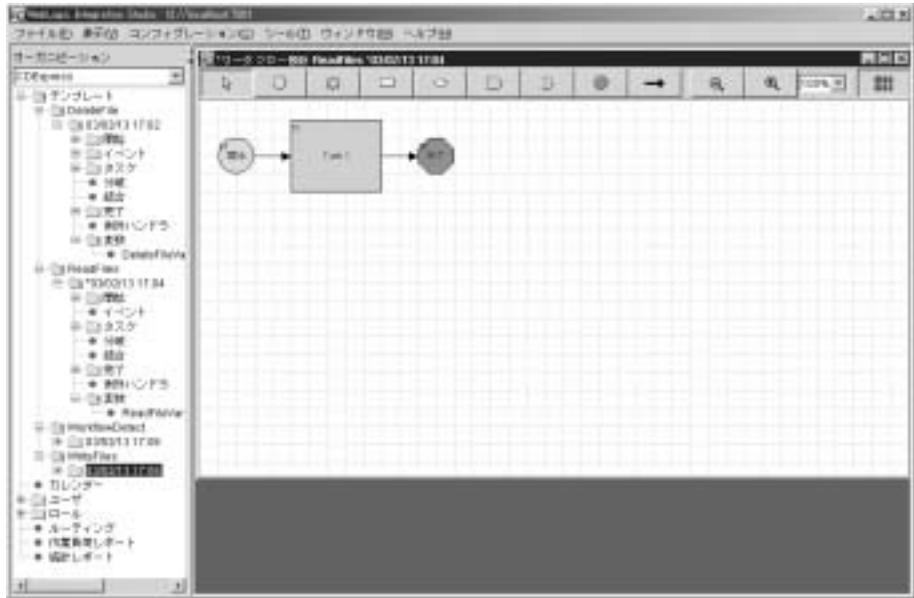
File Plug-in の使用方法

以下の節では、File Plug-in の使用方法を説明します。File Plug-in アクションを設定するステップごとの手順は含まず、WebLogic Integration ワークフローの設計に読者が精通していることを前提としています。ステップごとの手順については、第 2 章「File Plug-in の使用例」を参照してください。

File Plug-in アクションの概要

この節では、File Plug-in の使用に関する一般情報を示します。File Plug-in アクションを設定するために必要な Studio の機能を説明します。図 1-2 は、Studio で [ワークフロー設計] ウィンドウを開いたところを示しています。

図 1-2 WebLogic Integration Studio



[ワークフロー設計] ウィンドウでは、ワークフローを作成し、ワークフローを定義するプロパティにアクセスできます。

ほとんどの **File Plug-in** アクションには、ファイル内容を格納するための変数が必要です。ワークフロー アクションで使用する変数を定義するには、図 1-3 に示すように、[変数プロパティ] ウィンドウを使用します。

図 1-3 変数プロパティ



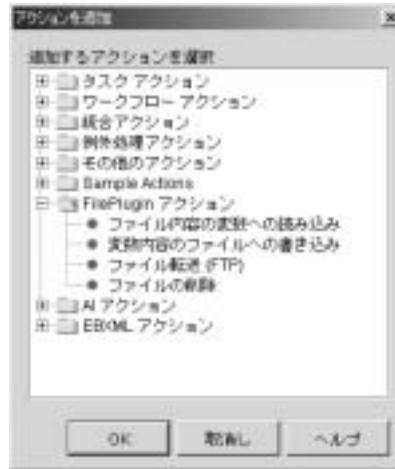
[ワークフロー設計] ウィンドウでタスク ノードを右クリックし、図 1-4 に示すような [タスクのプロパティ] ウィンドウを開いて、ワークフロープロパティを定義できます。

図 1-4 タスクのプロパティ



[追加] ボタンをクリックして [アクションを追加] ダイアログ ボックスを表示します。図 1-5 に示すように、[アクションを追加] ウィンドウでは多くの File Plug-in アクションにアクセスできます。

図 1-5 【アクションを追加】ウィンドウ



File Plug-in のウィンドウおよび Studio の他のウィンドウで使用できるアクションについては、この後の節で説明します。

ワークフローの式

Studio のウィンドウ内でフィールドの横に [式] ボタン  が表示される場合、そのフィールドにはワークフロー式の言語で定式化したエントリを入力する必要があります。File Plug-in アクション (ファイルの作成を契機としてワークフローを開始する場合を除く) で、式が要求される場合があるプロパティは以下のとおりです。

- [ファイル名]
- [ディレクトリ名]
- [時間間隔] - 整数を使用すること。
- [アーカイブ ディレクトリ]
- [FTP サーバ名]
- [リモート ファイル名]
- [ローカル ファイル名]

これらのプロパティの値は、文字列値を返す **Expression Builder** および **XPath Wizard** から取得できます。以下の値があります。

- 定数文字列 ("d:\\read\\read.xml" など)
- **String** 型のワークフロー変数 (\$file_name など)
- 文字列を返す複雑な式 (\$a+\$b+\$c+"a.txt" など)

注意: ファイル名またはディレクトリ名の文字列値には、そのファイルまたはディレクトリの絶対パスを指定する必要があります。パスには2種類の書式があります。

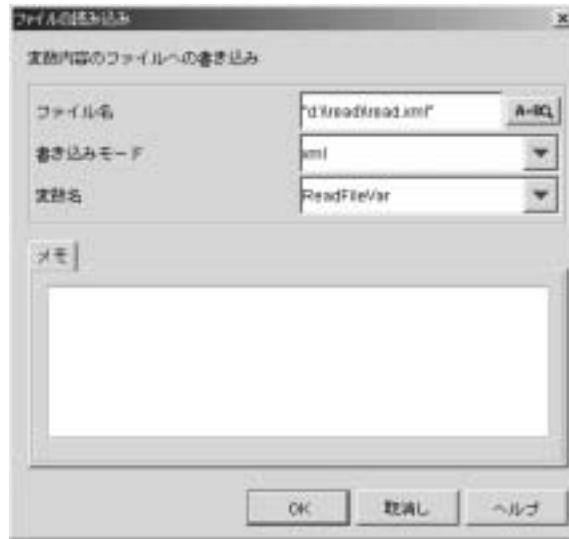
- DOS ("d:\\read\\read.xml" など)
- UNIX ("/home/workflow/input.txt" など)

式の詳細については、『*WebLogic Integration Studio ユーザーズガイド*』の「ワークフロー式の使用方法」を参照してください。

ファイルの読み込み

File Plug-in を使用してファイルを読み込み、ファイル内容をワークフロー変数に割り当てます。図 1-6 は、ファイルの読み込みアクションを定義するウィンドウを示しています。

図 1-6 [ファイルの読み込み] ウィンドウ



テンプレート定義

ファイルを読み込むには、以下のプロパティを設定します。

注意： ワークフローのタスクのプロパティを定義する前に、ワークフロー変数を定義する必要があります。ファイル内容はワークフロー変数にバインドされます。

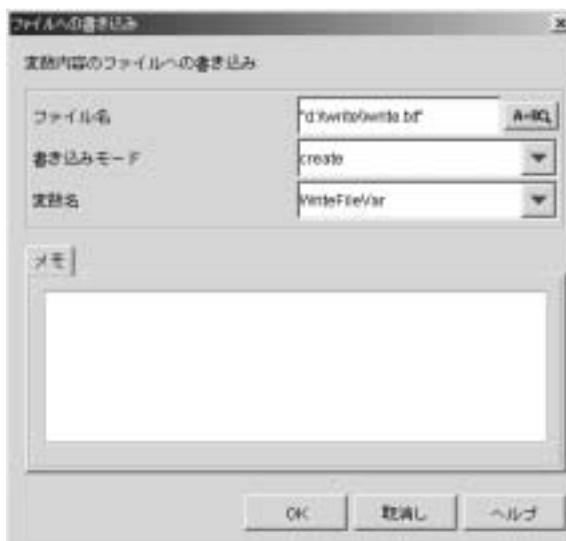
- ノード：ワークフロー アクションを使用できる任意のノード（タスク ノードなど）。
- [アクションを追加] ウィンドウで [FilePlugIn アクション]、[ファイル内容の変数への読み込み] の順に選択。
- [ファイルの読み込み] ウィンドウ（図 1-6）：
 - [ファイル名]：エントリは、ワークフロー式言語で定式化し、絶対パス名を使用すること（1-7 ページの「ワークフローの式」を参照）。
 - [ファイルタイプ]：XML、テキスト、またはバイナリ。
 - [変数名]：内容を格納する変数。

警告: 複数のワークフロー インスタンスから同じファイルを読み込むことができます。複数のスレッドから同じファイルを同時に変更することは避けてください。

ファイルへの書き込み

File Plug-in を使用して、指定したファイルに変数の内容を書き込むことができるタスクを定義します。このアクションを使用すると、ファイルの新規作成、変数内容の既存ファイルへの追加、または既存ファイルの上書きができます。図 1-7 は、ファイルへの書き込みアクションを定義するウィンドウを示しています。

図 1-7 [ファイルへの書き込み] ウィンドウ



ファイルに書き込むには、以下のプロパティを設定します。

注意: ワークフローのタスクのプロパティを定義する前に、ワークフロー変数を定義する必要があります。ファイル内容はワークフロー変数にバインドされます。

- ノード: ワークフロー アクションを使用できる任意のノード (タスク ノードなど)。

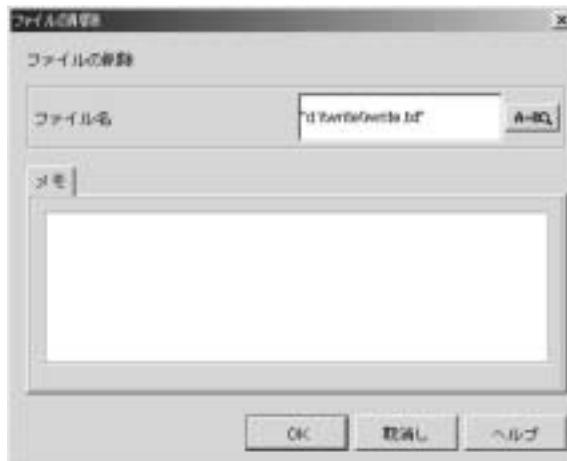
- [アクションを追加] ウィンドウで [FilePlugin アクション]、[変数内容のファイルへの書き込み] の順に選択。
- [ファイルへの書き込み] ウィンドウ (図 1-7):
 - [ファイル名]: エントリは、ワークフロー式言語で定義し、絶対パス名を使用すること (1-7 ページの「ワークフローの式」を参照)。使用できるファイルタイプは、テキスト、XML、およびバイナリです。
 - [書き込みモード]: 作成、追加、または上書き。
 - [変数名]: 内容を格納する変数。

警告: 複数のスレッドから同じファイルを更新することはできません。複数のスレッドから同じファイルを同時に変更することは避けてください。

ファイルの削除

File Plug-in を使用して、ファイルを削除できるワークフローを定義します。図 1-8 は、ファイルの削除アクションのウィンドウを示しています。

図 1-8 ファイルの削除



ファイルを削除するには、以下のプロパティを設定します。

- ノード : ワークフロー アクションを使用できる任意のノード (タスク ノード など)。
- [FilePlugin アクション]、[ファイルの削除] の順に選択。
- [ファイルの削除] ウィンドウ (図 1-8):
[ファイル名]: エントリは、ワークフロー式言語で定式化し、絶対パス名を使用すること (1-7 ページの「ワークフローの式」を参照)。

ファイル検出時にワークフローを再開

File Plug-in を使用してイベント ノードを定義すると、ワークフロー インスタンスがそのノードに到達したときに、ワークフローは指定されたファイルの存在を確認します。ファイルが存在する場合、ワークフローはファイルの内容を読み込んだ後、先に進みます。ファイルが存在しない場合、ワークフローはファイルが作成されるまで待機します。図 1-13 は、イベント ノードを定義するウィンドウを示しています。

注意: 1つのファイルで再開できるワークフロー インスタンスは1つだけです。

図 1-9 ファイルの再開に使用する [イベントのプロパティ] ウィンドウ



ファイルの再開では、以下のプロパティを設定します。

注意： ワークフローのタスクのプロパティを定義する前に、ワークフロー変数を定義する必要があります。ファイル内容はワークフロー変数にバインドされます。

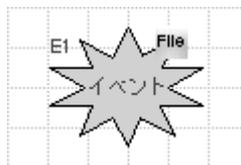
- ノード：[イベント]、[イベントのプロパティ]、[タイプ] ドロップダウン リストの [ファイルの再開] の順に選択。
- [イベントのプロパティ] ウィンドウ (図 1-9):
 - [ディレクトリ名]: エントリは、ワークフロー式言語で定式化し、絶対パス名を使用すること。
 - [ファイル名]: エントリは、ワークフロー式言語で定式化し、絶対パス名を使用すること。ワイルドカードは使用できません。

- [時間間隔]: エントリは、ワークフロー式言語で定式化すること。秒数は整数値にする必要があります。
- [ファイルタイプ]: テキスト、XML、またはバイナリ。
- [アーカイブ ディレクトリ]: エントリは、ワークフロー式言語で定式化し、絶対パス名を使用すること。

注意: ワークフロー式の詳細については、1-7 ページの「ワークフローの式」を参照してください。

プロパティを設定すると、図 1-10 に示すように、イベント ノードにアクションが設定されていることが表示されます。

図 1-10 ファイル アクションが設定されたイベント ノード



ファイル作成の検出時にワークフローを開始

File Plug-in を使用して、ファイルが作成されたときにワークフローを開始します。図 1-11 は、開始ノードを定義するウィンドウを示しています。

注意: 1つのファイルで開始できるワークフロー インスタンスは1つだけです。

図 1-11 ファイルの開始に使用する [開始のプロパティ] ウィンドウ



ファイルの作成を契機としてワークフローを開始するには、以下のプロパティを設定します。

注意： ワークフローのタスクのプロパティを定義する前に、ワークフロー変数を定義する必要があります。ファイル内容はワークフロー変数にバインドされます。

- ノード：[開始]、[イベント] ラジオ ボタン、イベントのドロップダウン リストの [ファイルの開始] の順に選択。

注意: [開始のプロパティ] ウィンドウで [イベント] ラジオ ボタンを選択すると、イベントのドロップダウン リストが表示されます。ドロップダウン リストから [ファイルの開始] を選択すると、[開始のプロパティ] ウィンドウの表示が変わり、ファイルの開始のプロパティが表示されます。

- [開始のプロパティ] ウィンドウ (図 1-11):
 - [ディレクトリ名]: 絶対パス名。たとえば、"d:\start\workflow.xml" とします。
 - [ファイル名マスク]: ファイルの名前。ワイルドカードは使用できません。
 - [時間間隔]: 秒単位のポーリング間隔を整数で設定します。
 - [ファイルタイプ]: テキスト、XML、またはバイナリ。
 - [変数名]: 内容を格納する変数。
 - [アーカイブ ディレクトリ]: 絶対パス名。たとえば、"d:\jan03\archive" とします。

プロパティを設定すると、図 1-12 に示すように、開始ノードにアクションが設定されていることが表示されます。

図 1-12 ファイル アクションが設定された開始ノード



FTP を使用したファイルの取得と転送

File Plug-in を使用して、ローカル システムとリモート システム間でファイルを移動します。

ファイルの *get* では、追加処理のためにファイルをリモート システムから取得します。追加処理には、ワークフローの開始、イベントのトリガ、情報のワークフローへの読み込みなどがあります。

ファイルの *put* では、アーカイブのために、またはリモート システム上のアプリケーションをトリガするために、ローカル システムからリモート システムへファイルを移動します。

図 1-13 は、get および put アクションを定義するウィンドウを示しています。

図 1-13 [ファイル転送 (FTP)] ウィンドウ



システム間でファイルを転送するには、以下のプロパティを設定します。

注意： 転送アクションを定義する前に、ファイルの読み込みアクションを定義する必要があります (1-8 ページの「ファイルの読み込み」を参照)。

ファイルの読み込みアクションの場合と同様に、ワークフロー変数を定義する必要があります。ファイル内容はワークフロー変数にバインドされます。

- ノード：ワークフローアクションを使用できる任意のノード（タスク ノードなど）。
- [アクションを追加] ウィンドウで [FilePlugIn アクション]、[ファイル転送 (FTP)] の順に選択。
- [ファイル転送] ウィンドウ (図 1-13):
 - [転送アクション]: put または get。

- [転送モード]: ASCII またはバイナリ。
- [FTP サーバ名]: エントリは、ワークフロー式言語で定式化し、絶対パス名を使用すること。
- [リモート ファイル名]: エントリは、ワークフロー式言語で定式化し、絶対パス名を使用すること。
- [ローカル ファイル名]: エントリは、ワークフロー式言語で定式化し、絶対パス名を使用すること。
- [ユーザ名]: FTP サーバへのログオンに必要な名前。
- [パスワード]: FTP サーバへのログオンに必要なパスワード。

警告: `get` では、同じファイル名がローカル システムにある場合、そのファイルを上書きします。`put` では、同じファイル名がリモート システムにある場合、そのファイルを上書きします。

注意: ワークフロー式の詳細については、1-7 ページの「ワークフローの式」を参照してください。

移行したドメインのための File Plug-in の コンフィグレーション

この節では、単一サーバドメインおよびクラスタドメインでのデータベーススキーマの更新方法および File Plug-in のコンフィグレーション方法について説明します。

BPM データベース テーブルの更新

File Plug-in では、FILEPOLL という新しいデータベース テーブルを使用しています。このテーブルを使用して BPM データベースを更新するには、次のスクリプトを実行します。

```
BEA_HOME\weblogic700\integration\dbscripts\<<database_type>\migrate\BPM_70-70SP2.sql
```

この `BEA_HOME` は、WebLogic Platform ホームです。

単一サーバドメインへの移行

単一サーバドメインで使用する File Plug-in をコンフィグレーションするには、ドメインの `config.xml` ファイルを次のように編集します。

1. `fileplugin-ejb.jar` を WebLogic Integration アプリケーションの 1 つのコネクトとしてデプロイするには、以下を追加します。

```
<EJBComponent Name="WLI-BPM File Plug-in"
  Targets="<Customer_Server_Name> URI="fileplugin-ejb.jar"/>
```

2. File Plug-in で使用する JMS キューを追加するには、以下を追加します。

```
<JMSQueue JNDIName="com.bea.wli.bpm.FilePluginQueue"
  Name="WLI_BPM_FP" Template="WLI_JMSTemplate"/>
```

クラスタドメインへの移行

クラスタドメインで使用する File Plug-in をコンフィグレーションするには、クラスタの `config.xml` ファイルを次のように編集します。

注意： この例は、1 つの管理サーバ (`myserver`) と 2 つの管理対象サーバ (`c1`、`c2`) があるクラスタシステム (`mycluster`) を示しています。

1. `fileplugin-ejb.jar` をクラスタサーバにデプロイするには、以下を追加します。

```
<EJBComponent Name="WLI BPM File Plug-in"
  Targets="mycluster"
  URI="fileplugin-ejb.jar"/>
```

2. File Plug-in で使用する JMS キューをコンフィグレーションするには、以下を追加します。

```
<JMSDistributedQueue Name="WLI_BPM_FP"
  JNDIName="com.bea.wli.bpm.FilePluginQueue"
  Targets="mycluster">
```

```
<JMSDistributedQueueMember Name="WLI_BPM_FP-c1"
  JMSQueue="WLI_BPM_FP-c1"
  Weight="1" />

<JMSDistributedQueueMember Name="WLI_BPM_FP-c2"
  JMSQueue="WLI_BPM_FP-c2"
  Weight="1" />

<JMSTemplate Name="WLI_BPM_FP" />

</JMSDistributedQueue>
```

3. クラスタ ノードの **JMS** サーバ c1 上で、以下を追加します。

```
<JMSQueue Name="WLI_BPM_FP-c1"
  JNDIName="com.bea.wli.bpm.FilePluginQueue-c1"
  StoreEnabled="true"
  Template="WLI_JMSTemplate-c1" />
```

4. クラスタ ノードの **JMS** サーバ c2 上で、以下を追加します。

```
<JMSQueue Name="WLI_BPM_FP-c2"
  JNDIName="com.bea.wli.bpm.FilePluginQueue-c2"
  StoreEnabled="true"
  Template="WLI_JMSTemplate-c2" />
```

2 File Plug-in の使用例

この章では、File Plug-in を使用して FTP によるリモート サーバへのファイル転送を行う例を示します。例の前半部分は、ワークフロー実行時にファイルを FTP で転送するように、WebLogic Integration Studio でワークフローを設定する方法を示しています。例の後半部分は、JSP Worklist を使用してワークフローを開始し、その結果としてテキスト ファイルを転送する方法を示しています。

注意： この例では、WebLogic Integration サンプルドメインを使用します。詳細については、『WebLogic Integration の起動、停止およびカスタマイズ』の「サンプルドメインのコンフィグレーションと起動」を参照してください。

ワークフローの設定

ファイルを FTP で転送するためのワークフローを Studio で設定するには、次の手順を実行します。

注意： WebLogic Integration Studio の使用方法の詳細については、『WebLogic Integration チュートリアル』(<http://edocs.beasys.co.jp/e-docs/wli/docs70/bpmtutor/index.htm>) を参照してください。

1. WebLogic Integration サーバを起動します。
 - Windows では、[スタート | プログラム | BEA WebLogic Platform 7.0 | WebLogic Integration 7.0 | Integration Examples | Start Server] を選択します。
 - UNIX では、
`BEA_HOME/weblogic700/samples/integration/config/samples/startWebLogic.sh` を実行します。
この `BEA_HOME` は、WebLogic Platform のホーム ディレクトリです。

2. サーバが実行中の状態になった後、**WebLogic Integration Studio** を起動します。

- Windows では、[スタート | プログラム | BEA WebLogic Platform 7.0 | WebLogic Integration 7.0 | Studio] を選択します。
- UNIX では、以下を実行します。

```
BEA_HOME/weblogic700/integration/bin/studio.sh
```

3. 次のように指定して **WebLogic Integration Studio** にログオンします (図 2-1 を参照)。

- [ユーザ名]: joe
- [パスワード]: password
- [サーバ URL]: t3://localhost:7001

図 2-1 WebLogic Integration Studio へのログオン



4. **WebLogic Integration Studio** が表示されたら、[オーガニゼーション] ドロップダウンリストから [CDEExpress] を選択します。

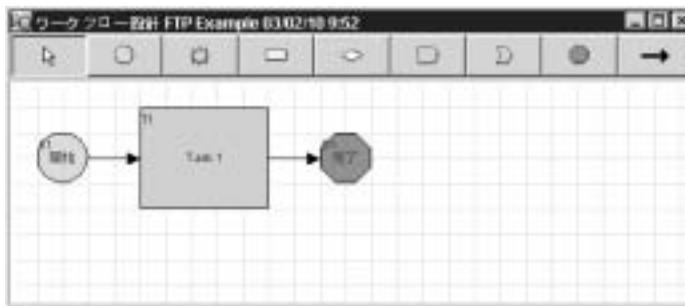
5. テンプレート ノードを右クリックし、[テンプレートの作成] を選択します。図 2-2 に示す [テンプレートのプロパティ] ウィンドウが表示されます。

図 2-2 [テンプレートのプロパティ] ウィンドウ



6. [一般] タブの [名前] フィールドに「FTP Example」と入力し、[OK] をクリックします。
7. オーガニゼーション ツリーでテンプレート ノードの FTP Example を右クリックし、[テンプレート定義を作成] を選択します。
8. [テンプレート定義 FTP Example] ウィンドウで、[OK] をクリックします。
図 2-3 に示すように、[ワークフロー設計] ウィンドウが右ペインに表示されます。

図 2-3 ワークフロー設計 – FTP Example



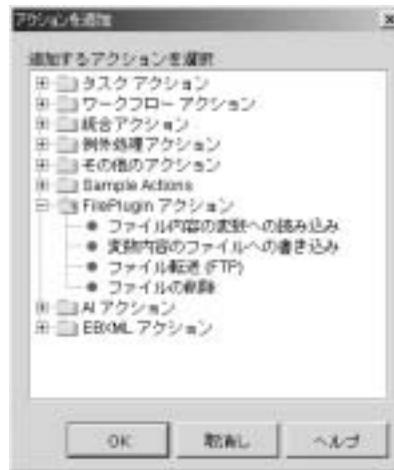
9. Task 1 を右クリックし、[プロパティ]を選択します。図 2-4 に示す [タスクのプロパティ] ウィンドウが表示されます。

図 2-4 [タスクのプロパティ] ウィンドウ



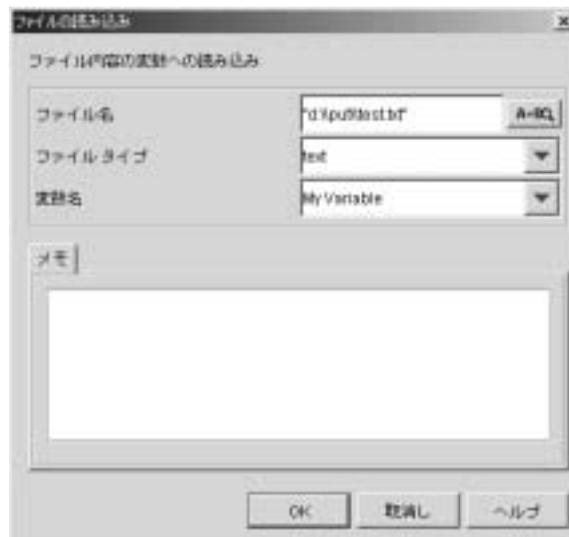
10. [アクティブ時] タブが選択されていることを確認します (実行時の状態)。
11. [追加] をクリックします。図 2-5 に示す [アクションを追加] ウィンドウが表示されます。

図 2-5 【アクションを追加】ウィンドウ



12. [FilePlugin アクション] フォルダを開き、[ファイル内容の変数への読み込み] を選択して、[OK] をクリックします。図 2-6 に示す [ファイルの読み込み] ウィンドウが表示されます。

図 2-6 【ファイルの読み込み】ウィンドウ



13. ファイルの読み込みアクションを次のように設定します。

- [ファイル名]: "d:\\put\\test.txt" (絶対パスを引用符で囲んで入力)
- [ファイルタイプ]: text
- [変数名]: My Variable

注意: Windows システムでは、バックスラッシュを 2 つ使用します。

14. [タスクのプロパティ] ウィンドウで、[実行時] タブを選択し、[追加] をクリックします。[アクションを追加] ダイアログ ボックスが表示されます。
15. [FilePlugin アクション] フォルダを開き、ファイル転送 (FTP) ノードを選択し、[OK] をクリックします。図 2-7 に示す [ファイル転送] ウィンドウが表示されます。

図 2-7 [ファイル転送] ウィンドウ



16. ファイル転送アクションを次のように設定します。
 - [転送アクション]: put
 - [転送モード]: binary
 - [FTP サーバ名]: 使用する FTP サーバの名前を引用符で囲んで入力

- [リモート ファイル名]: "c:\\receive\\gettest.txt" (引用符も入力)
 - [ローカル ファイル名]: "d:\\put\\test.txt" (引用符も入力)
 - [ユーザ名]: FTP サーバへのログオンに必要な名前。
 - [パスワード]: FTP サーバへのログオンに必要なパスワード。
17. [タスクのプロパティ] ウィンドウで、[ファイル転送 (FTP)] を選択して上矢印をクリックし、[OK] をクリックします。図 2-8 を参照してください。

図 2-8 タスクのプロパティ - ファイル転送タスクの移動



18. オーガニゼーション ツリーで変数ノードを右クリックし、[変数を作成] を選択します。図 2-9 に示す [変数プロパティ] ウィンドウが表示されます。

図 2-9 変数のプロパティ



19. 変数プロパティを次のように設定します。

- [名前]: My Variable
- [型]: String
- [パラメータ]: 入力

20. テンプレート定義フォルダを右クリックし、[プロパティ]を選択します。

21. [アクティブ]チェックボックスをチェックし、[OK]をクリックします。

22. テンプレートフォルダを右クリックし、[保存]を選択します。

注意: 未保存の変更がある場合は、フォルダの横にアスタリスクが表示されます。

23. ファイルシステムで、[ファイル転送]ウィンドウの[ローカルファイル名]に指定したテキストファイルを作成します。

ワークフローのアクティブ化

例のこの部分では、JSP Worklist によってワークフローが実行され、テキストファイルがリモートファイルシステムに転送されます。

JSP Worklist の詳細については『*WebLogic Integration JSP Worklist ユーザーズ ガイド*』を参照してください。

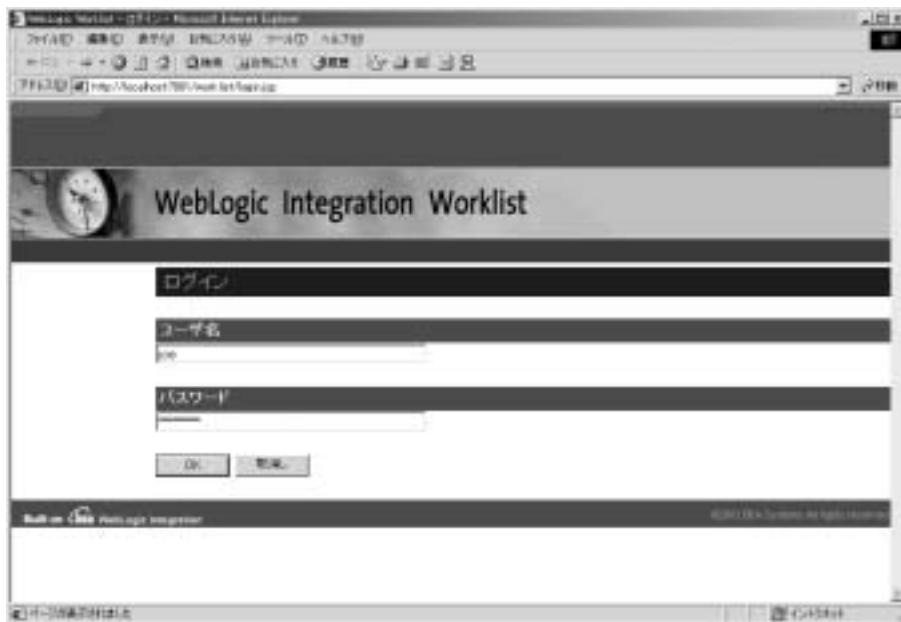
1. JSP Worklist は、以下のいずれかの方法で起動します。

- Windows システムでは、[スタート | プログラム | BEA WebLogic Platform 7.0 | WebLogic Integration 7.0 | Worklist] を選択します。
- UNIX システムでは、ブラウザを開いて次の URL を入力します。

`http://localhost:7001/worklist`

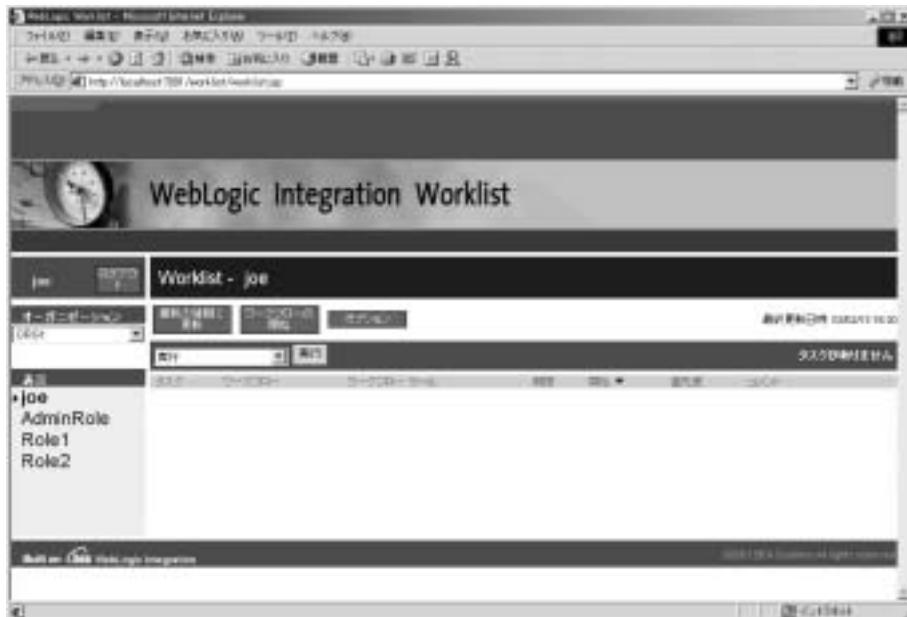
図 2-10 に示す [ログイン] ページが表示されます。

図 2-10 WebLogic Integration JSP Worklist へのログイン



2. [ユーザ名] フィールドに `joe` と入力します。
3. [パスワード] フィールドに `password` と入力します。
4. [OK] ボタンをクリックします。図 2-11 に示すように、JSP Worklist が表示されます。

図 2-11 JSP Worklist のメイン ページ



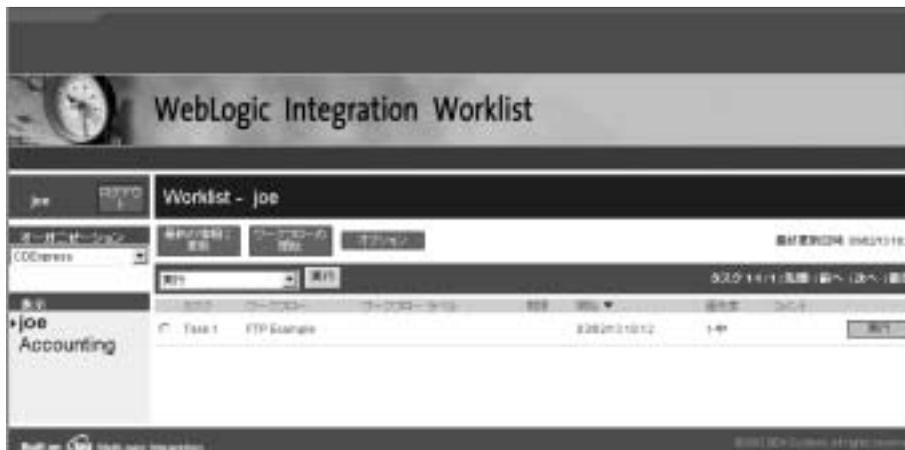
5. [オーガニゼーション] ドロップダウン リストから [CDEExpress] を選択します。
6. [ワークフローの開始] をクリックします。図 2-12 に示す [ワークフローの開始] ページが表示されます。

図 2-12 [ワークフローの開始] ページ



7. [開始するワークフローの選択] リストで [FTP Example] を選択し、[OK] をクリックします。図 2-13 に示すように、Worklist にタスクが表示されます。

図 2-13 新しいタスクが表示された Worklist



8. [実行] をクリックします。ファイルがターゲット ディレクトリに転送され、タスクがワークリストから消えます。
9. ターゲット ディレクトリに転送されたファイルがあることを確認します。

索引

D

Data Integration プラグイン 1-2

E

e-docs Web サイト v

F

File Plug-in の移行 1-18

File Plug-in の移行方法 1-18

File Plug-in の機能、リスト 1-1

File Plug-in の使用例 2-1

FTP 1-16

W

WebLogic Adapter for File 1-2

WebLogic Integration サンプルドメイン
2-1

あ

アクション、定義 1-1

[アクションを追加] ウィンドウ 1-6

い

一般情報 1-4

印刷、製品のマニュアル vi

か

概要 1-4

カスタマ サポート情報 vi

関連情報 vi

く

クラスタドメインへの移行 1-19

し

[式] ボタンについて 1-7

せ

前提条件 1-3

た

対象読者 v

単一サーバドメインへの移行 1-19

て

テクニカル サポート vii

データベースの移行 1-18

データベースへの移行 1-18

ひ

表記規則 vii

ふ

ファイル

get 1-16

put 1-16

上書き 1-10

書き込み 1-10

削除 1-11

追加 1-10

転送 2-1

ファイル転送の例 2-1

ファイルの `get` 1-16
ファイルの `put` 1-16
ファイルの上書き 1-10
ファイルの削除 1-11
ファイルへの書き込み 1-10
[プラグインのコンフィグレーション]
 ウィンドウ 1-3

へ

変数 1-5
変数内容のファイルへの追加 1-10

れ

例外ハンドラ 1-2

わ

ワークフロー
 アクティブ化 2-8
 開始 1-14
 再開 1-12
[ワークフロー設計] ウィンドウ 1-6
ワークフローのアクティブ化 2-8
ワークフローの式 1-7
ワークフローを開始 1-14
ワークフローを再開 1-12