



BEA WebLogic Portal

JavaServer Pages ガイド

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、複製、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 Portal、BEA WebLogic Server 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、関係各社がその権利を有します。

WebLogic Portal JavaServer Pages 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 9 月 | 7.0 サービス パック 1 |

目次

第1章 はじめに

| | |
|--------------------|-----|
| 作業別タグ一覧 | 1-1 |
| 名前順タグ一覧 | 1-2 |
| JSP テンプレート情報 | 1-3 |

第2章 ポータルおよびポートレット管理用 JSP タグ

| | |
|--|------|
| ポートレット タグ ライブラリ | 2-1 |
| <portlet:createWebflowURL> | 2-1 |
| 例 | 2-4 |
| <portlet:form> | 2-4 |
| <portlet:validatedForm> | 2-7 |
| 例 | 2-11 |
| <portlet:createPortletEditURL> | 2-12 |
| 例 | 2-14 |
| <portlet:createPortletUneditURL> | 2-15 |
| <portlet:createPortletMinimizeURL> | 2-17 |
| 例 | 2-19 |
| <portlet:createPortletUnminimizeURL> | 2-20 |
| 例 | 2-23 |
| <portlet:createPortletMaximizeURL> | 2-23 |
| 例 | 2-26 |
| <portlet:createPortletUnmaximizeURL> | 2-26 |
| 例 | 2-29 |
| <portlet:createPortletFloatURL> | 2-29 |
| 例 | 2-32 |
| <portlet:getException> | 2-32 |
| ポータル タグ ライブラリ | 2-33 |

| | |
|--|------|
| <portal:createWebflowURL> | 2-33 |
| <portal:form> | 2-36 |
| <portal:validatedForm> | 2-39 |
| <portal:createPortalPageChangeURL> | 2-42 |
| 例 | 2-44 |
| ユーティリティ タグ ライブラリ | 2-46 |
| <util:validURL> | 2-46 |
| <util:invalidURL> | 2-46 |
| 例 | 2-47 |

第3章 パーソナライゼーション JSP タグ

| | |
|---|------|
| 広告 | 3-2 |
| <ad:adTarget> | 3-2 |
| タグとコンテンツ管理サービス プロバイダ インタフェースとの マッピング | 3-2 |
| タグの属性 | 3-3 |
| 例 | 3-4 |
| コンテンツ管理 | 3-5 |
| <cm:getProperty> | 3-5 |
| タグとコンテンツ管理サービス プロバイダ インタフェースとの マッピング | 3-6 |
| タグの属性 | 3-6 |
| 例 | 3-8 |
| <cm:printDoc> | 3-9 |
| タグとコンテンツ管理サービス プロバイダ インタフェースとの マッピング | 3-9 |
| タグの属性 | 3-10 |
| 例 | 3-12 |
| <cm:printProperty> | 3-12 |
| タグとコンテンツ管理サービス プロバイダ インタフェースとの マッピング | 3-13 |
| タグの属性 | 3-13 |
| 例 | 3-15 |
| <cm:select> | 3-16 |
| タグとコンテンツ管理サービス プロバイダ インタフェースとの | |

| | |
|---|------|
| マッピング | 3-16 |
| タグの属性 | 3-17 |
| 例 | 3-19 |
| <cm:selectById> | 3-20 |
| タグとコンテンツ管理サービス プロバイダ インタフェースとの マッピング | 3-21 |
| タグの属性 | 3-21 |
| 例 | 3-24 |
| 国際化 | 3-25 |
| <i18n:localize> | 3-25 |
| タグの属性 | 3-26 |
| 例 1 | 3-28 |
| 例 2 | 3-29 |
| <i18n:getMessage> | 3-29 |
| タグの属性 | 3-30 |
| 例 1 | 3-31 |
| パーソナライゼーション タグ | 3-32 |
| pz タグと内部キャッシュ | 3-33 |
| <pz:contentQuery> | 3-33 |
| タグとコンテンツ管理サービス プロバイダ インタフェースとの マッピング | 3-34 |
| タグの属性 | 3-34 |
| 例 | 3-37 |
| <pz:contentSelector> | 3-38 |
| タグとコンテンツ管理サービス プロバイダ インタフェースとの マッピング | 3-39 |
| タグの属性 | 3-40 |
| contentHome の値を指定する | 3-43 |
| 例 | 3-43 |
| <pz:div> | 3-44 |
| タグの属性 | 3-44 |
| 例 | 3-45 |
| プレースホルダ | 3-46 |
| <ph:placeholder> | 3-46 |
| タグとコンテンツ管理サービス プロバイダ インタフェースとの マッピング | 3-47 |

| | |
|--|------|
| タグの属性 | 3-47 |
| 例 | 3-48 |
| プロパティ セット | 3-49 |
| <ps:getPropertyNames> | 3-49 |
| タグの属性 | 3-50 |
| 例 | 3-50 |
| <ps:getPropertySetNames> | 3-51 |
| タグの属性 | 3-51 |
| 例 | 3-52 |
| <ps:getRestrictedPropertyValues> | 3-53 |
| タグの属性 | 3-53 |
| 例 | 3-54 |
| ユーザ管理 : | |
| プロファイル管理タグ | 3-55 |
| <um:getProfile> | 3-55 |
| タグの属性 | 3-56 |
| 例 1 | 3-57 |
| 例 2 | 3-58 |
| <um:getProperty> | 3-58 |
| タグの属性 | 3-59 |
| 例 1 | 3-60 |
| <um:getPropertyAsString> | 3-60 |
| タグの属性 | 3-61 |
| 例 | 3-61 |
| <um:removeProperty> | 3-62 |
| タグの属性 | 3-62 |
| 例 | 3-63 |
| <um:setProperty> | 3-63 |
| タグの属性 | 3-64 |
| 例 | 3-64 |
| ユーザ管理 : | |
| グループおよびユーザ管理タグ | 3-65 |
| <um:addGroupToGroup> | 3-65 |
| タグの属性 | 3-66 |
| 例 | 3-67 |

| | |
|---------------------------------|------|
| <um:addUserToGroup> | 3-67 |
| タグの属性 | 3-68 |
| 例 | 3-68 |
| <um:createGroup> | 3-69 |
| タグの属性 | 3-70 |
| 例 | 3-70 |
| <um:createUser> | 3-71 |
| タグの属性 | 3-72 |
| 例 | 3-73 |
| <um:getChildGroupNames> | 3-73 |
| タグの属性 | 3-73 |
| 例 | 3-74 |
| <um:getGroupNamesForUser> | 3-74 |
| タグの属性 | 3-75 |
| 例 | 3-75 |
| <um:getParentGroupName> | 3-76 |
| タグの属性 | 3-76 |
| 例 | 3-77 |
| <um:getTopLevelGroups> | 3-77 |
| タグの属性 | 3-77 |
| 例 | 3-78 |
| <um:getUsernames> | 3-78 |
| タグの属性 | 3-79 |
| 例 | 3-80 |
| <um:getUsernamesForGroup> | 3-81 |
| タグの属性 | 3-81 |
| 例 | 3-83 |
| <um:removeGroup> | 3-83 |
| タグの属性 | 3-84 |
| 例 | 3-84 |
| <um:removeGroupFromGroup> | 3-85 |
| タグの属性 | 3-85 |
| 例 | 3-86 |
| <um:removeUser> | 3-86 |
| タグの属性 | 3-87 |

| | |
|--------------------------------|-------|
| 例 | 3-88 |
| <um.removeUserFromGroup> | 3-88 |
| タグの属性 | 3-89 |
| 例 | 3-90 |
| ユーザ管理 : セキュリティ タグ | 3-90 |
| <um.login> | 3-91 |
| タグの属性 | 3-91 |
| <um.logout> | 3-92 |
| タグの属性 | 3-93 |
| <um.setPassword> | 3-93 |
| タグの属性 | 3-94 |
| パーソナライゼーション ユーティリティ | 3-95 |
| <es.convertSpecialChars> | 3-95 |
| タグの属性 | 3-95 |
| 例 | 3-96 |
| <es.counter> | 3-96 |
| タグの属性 | 3-96 |
| 例 | 3-97 |
| <es.date> | 3-97 |
| タグの属性 | 3-97 |
| 例 | 3-98 |
| <es.forEachInArray> | 3-98 |
| タグの属性 | 3-99 |
| 例 | 3-99 |
| <es.isNull> | 3-99 |
| タグの属性 | 3-100 |
| 例 | 3-100 |
| <es.notNull> | 3-100 |
| タグの属性 | 3-101 |
| 例 | 3-101 |
| <es.transposeArray> | 3-102 |
| タグの属性 | 3-102 |
| 例 | 3-102 |
| <es.uriContent> | 3-103 |
| タグの属性 | 3-103 |

| | |
|------------------------|-------|
| 例 | 3-103 |
| WebLogic ユーティリティ | 3-105 |
| <wl:cache> | 3-105 |
| タグの属性 | 3-105 |
| <wl:process> | 3-106 |
| タグの属性 | 3-107 |
| 例 | 3-107 |
| <wl:repeat> | 3-108 |
| タグの属性 | 3-108 |

第4章 ナビゲーション (Webflow) 用 JSP タグ

| | |
|-----------------------------------|------|
| URL 作成タグ | 4-1 |
| <webflow:createWebflowURL> | 4-2 |
| 例 | 4-5 |
| <webflow:createResourceURL> | 4-6 |
| 例 | 4-7 |
| フォーム タグ | 4-7 |
| <webflow:form> | 4-7 |
| 例 | 4-10 |
| 検証機能付きフォーム タグ | 4-11 |
| <webflow:validatedForm> | 4-11 |
| <webflow:text> | 4-15 |
| <webflow:password> | 4-16 |
| <webflow:radio> | 4-17 |
| <webflow:checkbox> | 4-17 |
| <webflow:textarea> | 4-18 |
| <webflow:select> | 4-19 |
| <webflow:option> | 4-20 |
| 例 | 4-22 |
| Pipeline セッション タグ | 4-23 |
| <webflow:setProperty> | 4-23 |
| 例 | 4-24 |
| <webflow:getProperty> | 4-25 |
| 例 1 | 4-26 |

| | |
|-----------------------------------|------|
| 例 2 | 4-26 |
| <webflow:setValidatedValue> | 4-26 |
| 例 | 4-27 |
| <webflow:getValidatedValue> | 4-28 |
| 例 1 | 4-29 |
| 例 2 | 4-30 |
| <webflow:getException> | 4-30 |
| 例 | 4-31 |

第 5 章 カタログ開発用 JSP タグ

| | |
|-------------------------------------|------|
| カタログ用 JSP タグ | 5-1 |
| <catalog:getProperty> | 5-2 |
| 例 1 | 5-4 |
| 例 2 | 5-4 |
| <catalog:iterateViewIterator> | 5-5 |
| 例 1 | 5-6 |
| 例 2 | 5-6 |
| <catalog:iterateThroughView> | 5-7 |
| 例 1 | 5-8 |
| 例 2 | 5-8 |
| e ビジネス用 JSP タグ | 5-9 |
| <eb:smnav> | 5-9 |
| 例 | 5-10 |

第 6 章 イベントおよび行動追跡用 JSP タグ

| | |
|---------------------------------|-----|
| コンテンツ タグ | 6-2 |
| <tr:clickContentEvent> | 6-3 |
| 例 | 6-3 |
| <tr:displayContentEvent> | 6-5 |
| 例 | 6-5 |
| プロダクト タグ | 6-6 |
| <trp:clickProductEvent> | 6-6 |
| 例 | 6-7 |
| <trp:displayProductEvent> | 6-8 |

| | |
|---------|------|
| 例 | 6-10 |
|---------|------|

付録 A JSP テンプレート

| | |
|---|-----|
| テンプレートの場所 | A-1 |
| Commerce テンプレート | A-2 |
| Campaign テンプレート | A-2 |
| Catalog テンプレート | A-3 |
| Catalog および Campaign の include テンプレート | A-3 |
| Order テンプレート | A-5 |
| Registration テンプレート | A-7 |
| User テンプレート | A-7 |

付録 B JSP パフォーマンスのチューニング

| | |
|--|-----|
| ステップ 1: JSP のプリコンパイル | B-1 |
| ステップ 2: Web アプリケーションに対する Java コンパイラの指定 | B-2 |
| ステップ 3: JSP およびサーブレットの変更チェック間隔の調整 | B-3 |
| ページ チェック間隔のプロパティ | B-4 |
| 間隔の調節 | B-4 |

付録 C JSP タグ リファレンス (名前順)



まえがき

『JavaServer Pages ガイド』へようこそ。このマニュアルの他に、以下のリソースも活用されることをお勧めします。

オンライン マニュアルの参照 BEA 製品マニュアルは、BEA 社の Web サイトで公開しています。BEA Home ページで [製品のドキュメント] リンクをクリックするか、「e-docs」製品ドキュメント ページ (<http://edocs.beasys.co.jp/e-docs/>) に直接アクセスしてください。

マニュアルについてのフィードバック BEA WebLogic Portal マニュアルについてのフィードバックをお寄せください。ご質問やコメントがあれば、電子メールで docsupport-jp@bea.com までお送りください。なお、お送りいただく電子メールには、WebLogic Portal 7.0 のマニュアルをお使いであることを明記してください。

BEA WebSUPPORT への連絡 このバージョンの BEA WebLogic Portal について質問がある場合、または BEA WebLogic Portal のインストールや実行に問題がある場合には、BEA WebSUPPORT (<http://support.bea.com/welcome.jsp>) を通じて BEA カスタマ サポートにご連絡ください。製品パッケージに同梱のカスタマ サポート カードに記載されている連絡先にお問い合わせいただいても結構です。



第1章 はじめに

『JavaServer Pages ガイド』へようこそ。このマニュアルでは、WebLogic Portal で利用できるポータル、その他の JSP 機能の開発に使用する JSP タグについて詳しく解説しています。『開発者ガイド』の補足としてご利用ください。取り上げる内容は以下のとおりです。

- [作業別タグ一覧](#)
- [名前順タグ一覧](#)
- [JSP テンプレート情報](#)

作業別タグ一覧

ポータルおよびポータルレットの開発中に実行する必要があるさまざまな作業を中心に解説しています。したがって、1つの章で複数のタグライブラリのタグについて説明することもあります。それぞれの章および作業で説明しているタグライブラリは、[表 1-1](#) のとおりです。

表 1-1 JSP タグライブラリ

| タグライブラリ | 使用する作業 | 章 |
|-------------|---------------------|------------------------------------|
| ad.tld | 広告ブレースホルダ | パーソナライゼーション JSP タグ |
| catalog.tld | カタログ サービス管理 | カタログ開発用 JSP タグ |
| cm.tld | コンテンツ管理 | パーソナライゼーション JSP タグ |
| eb.tld | E-Business サービス管理 | カタログ開発用 JSP タグ |
| es.tld | パーソナライゼーション ユーティリティ | パーソナライゼーション JSP タグ |
| il8n.tld | 国際化管理 | パーソナライゼーション JSP タグ |

表 1-1 JSP タグ ライブラリ

| タグ ライブラリ | 使用する作業 | 章 |
|---------------------|------------------|--|
| ph.tld | プレースホルダ管理 | パーソナライゼーション JSP タグ |
| portal.tld | ポータル管理 | ポータルおよびポートレット管理用 JSP タグ |
| portlet.tld | ポートレット管理 | ポータルおよびポートレット管理用 JSP タグ |
| productTracking.tld | イベントおよび行動追跡管理 | イベントおよび行動追跡用 JSP タグ |
| ps.tld | プロパティ セット管理 | パーソナライゼーション JSP タグ |
| pz.tld | パーソナライゼーション管理 | パーソナライゼーション JSP タグ |
| tracking.tld | イベントおよび行動追跡管理 | イベントおよび行動追跡用 JSP タグ |
| um.tld | ユーザ管理 | パーソナライゼーション JSP タグ |
| util.tld | ポータルおよびポートレット管理 | ポータルおよびポートレット管理用 JSP タグ |
| webflow.tld | ナビゲーション管理 | ナビゲーション (Webflow) 用 JSP タグ |
| wl | WebLogic ユーティリティ | パーソナライゼーション JSP タグ |

名前順タグ一覧

作業中心のタグの解説に加えて、作業とは無関係にすべてのタグをアルファベット順に並べた一覧が [JSP タグ リファレンス \(名前順\)](#) にあります。この一覧で、各タグの簡単な内容とそのタグ ライブラリについて知ることができます。このマニュアルを Web ブラウザや PDF ビューアでオンライン表示している場合は、タグ名をクリックすると、作業別の章にある詳細な解説を見ることができます。

JSP テンプレート情報

このマニュアルの 2 つの付録、[JSP テンプレート](#)および[JSP パフォーマンスのチューニング](#)は、BEA が提供している JSP に適用されます。

- [JSP テンプレート](#)は、WebLogic Portal のインストールで利用できるすべての JSP テンプレートの一覧で、各テンプレートの内容について知ることができます。

[JSP パフォーマンスのチューニング](#)では、コンパイルおよび更新フェーズを調整して能力を最大化することにより JSP パフォーマンスを改善する方法について説明しています。

第2章 ポータルおよびポートレット管理用 JSP タグ

WebLogic Portal には、ポータルやポートレットの開発を支援するための一連の JSP タグが用意されています。これらの定義済みのタグを使えば、JSP ページに大量の Java コードを記述しなくてもよくなります。この章では、各タグセットを Web ページにインポートする方法と、各タグの目的について説明します。

以降の表の「必須」列では、属性が必須と省略可能のどちらであるかを「はい」または「いいえ」で示します。「R/C」列では、「C」は属性がコンパイル時表現であることを、「R」は属性が要求時表現とコンパイル時表現のいずれかであることを示します。

このトピックは、以下の節で構成されています。

- [ポートレットタグライブラリ](#)
- [ポータルタグライブラリ](#)
- [ユーティリティタグライブラリ](#)

ポートレットタグライブラリ

以下の節では、ポートレットタグとそれらの属性について説明します。

<portlet:createWebflowURL>

| | |
|---------|-------------|
| タグライブラリ | portlet.tld |
|---------|-------------|

| | |
|-----------------|---|
| インポート文 | <%@ taglib uri="portlet.tld" prefix="portlet" %> |
| 実装されるクラス | CreatePortletURLTag CreateWebflowURLTagExtraInfo |

<portlet:createWebflowURL> タグはポートレットで使用され、Webflow URL を JSP 内に動的に生成します。Webflow URL には、プロトコル、ドメイン名、ポート、Web アプリケーション URI、WebflowServlet URI、およびクエリ文字列を入れることができます。

表 2-1 は、<portlet:createWebflowURL> タグの属性についての説明です。

表 2-1 <portlet:createWebflowURL> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|---|-----|
| pageName | いいえ | String | イベントの発生元のページ。 | R |
| portletName | いいえ | String | 当該ポートレットの ID。これは通常、自動的に生成されるが、変更は可能。 | R |
| origin | いいえ | String | イベントの出発点となるノード（起点）で、<ノード名>.<ノードタイプ>の形式に従う。ページ名と同じ場合も異なる場合もある。省略された場合、JSP ページ名が使われる。 | R |
| event | はい | String | Webflow は、これを起点と組み合わせて使って、指定されたネームスペース XML ファイル内の遷移先を決定する。 | R |
| domainName | いいえ | String | URL のドメイン名や IP アドレスを変更するのに使う。Webflow が別のマシン上のプロキシサーバを経由する場合に、この属性を使用できる。 | R |

表 2-1 <portlet:createWebflowURL> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|--|-----|
| pathprefix | いいえ | String | <p>パス情報の接頭辞として使われる。この文字列は、Web アプリケーション URI の前に置かれる。プロキシ サーバが同一のマシン上にある場合に使われる。</p> <p>注意: プロキシ サーバ側では、このパス接頭辞を取り除いてから Webflow にリクエストを転送する必要がある。</p> | R |
| pathsuffix | いいえ | String | <p>パス情報の接尾辞として使われる。WebflowServlet URI に続けて、追加のパス情報が置かれる。これにより、この情報は、request.getPathInfo() メソッドから取得可能になる。</p> | R |
| extraParams | いいえ | String | <p>リクエストパラメータを名前と値のペアとして追加指定するのに使われる。</p> | R |
| doRedirect | いいえ | String | <p>プレゼンテーション ノードへの転送ではなくリダイレクトを行わせる。有効な値は true と false。デフォルトは false (リダイレクトしない)。</p> | C |
| httpsInd | いいえ | String | <p>Webflow 側でプロトコルを推定するか、あるいは HTTPS または HTTP を使うように指定する。このため、有効な値は calculate、http、および https。デフォルト値は http。</p> <p>Calculate を指定すると、起点 / イベントの分岐連鎖のリストに含まれているいずれかのノードが、アプリケーションの META-INF/web.xml ファイル内の HTTPS_URL_PATTERNS コンテキスト パラメータで指定されているパターンと一致する場合には、HTTPS が選択される。Calculate のほうがより動的で処理コストがかかるが、プロトコルを強制する必要がある場合には、それをここで指定することができる。</p> | C |

表 2-1 <portlet:createWebflowURL> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------|-----|--------|--|-----|
| encode | いいえ | String | Webflow 側で URL をコード化するように指定する。セッションの状態を保持したいが、ブラウザがクッキーを受け付けない場合には、URL をコード化の必要がある。有効な値は true と false。デフォルト値は true。ただし、ブラウザがクッキーを受け付けない場合にしか、URL をコード化する必要はない。 | C |

例

コードリスト 2-1 は、<portlet:createWebflowURL> JSP タグの使用法の例です。

コードリスト 2-1 <portlet:createWebflow> タグの使用例

```
<%@ taglib uri="portlet.tld" prefix="portlet" %>...  
  
<center>  
<font size="6" color="green">Portlet 2 - Page 1</font><BR>  
<p>  
<p>  
Portlet Webflow Test:  
<p>  
<a href="<portlet:createWebflowURL event="switch"/>"> 次のページ </a>  
<p>  
</center>
```

<portlet:form>

| | |
|---------|--|
| タグライブラリ | portlet.tld |
| インポート文 | <%@ taglib uri="portlet.tld" prefix="portlet" %> |

実装されるクラス PortletFormTag

<portlet:form> タグは JSP で使用され、HTML フォーム タグを動的に生成します。このタグは <portlet:validatedForm> タグほど複雑ではなく、もっと単純です。詳細については、<portlet:validatedForm> タグを参照してください。

表 2-2 は、<portlet:form> タグの属性についての説明です。

表 2-2 <portlet:form> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|---|-----|
| pageName | いいえ | String | イベントの発生元のページ。 | R |
| portletName | いいえ | String | 当該ポートレットの ID。これは通常、自動的に生成されるが、変更は可能。 | R |
| origin | いいえ | String | イベントの出発点となるノード（起点）で、<ノード名>.<ノードタイプ>の形式に従う。ページ名と同じ場合も異なる場合もある。省略された場合、JSP ページ名が使われる。 | R |
| event | はい | String | Webflow は、これを起点と組み合わせて使って、指定されたネームスペース XML ファイル内の遷移先を決定する。 | R |
| domainName | いいえ | String | URL のドメイン名や IP アドレスを変更するのに使う。Webflow が別のマシン上のプロキシサーバを経由する場合に、この属性を使用できる。 | R |
| pathprefix | いいえ | String | パス情報の接頭辞として使われる。この文字列は、Web アプリケーション URI の前に置かれる。プロキシサーバが同一のマシン上にある場合に使われる。 注意： プロキシサーバ側では、このパス接頭辞を取り除いてから Webflow にリクエストを転送する必要がある。 | R |

表 2-2 <portlet:form> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|--|-----|
| pathsuffix | いいえ | String | パス情報の接尾辞として使われる。WebflowServlet URI に続けて、追加のパス情報が置かれる。これにより、この情報は、request.getPathInfo() メソッドから取得可能になる。 | R |
| extraParams | いいえ | String | リクエスト パラメータを名前と値のペアとして追加指定するのに使われる。 | R |
| doRedirect | いいえ | String | WebflowServlet にプレゼンテーションノードへの転送ではなくリダイレクトを行わせる。有効な値は true と false。デフォルトは false (リダイレクトしない)。 | C |
| httpsInd | いいえ | String | Webflow 側でプロトコルを推定するか、あるいは HTTPS または HTTP を使うように指定する。このため、有効な値は calculate、http、および https。デフォルト値は http。 Calculate を指定すると、起点 / イベントの分岐連鎖のリストに含まれているいずれかのノードが、アプリケーションの META-INF/web.xml ファイル内の HTTPS_URL_PATTERNS コンテキストパラメータで指定されているパターンと一致する場合には、HTTPS が選択される。Calculate のほうがより動的で処理コストがかかるが、プロトコルを強制する必要がある場合には、それをここで指定することができる。 | C |
| encode | いいえ | String | Webflow 側で URL をコード化するように指定する。セッションの状態を保持したいが、ブラウザがクッキーを受け付けない場合には、URL をコード化する必要がある。有効な値は true と false。デフォルト値は true。ただし、ブラウザがクッキーを受け付けない場合にしか、URL をコード化する必要はない。 | C |

表 2-2 <portlet:form> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------|-----|--------|--|-----|
| hide | いいえ | String | false に設定した場合、ネームスペース、起点、およびイベントは、隠しフォームフィールドとしてではなく、コマンドライン上に表示される。 有効な値は true と false。デフォルト値は true。 | C |
| method | いいえ | String | フォーム用に使われるメソッド。有効な値は get と post。デフォルト値は post。 | C |
| name | いいえ | String | フォームの名前。 | C |

<portlet:validatedForm>

| | |
|----------|---|
| タグ ライブラリ | portlet.tld |
| インポート文 | <%@ taglib uri="portlet.tld" prefix="portlet" %> |
| 実装されるクラス | PortletValidatedFormTag WebflowValidatedFormTagExtraInfo |

<portlet:validatedForm> タグは、有効性の検証が可能な HTML フォームを動的に生成するのに使用されます。Web サイトへの訪問者が無効な情報を入力すると、訪問者の入力はいったん保存され、適切なエラー メッセージを付けて再表示されます。

表 2-3 は、<portlet:ValidatedForm> タグの属性についての説明です。

表 2-3 <portlet:validatedForm> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|----------|-----|--------|--------------------------|-----|
| pageName | いいえ | String | 検証機能付きフォーム URL の生成元のページ。 | R |

表 2-3 <portlet:validatedForm> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|---|-----|
| portletName | いいえ | String | 当該ポートレットの ID。これは通常、自動的に生成されるが、変更は可能。 | R |
| origin | いいえ | String | イベントの出発点となるノード (起点) で、<ノード名>.<ノードタイプ> の形式に従う。ページ名と同じ場合も異なる場合もある。省略された場合、JSP ページ名が使われる。 | R |
| event | はい | String | Webflow は、これを起点と組み合わせて使って、指定されたネームスペース XML ファイル内の遷移先を決定する。 | R |
| domainName | いいえ | String | URL のドメイン名や IP アドレスを変更するのに使う。Webflow が別のマシン上のプロキシサーバを経由する場合に、この属性を使用できる。 | R |
| pathprefix | いいえ | String | パス情報の接頭辞として使われる。この文字列は、Web アプリケーション URI の前に置かれる。プロキシサーバが同一のマシン上にある場合に使われる。 注意: プロキシサーバ側では、このパス接頭辞を取り除いてから Webflow にリクエストを転送する必要がある。 | R |
| pathsuffix | いいえ | String | パス情報の接尾辞として使われる。WebflowServlet URI に続けて、追加のパス情報が置かれる。これにより、この情報は、request.getPathInfo() メソッドから取得可能になる。 | R |
| extraParams | いいえ | String | リクエストパラメータを名前と値のペアとして追加指定するのに使われる。 | R |

表 2-3 <portlet:validatedForm> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------------|-----|--------|--|-----|
| doRedirect | いいえ | String | WebflowServlet にプレゼンテーションノードへの転送ではなくリダイレクトを行わせる。有効な値は true と false。デフォルトは false (リダイレクトしない)。 | C |
| httpsInd | いいえ | String | Webflow 側でプロトコルを推定するか、あるいは HTTPS または HTTP を使うように指定する。このため、有効な値は calculate、http、および https。デフォルト値は http。 Calculate を指定すると、起点/イベントの分岐連鎖のリストに含まれているいずれかのノードが、アプリケーションの META-INF/web.xml ファイル内の HTTPS_URL_PATTERNS コンテキストパラメータで指定されているパターンと一致する場合には、HTTPS が選択される。Calculate のほうがより動的で処理コストがかかるが、プロトコルを強制する必要がある場合には、それをここで指定することができる。 | C |
| encode | いいえ | String | Webflow 側で URL をコード化するように指定する。セッションの状態を保持したいが、ブラウザがクッキーを受け付けない場合には、URL をコード化する必要がある。有効な値は true と false。デフォルト値は true。ただし、ブラウザがクッキーを受け付けない場合にしか、URL をコード化する必要はない。 | C |
| hide | いいえ | String | false に設定した場合、ネームスペース、起点、およびイベントは、隠しフォームフィールドとしてではなく、コマンドライン上に表示される。有効な値は true と false。デフォルト値は true。 | C |

表 2-3 <portlet:validatedForm> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------------------|-----|--------|---|-----|
| method | いいえ | String | フォーム用に使われるメソッド。有効な値は get と post。デフォルト値は post。 | C |
| name | いいえ | String | フォームの名前。 | C |
| validStyle | いいえ | String | 有効な HTML フィールドを書式化するのに使用されるスタイル。 | R |
| invalidStyle | いいえ | String | HTML フィールドが無効な場合に、そのフィールドまたはメッセージを書式化するのに使用されるスタイル。 | R |
| unspecifiedStyle | いいえ | String | HTML フィールドの検証前の初期スタイルを指定するのに使われる。 | R |
| styleId | いいえ | String | フィールドのステータスが valid、invalid、unspecified のいずれであるかに応じて、invalidStyle、unspecifiedStyle、validStyle のいずれか 1 つに設定されるスクリプト変数。HTML フォームの書式化をより細かく制御するのに使用できる。 | R |
| applyStyle | いいえ | String | フィールドのステータスに関連付けられているスタイルをメッセージまたはフィールドに適用するか、あるいはそのいずれにも適用しない。このため、有効な値は message、field、および none。デフォルト値は message。 | C |
| messageAlign | いいえ | String | エラーメッセージの表示位置をフィールドの上、フィールドの右、フィールドの下のいずれかに指定する。このため、有効な値は top、right、および bottom。デフォルト値は right。 | C |

例

以下のサンプル コードは、`<portlet:validatedForm>` JSP タグの使用法の例です。

コード リスト 2-2 `<portlet:validatedForm>` タグの使用例

```
<%@ taglib uri="portlet.tld" prefix="portlet" %>
<%@ taglib uri="webflow.tld" prefix="webflow" %>

<center>
<% String validStyle="color: black; font-family: Arial"; %>
<% String invalidStyle="color: darkred; font-style: Arial"; %>
<!-- InvalidFormException が送出されたらメッセージを表示する -->
<font size="3" color="darkred"><portlet:getException/></font>
<br>
<webflow:getProperty id="welcomeStr" property="WELCOME_STRING"
type="java.lang.String" scope="request" namespace="portlet3"/>
<% if (welcomeStr == null || welcomeStr.length() < 1)
{
%>
    <portlet:validatedForm event="button.go" applyStyle="message"
messageAlign="right"
    validStyle="<%=validStyle%>"
invalidStyle="<%=invalidStyle%>"
unspecifiedStyle="<%=validStyle%>" >
        <p>
            Welcome :
            <webflow:text name="welcome" value="Hello" size="15"
maxlength="30" /> <br>
            <input type="submit" name="Submit" />
        </portlet:validatedForm>
        <br>
    <%
    }
    else
    {
%>
        <font size="5" color="green"><%= welcomeStr %></font>
    <%
    }
%>
</center>
<p>
```

<portlet:createPortletEditURL>

| | |
|-----------------|--|
| タグライブラリ | portlet.tld |
| インポート文 | <%@ taglib uri="portlet.tld" prefix="portlet" %> |
| 実装されるクラス | CreateEditURLTag |

<portlet:createPortletEditURL> タグは、ポートレットの編集を表す Webflow URL を生成します。

表 2-4 は、<portlet:createPortletEditURL> タグの属性についての説明です。

表 2-4 <portlet:createPortletEditURL> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|---|-----|
| pageName | いいえ | String | 編集 URL の生成元のページ。 | R |
| portletName | いいえ | String | 当該ポートレットの ID。これは通常、自動的に生成されるが、変更は可能。 | R |
| origin | いいえ | String | イベントの出発点となるノード（起点）で、<ノード名>.<ノードタイプ>の形式に従う。ページ名と同じ場合も異なる場合もある。省略された場合、JSP ページ名が使われる。 | R |
| domainName | いいえ | String | URL のドメイン名や IP アドレスを変更するのに使う。Webflow が別のマシン上のプロキシサーバを経由する場合に、この属性を使用できる。 | R |

表 2-4 <portlet:createPortletEditURL> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|--|-----|
| pathprefix | いいえ | String | <p>パス情報の接頭辞として使われる。この文字列は、Web アプリケーション URI の前に置かれる。プロキシ サーバが同一のマシン上にある場合に使われる。</p> <p>注意: プロキシ サーバ側では、このパス接頭辞を取り除いてから Webflow にリクエストを転送する必要がある。</p> | R |
| pathsuffix | いいえ | String | <p>パス情報の接尾辞として使われる。WebflowServlet URI に続けて、追加のパス情報が置かれる。これにより、この情報は、request.getPathInfo() メソッドから取得可能になる。</p> | R |
| extraParams | いいえ | String | <p>リクエストパラメータを名前と値のペアとして追加指定するのに使われる。</p> | R |
| doRedirect | いいえ | String | <p>WebflowServlet にプレゼンテーションノードへの転送ではなくリダイレクトを行わせる。有効な値は true と false。デフォルトは false (リダイレクトしない)。</p> | R |

表 2-4 <portlet:createPortletEditURL> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|----------|-----|--------|---|-----|
| httpsInd | いいえ | String | <p>Webflow 側でプロトコルを推定するか、あるいは HTTPS または HTTP を使うように指定する。このため、有効な値は calculate、http、および https。デフォルト値は http。</p> <p>Calculate を指定すると、起点 / イベントの分岐連鎖のリストに含まれているいずれかのノードが、アプリケーションの META-INF/web.xml ファイル内の HTTPS_URL_PATTERNS コンテキストパラメータで指定されているパターンと一致する場合には、HTTPS が選択される。Calculate のほうがより動的で処理コストがかかるが、プロトコルを強制する必要がある場合には、それをここで指定することができる。</p> | R |
| encode | いいえ | String | <p>Webflow 側で URL をコード化するように指定する。セッションの状態を保持したいが、ブラウザがクッキーを受け付けない場合には、URL をコード化する必要がある。有効な値は true と false。デフォルト値は true。ただし、ブラウザがクッキーを受け付けない場合にしか、URL をコード化する必要はない。</p> | R |

例

コード リスト 2-3 は、<portlet:createPortletEditURL> JSP タグの使用方法の例です。

コード リスト 2-3 <portlet:createPortletEditURL> タグの使用例

```

    }
    if (PortletRenderHelper.isEditable(request))
    {
%>
        <td width="1%" valign="middle"><a

```

```
href="<portlet:createPortletEditURL httpsInd='http' />"
  ' />" hspace="1"
vspace="2" border="0" alt="Edit" align="absmiddle"></a></td>
<%
```

<portlet:createPortletUneditURL>

| | |
|-----------------|--|
| タグ ライブラリ | portlet.tld |
| インポート文 | <%@ taglib uri="portlet.tld" prefix="portlet" %> |
| 実装されるクラス | CreatePortletUneditURLTag |

<createPortletUneditURL> JSP タグは、編集 URL から別のページへの移動を表す Webflow URL を生成します。

表 2-5 は、<portlet:createPortletUneditURL> タグの属性についての説明です。

表 2-5 <portlet:createPortletUneditURL> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|---|-----|
| pageName | いいえ | String | 編集解除 URL の生成元のページ。 | R |
| portletName | いいえ | String | 当該ポートレットの ID。これは通常、自動的に生成されるが、変更は可能。 | R |
| origin | いいえ | String | イベントの出発点となるノード（起点）で、<ノード名>.<ノードタイプ>の形式に従う。ページ名と同じ場合も異なる場合もある。省略された場合、JSP ページ名が使われる。 | R |
| domainName | いいえ | String | URL のドメイン名や IP アドレスを変更するのに使う。Webflow が別のマシン上のプロキシサーバを経由する場合に、この属性を使用できる。 | R |

表 2-5 <portlet:createPortletUneditURL> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|--|-----|
| pathprefix | いいえ | String | <p>パス情報の接頭辞として使われる。この文字列は、Web アプリケーション URI の前に置かれる。プロキシ サーバが同一のマシン上にある場合に使われる。</p> <p>注意: プロキシ サーバ側では、このパス接頭辞を取り除いてから Webflow にリクエストを転送する必要がある。</p> | R |
| pathsuffix | いいえ | String | <p>パス情報の接尾辞として使われる。WebflowServlet URI に続けて、追加のパス情報が置かれる。これにより、この情報は、request.getPathInfo() メソッドから取得可能になる。</p> | R |
| extraParams | いいえ | String | <p>リクエスト パラメータを名前と値のペアとして追加指定するのに使われる。</p> | R |
| doRedirect | いいえ | String | <p>WebflowServlet にプレゼンテーションノードへの転送ではなくリダイレクトを行わせる。有効な値は true と false。デフォルトは false (リダイレクトしない)。</p> | R |

表 2-5 <portlet:createPortletUneditURL> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|----------|-----|--------|---|-----|
| httpsInd | いいえ | String | <p>Webflow 側でプロトコルを推定するか、あるいは HTTPS または HTTP を使うように指定する。このため、有効な値は calculate、http、および https。デフォルト値は http。</p> <p>Calculate を指定すると、起点 / イベントの分岐連鎖のリストに含まれているいずれかのノードが、アプリケーションの META-INF/web.xml ファイル内の HTTPS_URL_PATTERNS コンテキストパラメータで指定されているパターンと一致する場合には、HTTPS が選択される。Calculate のほうがより動的で処理コストがかかるが、プロトコルを強制する必要がある場合には、それをここで指定することができる。</p> | R |
| encode | いいえ | String | <p>Webflow 側で URL をコード化するように指定する。セッションの状態を保持したいが、ブラウザがクッキーを受け付けない場合には、URL をコード化する必要がある。有効な値は true と false。デフォルト値は true。ただし、ブラウザがクッキーを受け付けない場合にしか、URL をコード化する必要はない。</p> | R |

<portlet:createPortletMinimizeURL>

| | |
|----------|--|
| タグ ライブラリ | portlet.tld |
| インポート文 | <%@ taglib uri="portlet.tld" prefix="portlet" %> |
| 実装されるクラス | CreatePortletMinimizeURLTag |

第 2 章 ポータルおよびポートレット管理用 JSP タグ

`<portlet:createPortletMinimizeURL>` JSP タグはポートレットの最小化を表す Webflow URL を生成します。

表 2-6 は、`<portlet:createPortletMinimizeURL>` タグの属性についての説明です。

表 2-6 `<portlet:createPortletMinimizeURL>` タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|---|-----|
| pageName | いいえ | String | 最小化 URL の生成元のページ。 | R |
| portletName | いいえ | String | 当該ポートレットの ID。これは通常、自動的に生成されるが、変更は可能。 | R |
| origin | いいえ | String | イベントの出発点となるノード（起点）で、 <code><ノード名>.<ノードタイプ></code> の形式に従う。ページ名と同じ場合も異なる場合もある。省略された場合、JSP ページ名が使われる。 | R |
| domainName | いいえ | String | URL のドメイン名や IP アドレスを変更するのに使う。Webflow が別のマシン上のプロキシサーバを経由する場合に、この属性を使用できる。 | R |
| pathprefix | いいえ | String | パス情報の接頭辞として使われる。この文字列は、Web アプリケーション URI の前に置かれる。プロキシサーバが同一のマシン上にある場合に使われる。 注意： プロキシサーバ側では、このパス接頭辞を取り除いてから Webflow にリクエストを転送する必要がある。 | R |
| pathsuffix | いいえ | String | パス情報の接尾辞として使われる。WebflowServlet URI に続けて、追加のパス情報が置かれる。これにより、この情報は、 <code>request.getPathInfo()</code> メソッドから取得可能になる。 | R |
| extraParams | いいえ | String | リクエストパラメータを名前と値のペアとして追加指定するのに使われる。 | R |

表 2-6 <portlet:createPortletMinimizeURL> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------------|-----|--------|--|-----|
| doRedirect | いいえ | String | WebflowServlet にプレゼンテーションノードへの転送ではなくリダイレクトを行わせる。有効な値は true と false。デフォルトは false (リダイレクトしない)。 | R |
| httpsInd | いいえ | String | Webflow 側でプロトコルを推定するか、あるいは HTTPS または HTTP を使うように指定する。このため、有効な値は calculate、http、および https。デフォルト値は http。 Calculate を指定すると、起点 / イベントの分岐連鎖のリストに含まれているいずれかのノードが、アプリケーションの META-INF/web.xml ファイル内の HTTPS_URL_PATTERNS コンテキストパラメータで指定されているパターンと一致する場合には、HTTPS が選択される。Calculate のほうがより動的で処理コストがかかるが、プロトコルを強制する必要がある場合には、それをここで指定することができる。 | R |
| encode | いいえ | String | Webflow 側で URL をコード化するように指定する。セッションの状態を保持したいが、ブラウザがクッキーを受け付けない場合には、URL をコード化する必要がある。有効な値は true と false。デフォルト値は true。ただし、ブラウザがクッキーを受け付けない場合にしか、URL をコード化する必要はない。 | R |

例

コードリスト 2-4 は、<portlet:createPortletMinimizeURL> JSP タグの使用方法の例です。

コード リスト 2-4 <portlet:createPortletMinimizeURL> タグの使用例

```

    }
    if (PortletRenderHelper.isMinimizable(request))
    {
%>
        <td width="1%" valign="middle"><a
href="<portlet:createPortletMinimizeURL httpsInd='http' />">
        ' />" hspace="1"
vspace="2" border="0" alt="Minimize" align="absmiddle"></a></td>
<%

```

<portlet:createPortletUnminimizeURL>

| | |
|-----------------|--|
| タグライブラリ | portlet.tld |
| インポート文 | <%@ taglib uri="portlet.tld" prefix="portlet" %> |
| 実装されるクラス | CreateUnminimizeURLTag |

<portlet:createPortletUnminimizeURL> タグは、ポートレットの最小化解除を表す Webflow URL を生成します。

表 2-7 は、<portlet:createPortletUnminimizeURL> タグの属性についての説明です。

表 2-7 <portlet:createPortletUnminimizeURL> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|--------------------------------------|-----|
| pageName | いいえ | String | 最小化解除 URL の生成元のページ。 | R |
| portletName | いいえ | String | 当該ポートレットの ID。これは通常、自動的に生成されるが、変更は可能。 | R |

表 2-7 <portlet:createPortletUnminimizeURL> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|---|-----|
| origin | いいえ | String | イベントの出発点となるノード (起点) で、<ノード名>.<ノード タイプ>の形式に従う。ページ名と同じ場合も異なる場合もある。省略された場合、JSP ページ名が使われる。 | R |
| domainName | いいえ | String | URL のドメイン名や IP アドレスを変更するのに使う。Webflow が別のマシン上のプロキシサーバを経由する場合に、この属性を使用できる。 | R |
| pathprefix | いいえ | String | パス情報の接頭辞として使われる。この文字列は、Web アプリケーション URI の前に置かれる。プロキシサーバが同一のマシン上にある場合に使われる。 注意: プロキシサーバ側では、このパス接頭辞を取り除いてから Webflow にリクエストを転送する必要がある。 | R |
| pathsuffix | いいえ | String | パス情報の接尾辞として使われる。WebflowServlet URI に続けて、追加のパス情報が置かれる。これにより、この情報は、request.getPathInfo() メソッドから取得可能になる。 | R |
| extraParams | いいえ | String | リクエストパラメータを名前と値のペアとして追加指定するのに使われる。 | R |
| doRedirect | いいえ | String | WebflowServlet にプレゼンテーションノードへの転送ではなくリダイレクトを行わせる。有効な値は true と false。デフォルトは false (リダイレクトしない)。 | R |

表 2-7 <portlet:createPortletUnminimizeURL> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|----------|-----|--------|---|-----|
| httpsInd | いいえ | String | <p>Webflow 側でプロトコルを推定するか、あるいは HTTPS または HTTP を使うように指定する。このため、有効な値は calculate、http、および https。デフォルト値は http。</p> <p>Calculate を指定すると、起点 / イベントの分岐連鎖のリストに含まれているいずれかのノードが、アプリケーションの META-INF/web.xml ファイル内の HTTPS_URL_PATTERNS コンテキストパラメータで指定されているパターンと一致する場合には、HTTPS が選択される。Calculate のほうがより動的で処理コストがかかるが、プロトコルを強制する必要がある場合には、それをここで指定することができる。</p> | R |
| encode | いいえ | String | <p>Webflow 側で URL をコード化するように指定する。セッションの状態を保持したいが、ブラウザがクッキーを受け付けない場合には、URL をコード化する必要がある。有効な値は true と false。デフォルト値は true。ただし、ブラウザがクッキーを受け付けない場合にしか、URL をコード化する必要はない。</p> | R |

例

コードリスト 2-5 は、`<portlet:createPortletUnminimizeURL>` JSP タグの使用方法の例です。

コードリスト 2-5 `<portlet:createPortletUnminimizeURL>` タグの使用例

```
<%-- ポートレットの最小化を解除するリンクを生成する --%>
<td width="1%" valign="middle">
  <a href="<portlet:createPortletUnminimizeURL/>">
    '/>" hspace="1" vspace="2"
      border="0" alt="Restore" align="absmiddle"></a></td>
```

`<portlet:createPortletMaximizeURL>`

| | |
|----------|---|
| タグライブラリ | portlet.tld |
| インポート文 | <code><%@ taglib uri="portlet.tld" prefix="portlet" %></code> |
| 実装されるクラス | CreateMaximizeURLTag |

`<portlet:createPortletMaximizeURL>` JSP タグは、ポートレットの最大化を表す Webflow URL を生成します。

表 2-8 は、`<portlet:createPortletMaximizeURL>` タグの属性についての説明です。

表 2-8 `<portlet:createPortletMaximizeURL>` タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|--------------------------------------|-----|
| pageName | いいえ | String | 最大化 URL の生成元のページ。 | R |
| portletName | いいえ | String | 当該ポートレットの ID。これは通常、自動的に生成されるが、変更は可能。 | R |

表 2-8 <portlet:createPortletMaximizeURL> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|--|-----|
| origin | いいえ | String | イベントの出発点となるノード (起点) で、 < ノード名 > . < ノード タイプ > の形式に従う。ページ名と同じ場合も異なる場合もある。省略された場合、 JSP ページ名が使われる。 | R |
| domainName | いいえ | String | URL のドメイン名や IP アドレスを変更するのに使う。 Webflow が別のマシン上のプロキシサーバを経由する場合に、この属性を使用できる。 | R |
| pathprefix | いいえ | String | パス情報の接頭辞として使われる。この文字列は、 Web アプリケーション URI の前に置かれる。プロキシサーバが同一のマシン上にある場合に使われる。 注意: プロキシサーバ側では、このパス接頭辞を取り除いてから Webflow にリクエストを転送する必要がある。 | R |
| pathsuffix | いいえ | String | パス情報の接尾辞として使われる。 WebflowServlet URI に続けて、追加のパス情報が置かれる。これにより、この情報は、 request.getPathInfo() メソッドから取得可能になる。 | R |
| extraParams | いいえ | String | リクエストパラメータを名前と値のペアとして追加指定するのに使われる。 | R |
| doRedirect | いいえ | String | WebflowServlet にプレゼンテーションノードへの転送ではなくリダイレクトを行わせる。有効な値は true と false。デフォルトは false (リダイレクトしない)。 | R |

表 2-8 <portlet:createPortletMaximizeURL> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|----------|-----|--------|---|-----|
| httpsInd | いいえ | String | <p>Webflow 側でプロトコルを推定するか、あるいは HTTPS または HTTP を使うように指定する。このため、有効な値は calculate、http、および https。デフォルト値は http。</p> <p>Calculate を指定すると、起点 / イベントの分岐連鎖のリストに含まれているいずれかのノードが、アプリケーションの META-INF/web.xml ファイル内の HTTPS_URL_PATTERNS コンテキストパラメータで指定されているパターンと一致する場合には、HTTPS が選択される。Calculate のほうがより動的で処理コストがかかるが、プロトコルを強制する必要がある場合には、それをここで指定することができる。</p> | R |
| encode | いいえ | String | <p>Webflow 側で URL をコード化するように指定する。セッションの状態を保持したいが、ブラウザがクッキーを受け付けない場合には、URL をコード化する必要がある。有効な値は true と false。デフォルト値は true。ただし、ブラウザがクッキーを受け付けない場合にしか、URL をコード化する必要はない。</p> | R |

例

コード リスト 2-6 は、`<portlet:createPortletMaximizeURL>` JSP タグの使用
方法の例です。

コード リスト 2-6 `<portlet:createPortletMaximizeURL>` タグの使用例

```
<%
    }
    if (PortletRenderHelper.isMaximizable(request))
    {
%>

<td width="1%" valign="middle">
  <!-- ポートレットを最大化するリンクを生成する -->
  <a href="<portlet:createPortletMaximizeURL
httpsInd='http' />"><img src=
  "<webflow:createResourceURL
resource='<%=imagesPath+"portlet_max.
gif"%>' />" hspace="1" vspace="2" border="0"
alt="Maximize" align="absmiddle"></a></td>
<%
```

`<portlet:createPortletUnmaximizeURL>`

| | |
|----------|--|
| タグ ライブラリ | portlet.tld |
| インポート文 | <%@ taglib uri="portlet.tld" prefix="portlet" %> |
| 実装されるクラス | CreateUnmaximizeURLTag |

`<portlet:createPortletUnmaximizeURL>` JSP タグは、ポートレットの最大化
解除を表す Webflow URL を生成します。

表 2-8 は、`<portlet:createPortletUnmaximizeURL>` タグの属性についての説
明です。

表 2-9 <portlet:createPortletUnmaximizeURL> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|---|-----|
| pageName | いいえ | String | 最大化解除 URL の生成元のページ。 | R |
| portletName | いいえ | String | 当該ポートレットの ID。これは通常、自動的に生成されるが、変更は可能。 | R |
| origin | いいえ | String | イベントの出発点となるノード（起点）で、<ノード名>.<ノードタイプ>の形式に従う。ページ名と同じ場合も異なる場合もある。省略された場合、JSP ページ名が使われる。 | R |
| domainName | いいえ | String | URL のドメイン名や IP アドレスを変更するのに使う。Webflow が別のマシン上のプロキシサーバを経由する場合に、この属性を使用できる。 | R |
| pathprefix | いいえ | String | パス情報の接頭辞として使われる。この文字列は、Web アプリケーション URI の前に置かれる。プロキシサーバが同一のマシン上にある場合に使われる。 注意： プロキシサーバ側では、このパス接頭辞を取り除いてから Webflow にリクエストを転送する必要がある。 | R |
| pathsuffix | いいえ | String | パス情報の接尾辞として使われる。WebflowServlet URI に続けて、追加のパス情報が置かれる。これにより、この情報は、request.getPathInfo() メソッドから取得可能になる。 | R |
| extraParams | いいえ | String | リクエストパラメータを名前と値のペアとして追加指定するのに使われる。 | R |
| doRedirect | いいえ | String | WebflowServlet にプレゼンテーションノードへの転送ではなくリダイレクトを行わせる。有効な値は true と false。デフォルトは false（リダイレクトしない）。 | R |

表 2-9 <portlet:createPortletUnmaximizeURL> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|----------|-----|--------|---|-----|
| httpsInd | いいえ | String | <p>Webflow 側でプロトコルを推定するか、あるいは HTTPS または HTTP を使うように指定する。このため、有効な値は calculate、http、および https。デフォルト値は http。</p> <p>Calculate を指定すると、起点 / イベントの分岐連鎖のリストに含まれているいずれかのノードが、アプリケーションの META-INF/web.xml ファイル内の HTTPS_URL_PATTERNS コンテキストパラメータで指定されているパターンと一致する場合には、HTTPS が選択される。Calculate のほうがより動的で処理コストがかかるが、プロトコルを強制する必要がある場合には、それをここで指定することができる。</p> | R |
| encode | いいえ | String | <p>Webflow 側で URL をコード化するように指定する。セッションの状態を保持したいが、ブラウザがクッキーを受け付けない場合には、URL をコード化する必要がある。有効な値は true と false。デフォルト値は true。ただし、ブラウザがクッキーを受け付けない場合にしか、URL をコード化する必要はない。</p> | R |

例

以下のサンプル コードは、`<portal:createPortletUnmaximizeURL>` JSP タグの使用方法の例です。

コード リスト 2-7 `<portal:createPortletUnmaximizeURL>` タグの使用例

```
<!-- ポートレットの最大化を解除するリンクを生成する --%>
<td width="1%" valign="middle">
  <a href="<portal:createPortletUnmaximizeURL/>">
    '/>" hspace="1" vspace="2" border="0"
      alt="Unmaximize"></a></td>
```

`<portal:createPortletFloatURL>`

| | |
|----------|---|
| タグ ライブラリ | portlet.tld |
| インポート文 | <code><%@ taglib uri="portlet.tld" prefix="portlet" %></code> |
| 実装されるクラス | CreateFloatURLTag |

`<portal:createPortletFloatURL>` タグは、「移動可能な」ポートレット（すなわち、独立したウィンドウに表示されるポートレット）の作成を表す Webflow URL を生成します。

表 2-10 は、`<portal:createPortletFloatURL>` タグの属性についての説明です。

表 2-10 `<portal:createPortletFloatURL>` タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|----------|-----|--------|---------------------|-----|
| pageName | いいえ | String | 移動可能化 URL の生成元のページ。 | R |

表 2-10 <portlet:createPortletFloatURL> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|---|-----|
| portletName | いいえ | String | 当該ポートレットの ID。これは通常、自動的に生成されるが、変更は可能。 | R |
| origin | いいえ | String | イベントの出発点となるノード (起点) で、<ノード名>.<ノードタイプ>の形式に従う。ページ名と同じ場合も異なる場合もある。省略された場合、JSP ページ名が使われる。 | R |
| domainName | いいえ | String | URL のドメイン名や IP アドレスを変更するのに使う。Webflow が別のマシン上のプロキシサーバを経由する場合に、この属性を使用できる。 | R |
| pathprefix | いいえ | String | パス情報の接頭辞として使われる。この文字列は、Web アプリケーション URI の前に置かれる。プロキシサーバが同一のマシン上にある場合に使われる。 注意: プロキシサーバ側では、このパス接頭辞を取り除いてから Webflow にリクエストを転送する必要がある。 | R |
| pathsuffix | いいえ | String | パス情報の接尾辞として使われる。WebflowServlet URI に続けて、追加のパス情報が置かれる。これにより、この情報は、request.getPathInfo() メソッドから取得可能になる。 | R |
| extraParams | いいえ | String | リクエストパラメータを名前と値のペアとして追加指定するのに使われる。 | R |
| doRedirect | いいえ | String | WebflowServlet にプレゼンテーションノードへの転送ではなくリダイレクトを行わせる。有効な値は true と false。デフォルトは false (リダイレクトしない)。 | R |

表 2-10 <portlet:createPortletFloatURL> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|----------|-----|--------|---|-----|
| httpsInd | いいえ | String | <p>Webflow 側でプロトコルを推定するか、あるいは HTTPS または HTTP を使うように指定する。このため、有効な値は calculate、http、および https。デフォルト値は http。</p> <p>Calculate を指定すると、起点 / イベントの分岐連鎖のリストに含まれているいずれかのノードが、アプリケーションの META-INF/web.xml ファイル内の HTTPS_URL_PATTERNS コンテキストパラメータで指定されているパターンと一致する場合には、HTTPS が選択される。Calculate のほうがより動的で処理コストがかかるが、プロトコルを強制する必要がある場合には、それをここで指定することができる。</p> | R |
| encode | いいえ | String | <p>Webflow 側で URL をコード化するように指定する。セッションの状態を保持したいが、ブラウザがクッキーを受け付けない場合には、URL をコード化する必要がある。有効な値は true と false。デフォルト値は true。ただし、ブラウザがクッキーを受け付けない場合にしか、URL をコード化する必要はない。</p> | R |

例

コードリスト 2-8 は、`<portal:createPortletFloatURL>` JSP タグの使用方の例です。

コード リスト 2-8 <portal:createPortletFloatURL> タグの使用例

```
<%
    }
    if (PortletRenderHelper.isFloatable(request))
    {
        url = portletState.getUrl(Portlet.URL_CONTENT);
    }
%>
    <util:validURL url="<%=url%>">
<% request.setAttribute( "fullscreenPortletURL", url); %>

<td width="1%" valign="middle">
    <a href="javascript:submitForm();"

onClick="openBrWindow('<portal:createPortletFloatURL
httpsInd='http'/>', '<%= portletWindowName%>',
'scrollbars=yes,width=780,height=550'); return false;">
' />" hspace="1"
vspace="2" border="0" alt="Float" align="absmiddle"></a></td>
</util:validURL>
```

<portlet:getException>

| | |
|-----------------|--|
| タグ ライブラリ | portlet.tld |
| インポート文 | <%@ taglib uri="portlet.tld" prefix="portlet" %> |
| 実装されるクラス | GetExceptionTag GetExceptionTagExtraInfo |

`<portlet:getException>` タグは、Webflow プロセッサから送出される例外やメッセージを取得するのに使用されます。対象となり得るのは、`InvalidFormFieldException` や `ProcessingException` に関連するメッセージです。

このタグは、例外に対して `getMessage()` を呼び出す箇所にインラインに埋め込むことも、例外そのものを表すスクリプト変数を返すようにすることもできます。

表 2-11 は、`<portlet:getException>` タグの属性についての説明です。

表 2-11 `<portlet:getException>` タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------------|-----|--------|-----------------------------------|-----|
| <code>id</code> | いいえ | String | 例外のインスタンスを取り出すのに使える Java スクリプト変数。 | R |
| <code>type</code> | いいえ | String | 例外をキャストするのに使える Java クラス名。 | R |

ポータル タグ ライブラリ

以下の節では、ポータル タグとそれらの属性について説明します。

`<portal:createWebflowURL>`

| | |
|----------|--|
| タグライブラリ | <code>portal.tld</code> |
| インポート文 | <code><%@ taglib uri="portal.tld" prefix="portal" %></code> |
| 実装されるクラス | <code>CreatePortalURLTag</code> <code>CreateWebflowURLTagExtraInfo</code> |

`<portal:createWebflowURL>` タグは JSP で使用され、Webflow URL を JSP 内に動的に生成します。Webflow URL は、プロトコル、ドメイン名、ポート、Web アプリケーション URI、WebflowServlet の URI、およびクエリ文字列から構成されます。

表 2-12 は、`<portal:createWebflowURL>` タグの属性についての説明です。

表 2-12 <portal:createWebflowURL> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|---|-----|
| pageName | いいえ | String | Webflow URL が生成されるページ。 | R |
| origin | いいえ | String | イベントの出発点となるノード（起点）で、<ノード名>.<ノードタイプ>の形式に従う。ページ名と同じ場合も異なる場合もある。省略された場合、JSP ページ名が使われる。 | R |
| event | はい | String | Webflow は、これを起点と組み合わせて使って、指定されたネームスペース XML ファイル内の遷移先を決定する。 | R |
| domainName | いいえ | String | URL のドメイン名や IP アドレスを変更するのに使う。Webflow が別のマシン上のプロキシ サーバを経由する場合に、この属性を使用できる。 | R |
| pathprefix | いいえ | String | パス情報の接頭辞として使われる。この文字列は、Web アプリケーション URI の前に置かれる。プロキシ サーバが同一のマシン上にある場合に使われる。 注意： プロキシ サーバ側では、このパス接頭辞を取り除いてから Webflow にリクエストを転送する必要がある。 | R |
| pathsuffix | いいえ | String | パス情報の接尾辞として使われる。WebflowServlet URI に続けて、追加のパス情報が置かれる。これにより、この情報は、request.getPathInfo() メソッドから取得可能になる。 | R |
| extraParams | いいえ | String | リクエスト パラメータを名前と値のペアとして追加指定するのに使われる。 | R |
| doRedirect | いいえ | String | WebflowServlet にプレゼンテーションノードへの転送ではなくリダイレクトを行わせる。有効な値は true と false。デフォルトは false（リダイレクトしない）。 | R |

表 2-12 <portal:createWebflowURL> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|----------|-----|--------|---|-----|
| httpsInd | いいえ | String | <p>Webflow 側でプロトコルを推定するか、あるいは HTTPS または HTTP を使うように指定する。このため、有効な値は calculate、http、および https。デフォルト値は http。</p> <p>Calculate を指定すると、起点 / イベントの分岐連鎖のリストに含まれているいずれかのノードが、アプリケーションの META-INF/web.xml ファイル内の HTTPS_URL_PATTERNS コンテキストパラメータで指定されているパターンと一致する場合には、HTTPS が選択される。Calculate のほうがより動的で処理コストがかかるが、プロトコルを強制する必要がある場合には、それをここで指定することができる。</p> | R |
| encode | いいえ | String | <p>Webflow 側で URL をコード化するように指定する。セッションの状態を保持したいが、ブラウザがクッキーを受け付けない場合には、URL をコード化する必要がある。有効な値は true と false。デフォルト値は true。ただし、ブラウザがクッキーを受け付けない場合にしか、URL をコード化する必要はない。</p> | R |

<portal:form>

| | |
|-----------------|--|
| タグライブラリ | portal.tld |
| インポート文 | <%@ taglib uri="portal.tld" prefix="portal" %> |
| 実装されるクラス | PortalFormTag CreateWebflowURLTagExtraInfo |

<portal:form> タグは JSP で使用され、HTML フォーム タグを動的に生成します。このタグは <portal:validatedForm> タグほど複雑ではなく、もっと単純です。<portal:validatedForm> タグの詳細については、次の節を参照してください。

表 2-13 は、<portal:form> タグの属性についての説明です。

表 2-13 <portal:form> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------------|-----|--------|---|-----|
| pageName | いいえ | String | フォームが生成されるページ。 | R |
| origin | いいえ | String | イベントの出発点となるノード（起点）で、<ノード名>.<ノードタイプ>の形式に従う。ページ名と同じ場合も異なる場合もある。省略された場合、JSP ページ名が使われる。 | R |
| event | はい | String | Webflow は、これを起点と組み合わせて使って、指定されたネームスペース XML ファイル内の遷移先を決定する。 | R |
| domainName | いいえ | String | URL のドメイン名や IP アドレスを変更するのに使う。Webflow が別のマシン上のプロキシサーバを経由する場合に、この属性を使用できる。 | R |

表 2-13 <portal:form> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|--|-----|
| pathprefix | いいえ | String | <p>パス情報の接頭辞として使われる。この文字列は、Web アプリケーション URI の前に置かれる。プロキシ サーバが同一のマシン上にある場合に使われる。</p> <p>注意: プロキシ サーバ側では、このパス接頭辞を取り除いてから Webflow にリクエストを転送する必要がある。</p> | R |
| pathsuffix | いいえ | String | <p>パス情報の接尾辞として使われる。WebflowServlet URI に続けて、追加のパス情報が置かれる。これにより、この情報は、request.getPathInfo() メソッドから取得可能になる。</p> | R |
| extraParams | いいえ | String | <p>リクエストパラメータを名前と値のペアとして追加指定するのに使われる。</p> | R |
| doRedirect | いいえ | String | <p>WebflowServlet にプレゼンテーションノードへの転送ではなくリダイレクトを行わせる。有効な値は true と false。デフォルトは false (リダイレクトしない)。</p> | R |

表 2-13 <portal:form> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|----------|-----|--------|---|-----|
| httpsInd | いいえ | String | <p>Webflow 側でプロトコルを推定するか、あるいは HTTPS または HTTP を使うように指定する。このため、有効な値は calculate、http、および https。デフォルト値は http。</p> <p>Calculate を指定すると、起点 / イベントの分岐連鎖のリストに含まれているいずれかのノードが、アプリケーションの META-INF/web.xml ファイル内の HTTPS_URL_PATTERNS コンテキストパラメータで指定されているパターンと一致する場合には、HTTPS が選択される。Calculate のほうがより動的で処理コストがかかるが、プロトコルを強制する必要がある場合には、それをここで指定することができる。</p> | R |
| encode | いいえ | String | <p>Webflow 側で URL をコード化するように指定する。セッションの状態を保持したいが、ブラウザがクッキーを受け付けない場合には、URL をコード化する必要がある。有効な値は true と false。デフォルト値は true。ただし、ブラウザがクッキーを受け付けない場合にしか、URL をコード化する必要はない。</p> | R |
| hide | いいえ | String | <p>false に設定した場合、ネームスペース、起点、およびイベントは、隠しフォームフィールドとしてではなく、コマンドライン上に表示される。有効な値は true と false。デフォルト値は true。</p> | R |
| method | いいえ | String | <p>フォーム用に使われるメソッド。有効な値は get と post。デフォルト値は post。</p> | R |
| name | いいえ | String | <p>フォームの名前。</p> | R |

<portal:validatedForm>

| | |
|----------|--|
| タグ ライブラリ | portal.tld |
| インポート文 | <%@ taglib uri="portal.tld" prefix="portal" %> |
| 実装されるクラス | PortalValidateFormTag WebflowValidateFormTagExtraInfo |

<portal:validatedForm> タグは、有効性の検証が可能な HTML フォームを動的に生成するのに使用されます。Web サイトへの訪問者が無効な情報を入力すると、訪問者の入力はいったん保存され、適切なエラー メッセージを付けて再表示されます。

表 2-14 は、<portal:validatedForm> タグの属性についての説明です。

表 2-14 <portal:validatedForm> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------------|-----|--------|---|-----|
| pageName | いいえ | String | 検証機能付きフォームの生成元のページ。 | R |
| origin | いいえ | String | イベントの出発点となるノード（起点）で、<ノード名>.<ノードタイプ>の形式に従う。ページ名と同じ場合も異なる場合もある。省略された場合、JSP ページ名が使われる。 | R |
| event | はい | String | Webflow は、これを起点と組み合わせて使って、指定されたネームスペース XML ファイル内の遷移先を決定する。 | R |
| domainName | いいえ | String | URL のドメイン名や IP アドレスを変更するのに使う。Webflow が別のマシン上のプロキシサーバを経由する場合に、この属性を使用できる。 | R |

表 2-14 <portal:validatedForm> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|--|-----|
| pathprefix | いいえ | String | <p>パス情報の接頭辞として使われる。この文字列は、Web アプリケーション URI の前に置かれる。プロキシ サーバが同一のマシン上にある場合に使われる。</p> <p>注意: プロキシ サーバ側では、このパス接頭辞を取り除いてから Webflow にリクエストを転送する必要がある。</p> | R |
| pathsuffix | いいえ | String | <p>パス情報の接尾辞として使われる。WebflowServlet URI に続けて、追加のパス情報が置かれる。これにより、この情報は、request.getPathInfo() メソッドから取得可能になる。</p> | R |
| extraParams | いいえ | String | <p>リクエスト パラメータを名前と値のペアとして追加指定するのに使われる。</p> | R |
| doRedirect | いいえ | String | <p>WebflowServlet にプレゼンテーションノードへの転送ではなくリダイレクトを行わせる。有効な値は true と false。デフォルトは false (リダイレクトしない)。</p> | R |

表 2-14 <portal:validatedForm> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|----------|-----|--------|---|-----|
| httpsInd | いいえ | String | <p>Webflow 側でプロトコルを推定するか、あるいは HTTPS または HTTP を使うように指定する。このため、有効な値は <code>calculate</code>、<code>http</code>、および <code>https</code>。デフォルト値は <code>http</code>。</p> <p><code>Calculate</code> を指定すると、起点 / イベントの分岐連鎖のリストに含まれているいずれかのノードが、アプリケーションの <code>META-INF/web.xml</code> ファイル内の <code>HTTPS_URL_PATTERNS</code> コンテキストパラメータで指定されているパターンと一致する場合には、HTTPS が選択される。<code>Calculate</code> のほうがより動的で処理コストがかかるが、プロトコルを強制する必要がある場合には、それをここで指定することができる。</p> | R |
| encode | いいえ | String | <p>Webflow 側で URL をコード化するように指定する。セッションの状態を保持したいが、ブラウザがクッキーを受け付けない場合には、URL をコード化する必要がある。有効な値は <code>true</code> と <code>false</code>。デフォルト値は <code>true</code>。ただし、ブラウザがクッキーを受け付けない場合にしか、URL をコード化する必要はない。</p> | R |
| hide | いいえ | String | <p><code>false</code> に設定した場合、ネームスペース、起点、およびイベントは、隠しフォームフィールドとしてではなく、コマンドライン上に表示される。有効な値は <code>true</code> と <code>false</code>。デフォルト値は <code>true</code>。</p> | R |
| method | いいえ | String | <p>フォーム用に使われるメソッド。有効な値は <code>get</code> と <code>post</code>。デフォルト値は <code>post</code>。</p> | R |
| name | いいえ | String | <p>フォームの名前。</p> | R |

表 2-14 <portal:validatedForm> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------------------|-----|--------|---|-----|
| validStyle | いいえ | String | 有効な HTML フィールドを書式化するのに使用されるスタイル。 | R |
| invalidStyle | いいえ | String | HTML フィールドが無効な場合に、そのフィールドまたはメッセージを書式化するのに使用されるスタイル。 | R |
| unspecifiedStyle | いいえ | String | HTML フィールドの検証前の初期スタイルを指定するのに使われる。 | R |
| styleId | いいえ | String | フィールドのステータスが valid、invalid、unspecified のいずれであるかに応じて、invalidStyle、unspecifiedStyle、validStyle のいずれか 1 つに設定されるスクリプト変数。HTML フォームの書式化をより細かく制御するのに使用できる。 | R |
| applyStyle | いいえ | String | フィールドのステータスに関連付けられているスタイルをメッセージまたはフィールドに適用するか、あるいはそのいずれにも適用しない。このため、有効な値は message、field、および none。デフォルト値は message。 | R |
| messageAlign | いいえ | String | エラー メッセージの表示位置をフィールドの上、フィールドの右、フィールドの下のいずれかに指定する。このため、有効な値は top、right、および bottom。デフォルト値は right。 | R |

<portal:createPortalPageChangeURL>

| | |
|---------|--|
| タグライブラリ | portal.tld |
| インポート文 | <%@ taglib uri="portal.tld" prefix="portal" %> |

実装されるクラス CreatePageChangeURLTag

<portal:createPortalPageChangeURL> タグは、ページ変化イベントに対応する Webflow URL を生成します。

表 2-15 は、<portal:createPortalPageChangeURL> タグの属性についての説明です。

表 2-15 <portal:createPortalPageChangeURL> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|---|-----|
| pageName | いいえ | String | イベントの発生元のページ。 | R |
| origin | いいえ | String | イベントの出発点となるノード（起点）で、<ノード名>.<ノードタイプ>の形式に従う。ページ名と同じ場合も異なる場合もある。省略された場合、JSP ページ名が使われる。 | R |
| domainName | いいえ | String | URL のドメイン名や IP アドレスを変更するのに使う。Webflow が別のマシン上のプロキシサーバを経由する場合に、この属性を使用できる。 | R |
| pathprefix | いいえ | String | パス情報の接頭辞として使われる。この文字列は、Web アプリケーション URI の前に置かれる。プロキシサーバが同一のマシン上にある場合に使われる。 注意： プロキシサーバ側では、このパス接頭辞を取り除いてから Webflow にリクエストを転送する必要がある。 | R |
| pathsuffix | いいえ | String | パス情報の接尾辞として使われる。WebflowServlet URI に続けて、追加のパス情報が置かれる。これにより、この情報は、request.getPathInfo() メソッドから取得可能になる。 | R |
| extraParams | いいえ | String | リクエストパラメータを名前と値のペアとして追加指定するのに使われる。 | R |

表 2-15 <portal:createPortalPageChangeURL> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------------|-----|--------|--|-----|
| doRedirect | いいえ | String | WebflowServlet にプレゼンテーションノードへの転送ではなくリダイレクトを行わせる。有効な値は true と false。デフォルトは false (リダイレクトしない)。 | R |
| httpsInd | いいえ | String | Webflow 側でプロトコルを推定するか、あるいは HTTPS または HTTP を使うように指定する。このため、有効な値は calculate、http、および https。デフォルト値は http。 Calculate を指定すると、起点 / イベントの分岐連鎖のリストに含まれているいずれかのノードが、アプリケーションの META-INF/web.xml ファイル内の HTTPS_URL_PATTERNS コンテキストパラメータで指定されているパターンと一致する場合には、HTTPS が選択される。Calculate のほうがより動的で処理コストがかかるが、プロトコルを強制する必要がある場合には、それをここで指定することができる。 | R |
| encode | いいえ | String | Webflow 側で URL をコード化するように指定する。セッションの状態を保持したいが、ブラウザがクッキーを受け付けない場合には、URL をコード化の必要がある。有効な値は true と false。デフォルト値は true。ただし、ブラウザがクッキーを受け付けない場合にしか、URL をコード化する必要はない。 | R |

例

コードリスト 2-9 は、<portal:createPortalPageChangeURL> JSP タグの使用方法の例です。

コード リスト 2-9 <portal:createPortalPageChangeURL> タグの使用例

```
        if (DISPLAY_TEXT_LINKS)
        {
            if (portalPageName.equals(selectedPage))
            {
                %>
                <td class="tabselected" nowrap align="center">
                <b>&nbsp;</b>
                <a href="<ptl:createPortalPageChangeURL pageName='<%=
                portalPageName %>' />"><span
                class="tabselected"><%=portalPageName%></span></a></b></td>
                <td class="tabselected" WIDTH="2"><b>&nbsp;</b></td>
                <%
            }
        }
        else
        {
            %>
            <td class="tabunselected" nowrap align="center">
            <b>&nbsp;</b>
            <a href="<ptl:createPortalPageChangeURL pageName='<%=
            portalPageName %>' />"><span
            class="tabunselected"><%=portalPageName%></span></a></b></td>
            <td class="tabunselected" WIDTH="2"><b>&nbsp;</b></td>
            <%
        }
    }
}
```

ユーティリティ タグ ライブラリ

以下の節では、ユーティリティ タグとそれらの属性について説明します。

<util:validURL>

| | |
|----------|--|
| タグ ライブラリ | util.tld |
| インポート文 | <%@ taglib uri="util.tld" prefix="util" %> |
| 実装されるクラス | ValidURLTag |

指定された URL が null でなく、かつ 4 文字以上である場合に、タグの本体を処理します。表 2-16 は、<util:validURL> タグの属性についての説明です。

表 2-16 <util:validURL> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------|----|--------|------------------------|-----|
| URL | はい | String | 指定される URL。有効であれば処理される。 | R |

<util:invalidURL>

| | |
|----------|--|
| タグ ライブラリ | util.tld |
| インポート文 | <%@ taglib uri="util.tld" prefix="util" %> |
| 実装されるクラス | InvalidURLTag |

`<utility:invalidURL>` タグは、指定された URL が null か、あるいは 4 文字未満の場合に、タグの本体を処理します。表 2-17 は、`<util:invalidURL>` タグの属性についての説明です。

表 2-17 `<util:invalidURL>` タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------|----|--------|-----------|-----|
| URL | はい | String | 指定される URL | R |

例

以下のサンプル コードは、`<utility:invalidURL>` JSP タグと `<utility:validURL>` JSP タグの使用方法の例です。

コードリスト 2-10 `<utility:invalidURL>` と `<utility:validURL>` タグの使用例

```
<%
// まず、代替ヘッダーの取得を試み、それが存在しなければ
// 通常のヘッダーを使用する

url = portletState.getUrl(Portlet.URL_ALTERNATE_HEADER);
debug.out("Alternate header:" + url);
%>
<util:invalidURL url="<%=url %>">
  <% url = portletState.getUrl(Portlet.URL_HEADER); %>
</util:invalidURL>
<util:validURL url="<%=url %>">
  <table cellpadding="4" cellspacing="0" width="100%"
    class="portlethead">
    <tr>
      <td width="100%">
        <jsp:include page="<%=url%>" />
      </td>
    </tr>
  </table>
</util:validURL>
<%
```

第3章 パーソナライゼーション JSP タグ

WebLogic Portal に用意されている JSP タグを使用すると、開発者は、Java を使ってプログラミングしなくてもパーソナライズされたアプリケーションを作成できます。

この節では、以下の JSP タグに関する情報を示します。

- [広告](#)
- [コンテンツ管理](#)
- [国際化](#)
- [パーソナライゼーション タグ](#)
- [ブレースホルダ](#)
- [プロパティ セット](#)
- [ユーザ管理 : プロファイル管理タグ](#)
- [ユーザ管理 : グループおよびユーザ管理タグ](#)
- [ユーザ管理 : セキュリティ タグ](#)
- [ユーティリティ タグ : パーソナライゼーション ユーティリティ](#)
- [ユーティリティ タグ : WebLogic ユーティリティ](#)

広告

広告タグは、コンテンツ管理システムにクエリを発行して広告を表示します。

以降の表の「必須」列では、属性が必須と省略可能のどちらであるかを「はい」または「いいえ」で示します。「R/C」列では、「C」は属性がコンパイル時表現であることを、「R」は属性が要求時表現とコンパイル時表現のいずれかであることを示します。

<ad:adTarget>

| | |
|----------|--|
| タグライブラリ | ad.tld |
| インポート文 | <%@ taglib uri="ad.tld" prefix="ad" %> |
| 実装されるクラス | adTargetTag |

<ad:adTarget> タグ (表 3-1) は、広告サービスを使用して、広告クエリをコンテンツ管理システムに送信します。<ph:placeholder> タグとは異なり、<ad:adTarget> タグは広告プレースホルダ内の他のクエリと競合しません。

このタグは、顧客に向けて特定の場所に確実に広告を表示する必要がある場合に使用します。どれだけ特化したクエリを作成するかにもよりますが、別の広告を表示する場合にはこのタグを削除または変更しなければならないことがあります。

広告クエリが複数の広告を返す場合、広告サービスは、表示する広告を各広告の adWeight 属性に基づいて決定します。

タグとコンテンツ管理サービス プロバイダ インタフェースとのマッピング

以下の表は、タグが、コンテンツ管理サービス プロバイダ インタフェース (SPI) 内のメソッドにアクセスする方法を示したものです。この表では、呼び出しの順序についても示しています。

| EJB/ データ オブジェクト | SPI |
|---|--|
| 1. AdService.getContent() | 3. DocumentProvider.findDocumentMetadata() |
| 2. DocumentManager.getContent() | 4. DocumentMetadataDef.getID() |
| 8. Document.getProperty() ... (その他の get* メソッド) | 5. DocumentMetadataDef.getName() ... (その他の get* メソッド) |
| 9. Document.getContent() | 6. DocumentMetadataDef.getAttributeNames() |
| 10. DocumentManager.getContentBlock() | 7. DocumentMetadataDef.getAttribute() |
| | 11. DocumentProvider.getDocument() |
| | 12. DocumentDef.openStream() |

タグの属性

表 3-1 は、<ad:adTarget> タグの属性についての説明です。

表 3-1 <ad:adTarget> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------|----|--------|--|-----|
| query | はい | String | <p>広告サービスがコンテンツを検索するために使用するクエリを格納する。 com.beasys.commerce.util.ExpressionHelper の Javadoc で説明されているクエリ構文を使用する。 クエリ構築の詳細については、 http://edocs.beasys.co.jp/e-docs/wlp/docs70/dev/conmgmt.htm にある『開発者ガイド』の「コンテンツ クエリの作成」を参照。</p> | R |

表 3-1 <ad:adTarget> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------|-----|------|---|-----|
| height | いいえ | int | <p>ブラウザでドキュメントを表示するために必要な HTML を生成するときには、プレースホルダが占める高さをピクセル単位で指定する。</p> <p>プレースホルダでは、表示サイズが適用されるコンテンツタイプだけを対象に、他の属性によってそのドキュメントに対して定義されたサイズがまだない場合にのみこの値が使用される。</p> <p>この値を指定せず、他の属性によって定義されてもいない場合、ドキュメントの高さはブラウザの動作によって決定される。</p> | R |
| width | いいえ | int | <p>ブラウザでドキュメントを表示するために必要な HTML を生成するときには、プレースホルダが占める幅をピクセル単位で指定する。</p> <p>プレースホルダでは、表示サイズが適用されるコンテンツタイプだけを対象に、他の属性によってそのドキュメントに対して定義されたサイズがまだない場合にのみこの値が使用される。</p> <p>この値を指定せず、他の属性によって定義されてもいない場合、ドキュメントの幅はブラウザの動作によって決定される。</p> | R |

例

コードリスト 3-1 では、広告グループ「Car」内の広告を 1 つ選択し、200 × 400 ピクセルの領域内に描画します。

コードリスト 3-1 <ad:adTarget> タグの使用例

```
<%@ taglib uri="ad.tld" prefix="ad" %>
:
```

```
<ad:adTarget query="group == 'ads'" />
```

コンテンツ管理

コンテンツ管理コンポーネントには、4種類の JSP タグが用意されています。JSP 開発者はこれらのタグを使用して、パーソナライズされていないコンテンツを HTML ベースのページに組み込むことができます。cm:select タグと cm:selectbyid タグでは、コンテンツ検索時にコンテンツをキャッシュすることができます。これらのタグはどちらも本体をサポートせず、また使うこともありません。

以降の表の「必須」列では、属性が必須と省略可能のどちらであるかを「はい」または「いいえ」で示します。「R/C」列では、「C」は属性がコンパイル時表現であることを、「R」は属性が要求時表現とコンパイル時表現のいずれかであることを示します。

<cm:getProperty>

| | |
|----------|--|
| タグライブラリ | cm.tld |
| インポート文 | <%@ taglib uri="cm.tld" prefix="cm" %> |
| 実装されるクラス | getPropertyTag getPropertyExtraInfo |

<cm:getProperty> タグは、指定されたコンテンツ メタデータ プロパティの値を取り出して、resultId で指定された変数に代入します。このタグには本体はありません。resultId が指定されない場合、<cm:printProperty> タグと同様に、値はページにインラインで埋め込まれます。このタグは、Content オブジェクトだけでなく、任意の ConfigurableEntity に作用します。ただし、ConfigurableEntity サクセサはサポートされません。

タグとコンテンツ管理サービス プロバイダ インタフェースとのマッピング

このタグは、コンテンツ管理システムからすでに取り出してあるデータを受け取るため、コンテンツ管理サービス プロバイダ インタフェース (SPI) への呼び出しは行いません。

タグの属性

表 3-2 は、`<cm:getProperty>` タグの属性についての説明です。

表 3-2 `<cm:getProperty>`

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|----------|-----|--------------------|---|-----|
| id | いいえ | String | 取得するプロパティが入った Content のインスタンスを格納している JSP スクリプト変数の名前。 | R |
| entity | いいえ | ConfigurableEntity | 取得するプロパティが入った <code>com.beasys.commerce.foundation.ConfigurableEntity</code> オブジェクトを指定する。この属性に null でない値が指定された場合、id 属性は無視される。そうでない場合には、id の値が使用される。 | R |
| name | はい | String | 出力するプロパティの名前。 | R |
| scope | いいえ | String | 取得するプロパティのスコープ名。指定しない場合、Document オブジェクトが想定する null が渡される。 | R |
| resultId | いいえ | String | プロパティの値が入る JSP スクリプト変数の名前。指定されない場合、プロパティの値は JSP の本体にインラインで埋め込まれる。指定される場合、 <code>encode</code> 、 <code>default</code> 、 <code>maxLength</code> 、 <code>dateFormat</code> 、および <code>numFormat</code> の各属性は無視される。 | C |

表 3-2 <cm:getProperty> (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------------|-----|-------------|---|-----|
| resultType | いいえ | String | プロパティの Java データ型。指定されない場合、 <code>java.lang.Object</code> が使用される。 | C |
| encode | いいえ | String | <p><i>html</i>、<i>url</i>、または <i>none</i> のいずれか。</p> <ul style="list-style-type: none"> ◆ <i>html</i> の場合、作成者の意図どおりに HTML 内で表現されるように、値が HTML 形式にコード化される (「&」は「&amp;」に、「<」は「&lt;」に、「>」は「&gt;」に、「"」は「&quot;」に変換される)。 ◆ <i>url</i> の場合、値は、<code>java.net.URLEncoder</code> を通じて <code>x-www-form-urlencoded</code> 形式にコード化される。 ◆ <i>none</i> または指定なしの場合、コード化は行われない。 | R |
| default | いいえ | String | プロパティが見つからなかった場合や値が <code>null</code> の場合に出力される値。この属性が未指定でプロパティ値が <code>null</code> の場合、何も出力されない。 | R |
| maxLength | いいえ | String, int | 出力するプロパティ値の最大長。この属性を指定した場合、最大長よりも長い値は切り詰められる。 | R |

表 3-2 <cm:getProperty> (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|-----------------|---|-----|
| failOnError | いいえ | String, Boolean | この属性の取り得る値は、以下の 2 つのいずれか。 false (デフォルト値): JSP 処理エラーを内々に処理し、エラーが発生しても何も出力しない。 true: 例外を送出する。その例外をコード内で処理することも、通常のエラーページを表示させることも、あるいはアプリケーション サーバにその例外を外部にわかるように処理させることもできる。 | R |
| dateFormat | いいえ | String | プロパティが java.util.Date の場合に、それを出力するために用いられる java.text.SimpleDateFormat 文字列。プロパティが Date でない場合には、この属性は無視される。この属性を設定しない場合、Date のデフォルトの toString メソッドが使用される。 | R |
| numFormat | いいえ | String | プロパティが java.lang.Number の場合に、それを出力するために用いられる java.text.DecimalFormat 文字列。プロパティが Number でない場合には、この属性は無視される。この属性を設定しない場合、Number のデフォルトの toString メソッドが使用される。 | R |

例

コード リスト 3-2 は <cm:getProperty> タグの使用方法の例です。doc に格納された Content オブジェクトから name プロパティの String 値を取得し、その値を contentName 変数に代入します。

コードリスト 3-2 <cm:getProperty> タグの使用例

```

<%@ taglib uri="cm.tld" prefix="cm" %>
.
.
.
<cm:getProperty resultId="contentName" resultType="String"
  id="content" name="name" />
<es:NotNull item="<%=contentName%>">
The name is not null.
</es:NotNull>

```

<cm:printDoc>

| | |
|-----------------|--|
| タグライブラリ | cm.tld |
| インポート文 | <%@ taglib uri="cm.tld" prefix="cm" %> |
| 実装されるクラス | printDocTag |

<cm:printDoc> タグは、Document オブジェクトの未処理バイトを JSP 出力ストリームにインラインで埋め込みます。このタグは本体をサポートしておらず、また Document オブジェクトだけをサポートします。このタグでは、テキストとバイナリ データは区別されません。

タグとコンテンツ管理サービス プロバイダ インタフェースとのマッピング

以下の表は、タグが、コンテンツ管理サービス プロバイダ インタフェース (SPI) 内のメソッドにアクセスする方法を示したものです。この表では、呼び出しの順序についても示しています。

| EJB/ データ オブジェクト | SPI |
|--------------------------------------|-----------------------------------|
| 1. Document.getIdentifier() | 5. DocumentProvider.getDocument() |
| 2. Document.getPropertyAsString() | 6. DocumentDef.openStream() |
| 3. Document.getContent() | |
| 4. DocumentManager.getContentBlock() | |
| 7. Document.getSize() | |

タグの属性

表 3-3 は、<cm:printDoc> タグの属性についての説明です。

表 3-3 <cm:printDoc> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-----------|-----|-------------|---|-----|
| id | いいえ | String | 取得するプロパティが入った Content のインスタンスを格納している JSP スクリプト変数の名前。 | R |
| blockSize | いいえ | String, int | 読み込むデータ ブロックのサイズ。デフォルトは 8K。バイト ブロック全体を 1 回の操作で読み込むには、0 以下の値を指定する。 | R |
| start | いいえ | String, int | バイト データ内の読み込み開始位置を示すインデックスを指定する。デフォルトは 0。 | R |
| end | いいえ | String, int | バイト データ内の読み込み終了位置を示すインデックスを指定する。デフォルトでは、バイト データの最後まで読み込む。 | R |

表 3-3 <cm:printDoc> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|-----------------|---|-----|
| encode | いいえ | String | <p>html、url、none のいずれか。</p> <ul style="list-style-type: none"> ◆ html の場合、作成者の意図どおりに HTML 内で表現されるように値が HTML 形式にコード化される (「&」は「&amp;」に、「<」は「&lt;」に、「>」は「&gt;」に、「"」は「&quot;」に変換される)。 ◆ url の場合、値は、<code>java.net.URLEncoder</code> を通じて <code>x-www-form-urlencoded</code> 形式にコード化される。 ◆ none または未指定の場合、コード化は行われない。 | R |
| document | いいえ | Document | <p>使用する <code>com.bea.p13n.content.document.Document</code> を指定する。この属性に null 以外の値を指定した場合、id 属性は無視される。そうでない場合には、id の値が使用される。</p> | R |
| failOnError | いいえ | String, Boolean | <p>この属性の取り得る値は、以下の 2 つのいずれか。</p> <p><code>false</code> (デフォルト値): JSP 処理エラーを内々に処理し、エラーが発生しても何も出力しない。</p> <p><code>true</code>: 例外を送出する。その例外をコード内で処理することも、通常のエラーページを表示させることも、あるいはアプリケーション サーバにその例外を外部にわかるように処理させることもできる。</p> | R |
| baseHref | いいえ | String | <p>ドキュメントの BASE HREF の URL。絶対 URL または相対 URL のどちらでも指定できる。</p> | R |

注意: `baseHref` が指定されている場合、`<cm:printDoc>` タグは、`baseHref` パラメータの値を使用して起点の `<BASE HREF>` を出力します。`baseHref` の URL が不完全な場合、欠けている部分は、最も外側のページの URL に基づいて補われます。

また、`baseHref` が指定されている場合、`<cm:printDoc>` タグはドキュメントの出力後、最も外側のページの URL に基づいて `<BASE HREF>` を出力します。

例

コードリスト 3-3 は `<cm:printDoc>` タグの使用法の例です。`request` 属性内の `id` から Document オブジェクトを取得し、その Document のテキスト（相対リンクを含む場合がある）をインラインに埋め込みます。

コード リスト 3-3 `<cm:printDoc>` タグの使用例

```
<%@ taglib uri="cm.tld" prefix="cm" %>
.
.
.
.<% String contentId = request.getParameter("contentId"); %>
<cm:selectById contentId="<%=contentId%>" id="doc" />
<cm:printDoc id="doc" blockSize="1000" baseHref="/ShowDocServlet"
/>
```

`<cm:printProperty>`

| | |
|----------|--|
| タグ ライブラリ | cm.tld |
| インポート文 | <%@ taglib uri="cm.tld" prefix="cm" %> |
| 実装されるクラス | printPropertyTag |

`<cm:printProperty>` タグでは、指定されたコンテンツ メタデータ プロパティの値を文字列としてインラインに埋め込みます。このタグには本体はありません。このタグは、Content オブジェクトだけでなく、任意の `ConfigurableEntity` に作用します。ただし、`ConfigurableEntity` サクセサはサポートされません。

タグとコンテンツ管理サービス プロバイダ インタフェースとのマッピング

このタグは、コンテンツ管理システムからすでに取り出してあるデータを受け取るため、コンテンツ管理サービス プロバイダ インタフェース (SPI) への呼び出しは行いません。

タグの属性

表 3-4 は、`<cm:printProperty>` タグの属性についての説明です。

表 3-4 `<cm:printProperty>`

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------|-----|--------------------|---|-----|
| id | いいえ | String | 取得するプロパティが入った Content のインスタンスを格納している JSP スクリプト変数の名前。 | R |
| name | はい | String | 出力するプロパティの名前。 | R |
| entity | いいえ | ConfigurableEntity | 取得するプロパティが入った <code>com.beasys.commerce.foundation.ConfigurableEntity</code> オブジェクトを指定する。この属性に null でない値が指定された場合、id 属性は無視される。そうでない場合には、id の値が使用される。 | R |
| scope | いいえ | String | 取得するプロパティの範囲名。指定しない場合、Document オブジェクトが想定する null が渡される。 | R |

表 3-4 <cm:printProperty> (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|-----------------|---|-----|
| encode | いいえ | String | <p><i>html</i>、<i>url</i>、または <i>none</i> のいずれか。</p> <ul style="list-style-type: none"> ◆ <i>html</i> の場合、作成者の意図どおりに HTML 内で表現されるように、値が HTML 形式にコード化される (「&」は「&amp;」に、「<」は「&lt;」に、「>」は「&gt;」に、「"」は「&quot;」に変換される)。 ◆ <i>url</i> の場合、値は、<code>java.net.URLEncoder</code> を通じて <code>x-www-form-urlencoded</code> 形式にコード化される。 ◆ <i>none</i> または指定なしの場合、コード化は行われない。 | R |
| default | いいえ | String | プロパティが見つからなかった場合や値が <code>null</code> の場合に出力される値。この属性が未指定でプロパティ値が <code>null</code> の場合、何も出力されない。 | R |
| maxLength | いいえ | String, int | 出力するプロパティ値の最大長。この属性を指定した場合、最大長よりも長い値は切り詰められる。 | R |
| failOnError | いいえ | String, Boolean | <p>この属性の取り得る値は、以下の 2 つのいずれか。</p> <p><code>false</code> (デフォルト値): JSP 処理エラーを内々に処理し、エラーが発生しても何も出力しない。</p> <p><code>true</code>: 例外を送出する。その例外をコード内で処理することも、通常のエラーページを表示させることも、あるいはアプリケーション サーバにその例外を外部にわかるように処理させることもできる。</p> | R |

表 3-4 <cm:printProperty> (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------------|-----|--------|--|-----|
| dateFormat | いいえ | String | プロパティが java.util.Date の場合に、それを出力するために用いられる java.text.SimpleDateFormat 文字列。プロパティが Date でない場合には、この属性は無視される。この属性を設定しない場合、Date のデフォルトの toString メソッドが使用される。 | R |
| numFormat | いいえ | String | プロパティが java.lang.Number の場合に、それを出力するために用いられる java.text.DecimalFormat 文字列。プロパティが Number でない場合には、この属性は無視される。この属性を設定しない場合、Number のデフォルトの toString メソッドが使用される。 | R |

例

コードリスト 3-4 は、<cm:printProperty> タグの使用方法の例です。テキスト入力フィールドのデフォルト値を、doc に格納された Content オブジェクトの件名の先頭 75 文字に設定します。

コードリスト 3-4 <cm:printProperty> タグの使用例

```
<%@ taglib uri="cm.tld" prefix="cm" %>
.
.
.
<form action="javascript:void(0)">
  Subject: <input type="text" size="75" name="subject"
    value="<cm:printProperty id="doc" name="Subject" maxLength="75"
    encode="html"/>" >
</form>
```

<cm:select>

| | |
|-----------------|--|
| タグライブラリ | cm.tld |
| インポート文 | <%@ taglib uri="cm.tld" prefix="cm" %> |
| 実装されるクラス | selectTag selectExtraInfo |

このタグでは、検索式のクエリ構文だけを使用してコンテンツを選択します。このタグは本体をサポートせず、また使うこともありません。このタグが <es:forEachInArray> タグ (3-98 ページの「<es:forEachInArray>」を参照) を返した後で、Content オブジェクトの配列の反復処理にゼロを使用できます。このタグは、ContentManager インタフェースを通じて、汎用の Content をサポートします。

タグとコンテンツ管理サービス プロバイダ インタフェースとのマッピング

以下の表は、タグが、コンテンツ管理サービス プロバイダ インタフェース (SPI) 内のメソッドにアクセスする方法を示したものです。この表では、呼び出しの順序についても示しています。

| EJB/ データ オブジェクト | SPI |
|---------------------------------|---|
| 1. DocumentManager.getContent() | 2. DocumentProvider.findDocumentMetadata() |
| | 3. DocumentMetadataDef.getID() |
| | 4. DocumentMetadataDef.getName() ... (その他の get* メソッド) |
| | 5. DocumentMetadataDef.getAttributeNames() |
| | 6. DocumentMetadataDef.getAttribute() |

タグの属性

表 3-5 は、`<cm:select>` タグの属性についての説明です。

表 3-5 `<cm:select>` タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|-----------------|---|-----|
| contentHome | いいえ | String | コンテンツの検索に使用する ContentManager EJB ホームの JNDI 名。この名前の JNDI オブジェクトは、ContentManager インタフェースを実装するオブジェクトを返す create メソッドを実装しなければならない。この属性を指定しない場合、システムはデフォルトのコンテンツ ホームを探索する。 | R |
| max | いいえ | String, long | 返されるコンテンツ項目の最大数を制限する。この属性を指定しないか、ゼロ以下の値を指定した場合、見つかったコンテンツ項目がすべて返される。 | R |
| sortBy | いいえ | String | コンテンツのソートに使用するドキュメント属性を列挙する。形式は SQL の <i>order by</i> 句の構文に従う。ソートの指定は、メタデータ属性名とキーワード ASC および DESC の列挙に限られる。 例： sortBy="creationDate" sortBy="creationDate ASC, title DESC" | R |
| failOnError | いいえ | String, Boolean | この属性の取り得る値は、以下の 2 つのいずれか。 false (デフォルト値): JSP 処理エラーを内々に処理し、エラーが発生すると空の配列を返す。 true: 例外を送出し、それによって JSP ページの処理が停止する。その例外をコード内で処理することも、通常のエラーページを表示させることも、あるいはアプリケーション サーバにその例外を外部にわかるように処理させることもできる。 | R |

表 3-5 <cm:select> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------------|-----|-----------------|---|-----|
| id | はい | String | このタグの処理終了後に Content オブジェクトの配列が入る JSP スクリプト変数の名前。 | C |
| query | いいえ | String | コンテンツの検索に使われるコンテンツクエリ文字列。 例 : query="mimetype contains 'text' && author='Proulx'". クエリ構築の詳細については、 http://edocs.beasys.co.jp/e-docs/wlp/docs70/dev/conmngmt.htm の『開発者ガイド』の「コンテンツクエリの作成」を参照。 | R |
| expression | いいえ | Expression | コンテンツの検索に使用する com.beasys.commerce.foundation.expression.Expression オブジェクト。この属性が null または未指定の場合、query 属性を指定しなければならない。この属性に null 以外の値を指定した場合、query 属性は無視される。 | R |
| useCache | いいえ | String, Boolean | Content をキャッシュするかどうかを指定する。 この属性の取り得る値は、以下の 2 つのいずれか。 false (デフォルト値) : ContentCache は使用されない。false (または未指定) の場合、cacheId、cacheScope、および cacheTimeout の各属性の設定は無視される。 true: ContentCache が使われる。 | R |
| cacheId | いいえ | String | Content をキャッシュするために使われる識別子名。内部的には、キャッシュは Map として実装され、この属性はそのキーとなる。この属性を指定しない場合、タグの id 属性が使われる。 | R |

表 3-5 <cm:select> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|---------------|-------|--------------------------|--|-----|
| cacheTimeout | いいえ | String, long | キャッシュされた Content の有効時間 (ミリ秒単位)。Content がキャッシュされてからこの時間が経過すると、キャッシュされた Content はクリアされ、再び取得されてからキャッシュに置かれる。 タイムアウトしない (常にキャッシュされた Content を使用する) 場合には -1 を指定する。デフォルトは -1。 | R |
| cacheScope | いいえ | String | コンテンツ キャッシュのライフサイクル スコープを指定する。<jsp:useBean>と同様。 この変数が取り得る値： ◆ application ◆ session (デフォルト) ◆ page ◆ request | R |
| contextParams | いいえ | String または java.util.Map | ContentManager に渡す追加の検索パラメータ。ContentManager の実装によっては、この属性がサポートされていることがある。 | R |
| readOnly | 無視される | | この属性は非推奨となり、今後は使用されない。この属性が見つかっても無視される。 | |

例

コードリスト 3-5 は <cm:select> タグの使用方法の例です。ContentCache を使用して、その日の夕方のニュース記事であることを示す Content オブジェクトの最初の 5 つのテキストを検索し、それらのタイトルをリストに出力します。

コードリスト 3-5 <cm:select> タグの使用例

```
<%@ taglib uri="cm.tld" prefix="cm" %>
.
.
.
<cm:select
contentHome="<%=ContentHelper.DEF_CONTENT_MANAGER_HOME%>" max="5"
useCache="true" cacheTimeout="300000" cacheId="Evening News"
sortBy="creationDate ASC, title ASC" query="
    type = 'News' && timeOfDay = 'Evening' && mimetype like
    'text/*' " id="newsList"/>

<ul>
  <es:forEachInArray array="<%=newsList%>" id="newsItem"
    type="com.bea.pl3n.content.Content">
    <li><cm:printProperty id="newsItem" name="Title"
      encode="html" />
    </es:forEachInArray>
</ul>
```

<cm:selectById>

| | |
|-----------------|--|
| タグライブラリ | cm.tld |
| インポート文 | <%@ taglib uri="cm.tld" prefix="cm" %> |
| 実装されるクラス | selectByIdTag selectByIdExtraInfo |

<cm:selectById> タグ (表 3-6) は、Content の固有の識別子を使ってコンテンツを検索します。このタグには本体はありません。このタグは基本的には select タグのラッパーです。文字列で指定可能な主キーを持つ、すべての Content オブジェクトに対して機能します。

タグとコンテンツ管理サービス プロバイダ インタフェースとのマッピング

以下の表は、タグが、コンテンツ管理サービス プロバイダ インタフェース (SPI) 内のメソッドにアクセスする方法を示したものです。この表では、呼び出しの順序についても示しています。

| EJB/ データ オブジェクト | SPI |
|--|--|
| 1. <code>DocumentManager.getContent()</code> | 2. <code>DocumentProvider.findDocumentMetadata()</code> |
| | 3. <code>DocumentMetadataDef.getID()</code> |
| | 4. <code>DocumentMetadataDef.getName() ... (その他の get* メソッド)</code> |
| | 5. <code>DocumentMetadataDef.getAttributeNames()</code> |
| | 6. <code>DocumentMetadataDef.getAttribute()</code> |

タグの属性

表 3-6 は、`<cm:selectById>` タグの属性についての説明です。

表 3-6 `<cm:selectById>`

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------------------------|-----|--------|--|-----|
| <code>contentHome</code> | いいえ | String | コンテンツの検索に使用する <code>ContentManager EJB</code> ホームの JNDI 名。この名前の JNDI オブジェクトは、 <code>ContentManager</code> インタフェースを実装するオブジェクトを返す <code>create</code> メソッドを実装しなければならない。この属性を指定しない場合、システムはデフォルトのコンテンツ ホームを探索する。 | R |
| <code>contentId</code> | はい | String | コンテンツの断片の文字列識別子。 | R |

表 3-6 <cm:selectById> (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|-----------------|--|-----|
| onNotFound | いいえ | String | contentId で指定されたコンテンツ オブジェクトが見つからない場合の振る舞いを制御する。この属性が設定されていれば、その値をメッセージとして例外 (Exception) が送出され、指定されていない場合はタグは null を返す。 | R |
| failOnError | いいえ | String, Boolean | この属性の取り得る値は、以下の 2 つのいずれか。 false (デフォルト値): JSP 処理エラーを内々に処理し、エラーが発生した場合 null を返す。 true: 例外を送出し、それによって JSP ページの処理が停止する。その例外をコード内で処理することも、通常のエラーページを表示させることも、あるいはアプリケーション サーバにその例外を外部にわかるように処理させることもできる。 | R |
| id | はい | String | このタグの処理終了後に Content オブジェクトが入る JSP スクリプト変数の名前。指定された識別子を持つ Content オブジェクトが存在しなければ、変数には null が代入される。 | C |
| useCache | いいえ | String, Boolean | Content をキャッシュするかどうかを指定する。 この属性の取り得る値は、以下の 2 つのいずれか。 false (デフォルト値): ContentCache は使用されない。false (または未指定) の場合、cacheId、cacheScope、および cacheTimeout の各属性の設定は無視される。 true: ContentCache が使われる。 | R |

表 3-6 <cm:selectById> (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|---------------|-------|--------------------------|--|-----|
| cacheId | いいえ | String | Content をキャッシュするために使われる識別子名。内部的には、キャッシュは Map として実装され、この属性はそのキーとなる。 この属性を指定しない場合、タグの id 属性が使われる。 | R |
| cacheTimeout | いいえ | String, long | キャッシュされた Content の有効時間 (ミリ秒単位)。Content がキャッシュされてからこの時間が経過すると、キャッシュされた Content はクリアされ、再び取得されてからキャッシュに置かれる。 タイムアウトしない (常にキャッシュされた Content を使用する) 場合には -1 を指定する。デフォルトは -1。 | R |
| cacheScope | いいえ | String | コンテンツ キャッシュのライフサイクル スコープを指定する。<jsp:useBean>と同様。 この変数が取り得る値： ◆ application ◆ session (デフォルト) ◆ page ◆ request | R |
| contextParams | いいえ | String または java.util.Map | ContentManager に渡す追加の検索パラメータ。ContentManager の実装によっては、この属性がサポートされていることがある。 | R |
| readOnly | 無視される | | この属性は非推奨となり、今後は使用されない。この属性が見つかっていても無視される。 | |

例

コードリスト 3-6 は `<cm:selectById>` タグの使用法の例です。コンテンツキャッシュを有効にし、識別子 `1234` を指定して Document をフェッチし、そのコンテンツをインラインで埋め込みます。

コードリスト 3-6 `<cm:selectById >` タグの使用例

```
<%@ taglib uri="cm.tld" prefix="cm" %>
.
.
.
<cm:selectById
contentHome="<%=ContentHelper.DEF_CONTENT_MANAGER_HOME%>"
contentId="contentportlet/sports1.htm"
id="doc" useCache="true" cacheTimeout="30000" cacheId="1234" />
<cm:printDoc id="doc" />
```

国際化

これらのタグは、国際化する必要のある JSP ページをローカライズする際に使われます。

以降の表の「必須」列では、属性が必須と省略可能のどちらであるかを「はい」または「いいえ」で示します。「R/C」列では、「C」は属性がコンパイル時表現であることを、「R」は属性が要求時表現とコンパイル時表現のいずれかであることを示します。

注意： `<i18n>` タグを使用して、JSP で表示するためのローカライズされたコンテンツを含むリソースバンドルファイルを指定することができます。リソースバンドルコンテンツが更新されているかどうかを WebLogic Portal がチェックする間隔設定についての詳細は、<http://edocs.beasys.co.jp/e-docs/wlp/docs70/admin/sysadmin.htm> にある『*管理者ガイド*』の「*インターナショナルライズのパフォーマンスのチューニング*」を参照してください。

`<i18n:localize>`

| | |
|----------|---|
| タグライブラリ | i18n.tld |
| インポート文 | <code><%@ taglib uri="i18n.tld" prefix="i18n" %></code> |
| 実装されるクラス | LocalizeTag LocalizeExtraInfo |

このタグでは、`<i18n:getmessage>` タグを通じてリソースバンドルにアクセスするときにページ内で使われる言語コード、国別コード、ロケール変化形、および基本バンドル名を定義できます。

このタグでは、JSP ページに対して指定される文字コード化方式とコンテンツタイプも指定します。そのため、このタグは、バイトデータが正しくコード化されるように、できるだけページの始めの方（出力ストリームに書き込みを行う前）で使用しなければなりません。

HTML ページがより大きいページに組み込まれる場合には、`<i18n:localize>` タグは親ページ（組み込む側のページ）でしか使用できません。

`<i18n:localize>` タグでそのページのコード化方式が設定される一方で、コード化方式は応答の出力ストリームにバイト データを書き込む前に親ページで設定されなければならないことがその理由です。親ページでは、その中に組み込まれるページだけでなく、そのページ上のあらゆるコンテンツにも十分対応できるコード化方式を設定しなければなりません。

すべての文字列を `<i18n:getMessage>` タグで動的に取り出し、文字列を JSP ページに静的に埋め込むのを避ける（つまり、文字列をハードコード化するのを避ける）のが望ましいアプローチです。

ページに動的文字列（`<i18n:getMessage tag>` タグを使って取り出す文字列）だけが含まれている場合は、`<i18n:localize>` タグを、JSP 仕様で定義されている `<%@ page contentType="<something>" >` ページ ディレクティブと併用しないでください。`<i18n:localize>` タグを使う場合には同ディレクティブは不要であり、同ディレクティブを一緒に使うと、結果的に `contentType` 宣言が矛盾したり誤ったものになるおそれがあります。

同一ページ内に静的文字列と動的文字列を混在させる必要がある場合、同ページ ディレクティブを使う必要が出てきます。必ず、`<i18n:localize>` タグで指定される文字セットと、同ページ ディレクティブで指定される文字セットの互換性を保つようにしてください。

タグの属性

表 3-7 は、`<i18n:localize>` タグの属性についての説明です。

表 3-7 <i18n:localize> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------------|-----|-----------------------------|--|-----|
| bundleName | いいえ | String | MessageBundle の基本名は、JSP ページ用のローカライズされたテキストを取り出すために使われる。 リソース バンドル コンテンツが更新されているかどうかを WebLogic Portal がチェックする間隔設定についての詳細は、 http://edocs.beasys.co.jp/e-docs/wlp/docs70/admin/sysadmin.htm にある『 <i>管理者ガイド</i> 』の「 <i>インターナショナルライズのパフォーマンスのチューニング</i> 」を参照。 | R |
| language | いいえ | String または String [] | ユーザの使用言語を示す文字列(2文字のISO言語コード)、またはユーザの使用言語コードのリストを優先度の高い順に格納した String [] (String 配列)。 | R |
| country | いいえ | String | 2文字のISO国別コード。たとえばこのコードは、英語を話す英国在住のユーザではなく、英語を話す米国在住のユーザ向けにローカライズされたテキストを格納した MessageBundle を探すなどの目的に使われる。 | R |
| variant | いいえ | String | ロケールの変化形を表す文字列。この変化形は、言語コードと国別コードだけでは表しきれない詳細なロケールがローカライゼーションに必要な場合に使われる。 | R |
| locale | いいえ | java.util.Locale | 言語、国、およびロケール変化形を文字列として指定する代わりに、java.util.Locale オブジェクトを指定できる。これが指定された場合、その Locale の language、country、および variant の各フィールド値が有効になり、このタグの language、country、および variant の各属性に文字列で指定された値はすべて無効になる。 | R |

表 3-7 <i18n:localize> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|---|-----|
| charset | いいえ | String | このページで使用する文字コード化方式セットの名前。選択された言語のコード化方式を特定できない場合のデフォルトは「UTF-8」 ¹⁾ 。特定できる場合、選択された言語に適したコード化方式が使われる。 | R |
| contentType | いいえ | String | ページに含まれているコンテンツのタイプ (デフォルトは「text/html」) | R |

例 1

コードリスト 3-7 の例では、<i18n:localize> タグを使用して単一の使用言語を定義します。

コードリスト 3-7 <i18n:localize> タグの使用例 (その 1)

```
<%@ taglib uri="i18n.tld" prefix="i18n" %>
.
.
.
<%
// 使用言語を 1 つだけ定義
String language = "en";
%>

<i18n:localize language="<%=language%>"
bundleName="i18nExampleResourceBundle"/>
<html>
<body>
<i18n:getMessage messageName="greeting"/>
</body>
</html>
```

例 2

コードリスト 3-8 の例では、`<i18n:localize>` タグを使用して 2 つの使用言語（英語とスペイン語）を定義します。

コードリスト 3-8 `<i18n:localize>` タグの使用例 (その 2)

```
<%@ taglib uri="i18n.tld" prefix="i18n" %>
.
.
.
<%
// 使用言語を 2 つ定義する配列
// （優先順位は、英語、スペイン語の順）
String[] languages = new String[] { "en", "es" };
%>

<i18n:localize language="<%=languages%>"
bundleName="i18nExampleResourceBundle"/>
<html>
<body>
<i18n:getMessage messageName="greeting"/>
</body>
</html>
```

`<i18n:getMessage>`

| | |
|-----------------|---|
| タグライブラリ | i18n.tld |
| インポート文 | <code><%@ taglib uri="i18n.tld" prefix="i18n" %></code> |
| 実装されるクラス | GetMessageTag GetMessageExtraInfo |

このタグは、`<i18:localize>` タグと組み合わせて、ローカライズされた静的テキストやメッセージを `JspMessageBundle` から取り出すために使用します。

タグの属性

表 3-8 は、`<i18n:getMessage>` タグの属性についての説明です。

表 3-8 `<i18n:getMessage>` タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|------------|--|-----|
| id | いいえ | String | JSP ページ内のラベル (またはメッセージ) の値を格納する。 | C |
| messageName | はい | String | メッセージバンドルに対するキー。 | R |
| messageArgs | いいえ | Object [] | メッセージバンドルに渡される引数。引数が指定されない場合、(メッセージではなく) 静的テキストが返されるものと仮定される。 たとえば、引数として {"Wednesday", "78"} を渡し、「Today is Wednesday, and the temperature is 78 degrees Fahrenheit.」のようなメッセージを合成する目的でこの属性を使用する。 | R |
| bundleName | いいえ | String | <code><i18n:localize></code> タグで正しく初期化されていれば、特定のタグ呼び出しで異なったバンドルを使いたい場合を除いて、このタグ属性を指定する必要はない。 リソースバンドルコンテンツが更新されているかどうかを WebLogic Portal がチェックする間隔設定についての詳細は、 http://edocs.beasys.co.jp/e-docs/wlp/docs70/admin/sysadmin.htm にある『 <i>管理者ガイド</i> 』の「インターナショナルライズのパフォーマンスのチューニング」を参照。 | R |
| language | いいえ | String | <code><i18n:localize></code> タグで正しく初期化されていれば、特定のタグ呼び出しで異なった言語コードを使いたい場合を除いて、このタグ属性を指定する必要はない。 | R |
| country | いいえ | String | <code><i18n:localize></code> タグで正しく初期化されていれば、特定のタグ呼び出しで異なった国別コードを使いたい場合を除いて、このタグ属性を指定する必要はない。 | R |

表 3-8 <i18n:getMessage> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|---------|-----|------------------|---|-----|
| variant | いいえ | String | <i18n:localize> タグで正しく初期化されていれば、特定のタグ呼び出しで異なったロケール変化形を使いたい場合を除いて、このタグ属性を指定する必要はない。 | R |
| locale | いいえ | java.util.Locale | <i18n:localize> タグで正しく初期化されていれば、特定のタグ呼び出しで異なったロケール (言語コード、国別コード、およびロケール変化形) を使いたい場合を除いて、このタグ属性を指定する必要はない。 | R |

例 1

コード リスト 3-9 の内容を JSP に組み込むと、次の出力が生成されます。

```
Welcome To This Page! 14 out of 100 files have been saved.
```

コード リスト 3-9

```
<%@ taglib uri="i18n.tld" prefix="i18n" %>
.
.
.
<%
// 使用言語を 1 つだけ定義
String language = "en";

// メッセージ引数の作成
Object[] args = new Object[]
{
    new Integer(14),
    new Integer(100)
};
%>

<i18n:localize language="<%=language%>"
bundleName="i18nExampleResourceBundle"/>
<html>
<body>
```

```
<i18n:getMessage messageName="greeting" />
<i18n:getMessage messageName="message" messageArgs="<%=args%>" />
</body>
</html>
```

次のコードは、`i18nExampleResourceBundle.properties` という名前のプロパティ ファイル内のエントリを示します。

```
greeting=Welcome To This Page!
message={0} out of {1} files have been saved.
```

パーソナライゼーション タグ

`<pz:div>`、`<pz:contentSelector>`、および `<pz:contentQuery>` の各タグでは、それぞれ Advisor を使用してのユーザの分類、コンテンツの選択、およびコンテンツの取得を行います。

以降の表の「必須」列では、属性が必須と省略可能のどちらであるかを「はい」または「いいえ」で示します。「R/C」列では、「C」は属性がコンパイル時表現であることを、「R」は属性が要求時表現とコンパイル時表現のいずれかであることを示します。

この節では、以下の内容について説明します。

- `pz` タグと内部キャッシュ
- `<pz:contentQuery>`
- `<pz:contentSelector>`
- `<pz:div>`

pz タグと内部キャッシュ

コンテンツ検索用の `contextParams` は、検索単位のコンフィグレーション属性をサポートし、内部キャッシュを使用するかどうかの判断に使用できます。

`<cm:select>` タグおよび `<cm:selectById>` タグは、`contextParams` の設定をサポートしますが、`pz` タグにはこの機能はありません。`<pz:contentSelector>` タグで内部キャッシュの使用を制御するには、次の方法に従います。

```
<pz:content*> タグに次の内容を追加します。
contextParams="someName=someValue"
```

次のような実行時式の使用が推奨されます。

```
contextParams='<%= "aName=aValue bName=bValue cName=cValue" %>'
```

<pz:contentQuery>

| | |
|----------|--|
| タグライブラリ | pz.tld |
| インポート文 | <%@ taglib uri="pz.tld" prefix="pz" %> |
| 実装されるクラス | ContentQueryTag ContentQueryExtraInfo |

`<pz:contentQuery>` タグでは、コンテンツ マネージャ内のコンテンツを対象にコンテンツ属性検索を実行します。`useCache` 属性が `true` に設定されている場合、コンテンツ管理クエリの結果がキャッシュされます。このタグには開始タグしかなく、本体と終了タグはありません。このタグは、コンテンツクエリの実行結果としてコンテンツ マネージャから返された `Content` オブジェクトの配列を返します。

こうして返される `Content` オブジェクトに JSP 開発者がアクセスするために必要なパーソナライゼーション コンテンツ タグには、以下のものがあります。

オブジェクト配列イテレータ タグ。このタグを使用すると、配列内の `Content` オブジェクトに対して反復処理を行うことができます。`Object` の配列に対して反復処理を行うには、`<es:forEachInArray>` タグを使用します (詳細については、[3-98 ページの「<es:forEachInArray>」](#)を参照)。

- コンテンツ アクセス タグ。コンテンツ タグは、コンテンツ内のメタデータ属性にアクセスし、コンテンツを取り出して HTTP 応答出力ストリームに書き出します (詳細については、[3-5 ページの「コンテンツ管理」](#)を参照)。

タグとコンテンツ管理サービス プロバイダ インタフェースとのマッピング

以下の表は、タグが、コンテンツ管理サービス プロバイダ インタフェース (SPI) 内のメソッドにアクセスする方法を示したものです。この表では、呼び出しの順序についても示しています。

| EJB/ データ オブジェクト | SPI |
|-------------------------------------|---|
| 1. EJBAdvisor.getAdvice() | 4. DocumentProvider.findDocumentMetadata() |
| 2. ContentQueryAdvislet.getAdvice() | 5. DocumentMetadataDef.getID() |
| 3. DocumentManager.getContent() | 6. DocumentMetadataDef.getName() ... (その他の get* メソッド) |
| | 7. DocumentMetadataDef.getAttributeNames() |
| | 8. DocumentMetadataDef.getAttribute() |

タグの属性

[表 3-9](#) は、`<pz:contentQuery>` タグの属性についての説明です。

表 3-9 `<pz:contentQuery>` タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------|-----|--------------|---|-----|
| max | いいえ | String, long | 返されるコンテンツ項目の最大数を制限する。この属性を指定しない場合、見つかったコンテンツ項目がすべて返される。 | R |

表 3-9 <pz:contentQuery> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|--|-----|
| sortBy | いいえ | String | <p>コンテンツのソートに使用するドキュメント属性を列挙する。形式は SQL の <i>order by</i> 句の構文に従う。ソートの指定は、メタデータ属性名とキーワード ASC および DESC の列挙に限られる。</p> <p>例： sortBy="creationDate" sortBy="creationDate ASC, title DESC"</p> | R |
| query | はい | String | <p>コンテンツの検索に使われるコンテンツクエリ文字列。</p> <p>例： query="mimetype contains 'text' && author='Proulx'"</p> <p>クエリ構築の詳細については、 http://edocs.beasys.co.jp/e-docs/wlp/docs70/dev/conmngmt.htm の『開発者ガイド』の「コンテンツクエリの作成」を参照。</p> | R |
| contentHome | はい | String | <p>ContentManager EJB ホームの JNDI 名。この名前前の JNDI オブジェクトは、ContentManager インタフェースを実装するオブジェクトを返す create メソッドを実装しなければならない。</p> <p>詳細については、3-43 ページの「contentHome の値を指定する」を参照。</p> | R |
| id | はい | String | <p>見つかったコンテンツ オブジェクトを格納する配列変数の名前。コンテンツが見つからない場合、Content オブジェクトの空の配列 (null ではない) が変数に格納される。</p> | C |

表 3-9 <pz:contentQuery> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------------|-----|-----------------|---|-----|
| useCache | いいえ | String, Boolean | Content をキャッシュするかどうかを指定する。 この属性の取り得る値は、以下の 2 つのいずれか。 false (デフォルト値): ContentCache は使用されない。false (未指定) の場合、cacheId、cacheScope、および cacheTimeout の各属性の設定は無視される。 true: ContentCache が使われる。 | R |
| cacheId | いいえ | String | Content をキャッシュするために使われる識別子名。内部的には、キャッシュは Map として実装され、この属性はそのキーとなる。この属性を指定しない場合、タグの id 属性が使われる。 | R |
| cacheTimeout | いいえ | String, long | キャッシュされた Content の有効時間 (ミリ秒単位)。Content がキャッシュされてからこの時間が経過すると、キャッシュされた Content はクリアされ、再び取得されてからキャッシュに置かれる。 タイムアウトしない (常にキャッシュされた Content を使用する) 場合には -1 を指定する。デフォルトは -1。 | R |

表 3-9 <pz:contentQuery> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------------|-----|--------|---|-----|
| cacheScope | いいえ | String | <p>コンテンツ キャッシュのライフサイクル スコープを指定する。<jsp:useBean>と同様。</p> <p>この変数を取り得る値:</p> <ul style="list-style-type: none"> ◆ application。Web アプリケーション内にある、顧客が要求するすべての JSP がキャッシュにアクセスできる。 ◆ session (デフォルト)。Web アプリケーション内にある、現在の顧客が要求するすべての JSP がキャッシュにアクセスできる。 ◆ page。顧客が要求する現在の JSP だけがキャッシュにアクセスできる。 ◆ request。現在の顧客リクエストだけがキャッシュにアクセスできる。顧客が再びページを要求した場合、コンテンツ セレクタはクエリを再実行し、キャッシュを作成し直す。 | R |

例

コード リスト 3-10 は、<pz:contentQuery> タグの使用方法の例です。

コード リスト 3-10 <pz:contentQuery> タグの使用例

```
<%@ page import="bea.pl3n.content.ContentHelper" %>
<%@ taglib uri="es.tld" prefix="es" %>
<%@ taglib uri="cm.tld" prefix="cm" %>
<%@ taglib uri="pz.tld" prefix="pz" %>
.
.
.
<pz:contentQuery id="docs"
```

```
contentHome="<%=ContentHelper.DEF_DOCUMENT_MANAGER_HOME%>"
query="author = 'Hemingway'" />

<ul>

<es:forEachInArray array="<%=docs%>" id="aDoc"
type="com.bea.pl3n.content.Content">
  <li>The document title is: <cm:printProperty id="aDoc"
name="Title" encode="html" />
</es:forEachInArray>
</ul>
```

<pz:contentSelector>

| | |
|-----------------|--|
| タグライブラリ | pz.tld |
| インポート文 | <%@ taglib uri="pz.tld" prefix="pz" %> |
| 実装されるクラス | ContentSelectorTag ContentSelectorExtraInfo |

<pz:contentSelector> タグを使用すると、コンテンツ セレクタ ルールに基づいて、パーソナライズされた任意のコンテンツを推奨できるようになります。

コンテンツ セレクタ ルールはまず、E-Business Control Center で定義された 1 セットの条件 (たとえば、あるユーザが指定した分類に当てはまるかどうか) を評価します。条件が true と評価されると、E-Business Control Center で定義されたコンテンツ クエリに基づいて、コンテンツが Content Manager から取り出されます。

注意： ルールは、E-Business Control Center を使って作成します。この GUI ツールは、ビジネス アナリスト (BA) が独自の顧客セグメントを開発できるように設計されています。ビジネス アナリストはルールの概念に詳しくないため、コンテンツ セレクタ ルールのことを単に「コンテンツ セレクタ 」、また分類子ルールを「顧客セグメント」と呼ぶことがあります。

コンテンツ セレクタ ルールの結果をキャッシュするには、`useCache` 属性を `true` に設定します。キャッシュがタイムアウトしていなければ、`<pz:contentSelector>` タグへの後続の呼び出しは、そのコンテンツ クエリを再評価することなくキャッシュされた値を返します。

`<pz:contentSelector>` タグには開始タグしかなく、本体と終了タグはありません。このタグは、コンテンツ クエリの実行結果として Content Manager から返された Content オブジェクトの配列を返します。

返される Content オブジェクトに JSP 開発者がアクセスするために必要なタグには、以下のものがあります。

- オブジェクト配列イテレータ タグ。このタグを使用すると、配列内の Content オブジェクトに対して反復処理を行うことができます。Object の配列に対して反復処理を行うには、`<es:forEachInArray>` タグを使用します。
- コンテンツ アクセス タグ。コンテンツ タグは、コンテンツ内のメタデータ 属性にアクセスし、コンテンツを取り出して HTTP 応答出力ストリームに書き出します。

タグとコンテンツ管理サービス プロバイダ インタフェースとのマッピング

以下の表は、タグが、コンテンツ管理サービス プロバイダ インタフェース (SPI) 内のメソッドにアクセスする方法を示したものです。この表では、呼び出しの順序についても示しています。

| EJB/ データ オブジェクト | SPI |
|--|--|
| 1. <code>EJBAdvisor.getAdvice()</code> | 4. <code>DocumentProvider.findDocumentMetadata()</code> |
| 2. <code>ContentQueryAdvislet.getAdvice()</code> | 5. <code>DocumentMetadataDef.getID()</code> |
| 3. <code>DocumentManager.getContent()</code> | 6. <code>DocumentMetadataDef.getName() ... (その他の get* メソッド)</code> |
| | 7. <code>DocumentMetadataDef.getAttributeNames()</code> |
| | 8. <code>DocumentMetadataDef.getAttribute()</code> |

タグの属性

表 3-10 は、<pz:contentSelector> タグの属性についての説明です。

表 3-10 <pz:contentSelector> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------------|---|-----|
| rule | はい | String | E-Business Control Center で作成されたコンテンツセレクタ定義内でのコンテンツセレクタの名前。 | R |
| contentHome | はい | String | ContentManager EJB ホームの JNDI 名。この名前の JNDI オブジェクトは、ContentManager インタフェースを実装するオブジェクトを返す create メソッドを実装しなければならない。詳細については、 3-43 ページの「contentHome の値を指定する」 を参照。 | R |
| max | いいえ | String, long | 返されるコンテンツ項目の最大数を制限する。未指定または -1 に設定された場合、見つかったコンテンツ項目がすべて返される。 | R |
| sortBy | いいえ | String | コンテンツのソートに使用するドキュメント属性を列挙する。形式は SQL の <i>order by</i> 句の構文に従う。ソートの指定は、メタデータ属性名とキーワード ASC および DESC の列挙に限られる。 例： sortBy="creationDate" sortBy="creationDate ASC, title DESC" | R |

表 3-10 <pz:contentSelector> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|----------|-----|-----------------|---|-----|
| query | いいえ | String | <p>コンテンツの検索に使われるコンテンツクエリ文字列。このクエリは、ビジネスアナリストが E-Business Control Center で作成したどのクエリよりも優先される。</p> <p>例 : query="mimetype contains 'text' && author='Salinger'"</p> <p>クエリ構築の詳細については、http://edocs.beasys.co.jp/e-docs/wlp/docs70/dev/conmgmt.htm の『開発者ガイド』の「コンテンツクエリの作成」を参照。</p> | R |
| id | はい | String | <p>見つかったコンテンツ オブジェクトを格納する配列変数の名前。コンテンツが見つからない場合、Content オブジェクトの空の配列 (null ではない) が変数に格納される。</p> | C |
| useCache | いいえ | String, Boolean | <p>Content をキャッシュするかどうかを指定する。</p> <p>この属性の取り得る値は、以下の 2 つのいずれか。</p> <p>false (デフォルト値) : ContentCache は使用されない。false (未指定) の場合、cacheId、cacheScope、および cacheTimeout の各属性の設定は無視される。</p> <p>true: ContentCache が使われる。</p> | R |
| cacheId | いいえ | String | <p>Content をキャッシュするために使われる識別子名。内部的には、キャッシュは Map として実装され、この属性はそのキーとなる。この属性を指定しない場合、タグの id 属性が使われる。</p> | R |

表 3-10 <pz:contentSelector> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------------|-----|--------------|--|-----|
| cacheTimeout | いいえ | String, long | <p>キャッシュされた Content の有効時間 (ミリ秒単位)。Content がキャッシュされてからこの時間が経過すると、キャッシュされた Content はクリアされ、再び取得されてからキャッシュに置かれる。</p> <p>タイムアウトしない (常にキャッシュされた Content を使用する) 場合には -1 を指定する。デフォルトは -1。</p> | R |
| cacheScope | いいえ | String | <p>コンテンツ キャッシュのライフサイクル スコープを指定する。 <jsp:useBean> と同様。 この変数が取り得る値：</p> <ul style="list-style-type: none"> ◆ application。Web アプリケーション内にある、顧客が要求するすべての JSP ページがキャッシュにアクセスできる。 ◆ session (デフォルト)。Web アプリケーション内にある、現在の顧客が要求するすべての JSP ページがキャッシュにアクセスできる。 ◆ page。顧客が要求する現在の JSP だけがキャッシュにアクセスできる。 ◆ request。現在の顧客リクエストだけがキャッシュにアクセスできる。顧客が再びページを要求した場合、コンテンツ セレクタはクエリを再実行し、キャッシュを作成し直す。 | R |

contentHome の値を指定する

コンテンツ セレクタ タグでは、contentHome を使って、コンテンツ管理システムの JNDI ホームを指定する必要があります。参考版またはサードパーティ製のコンテンツ管理システムを使用している場合、スクリプトレットを使ってデフォルトのコンテンツ ホームを参照することができます。スクリプトレットは ContentHelper クラスを使うので、先に、以下のタグを使ってクラスを JSP にインポートしておく必要があります。

```
<%@ page import="com.bea.pl3n.content.ContentHelper"%>
```

次に、コンテンツ セレクタ タグを使うときに、以下のようにして contentHome を指定します。

```
<pz:contentSelector
contentHome="<%=ContentHelper.DEF_DOCUMENT_MANAGER_HOME %%"
... />
```

独自のコンテンツ管理システムを作成する場合、ContentHelper スクリプトレットを使うのではなく、JNDI ホームを指定する必要があります。また、独自のコンテンツ管理システムで JNDI ホームが指定されている場合は、ContentHelper スクリプトレットを使う代わりに、その JNDI ホームを指定することができます。

例

[コード リスト 3-10](#) は、<pz:contentSelector> タグの使用法の例です。

コード リスト 3-11 <pz:contentSelector> タグの使用例

```
<%@ page import="com.bea.pl3n.content.ContentHelper" %>
<%@ taglib uri="es.tld" prefix="es" %>
<%@ taglib uri="cm.tld" prefix="cm" %>
<%@ taglib uri="pz.tld" prefix="pz" %>
<%@ taglib uri="um.tld" prefix="um" %>
.
.
.
<um:getProfile profileKey="bob"
profileId="myProfile" scope="session"/>
<pz:contentSelector rule="PremierCustomerSpotlight"
contentHome="<%=ContentHelper.DEF_DOCUMENT_MANAGER_HOME %%"
id="docs" />
<ul>
  <es:forEachInArray array="<%=docs%%" id="aDoc"
```

```
type="com.bea.pl3n.content.Content">
  <li>The document title is: <cm:printproperty id="aDoc"
    name="Title" encode="html" />
</es:forEachInArray>
</ul>
```

注意： max 属性と組み合わせて使われるときの sortBy 属性の作用は、明示的な属性（システム定義）と暗黙的な属性（ユーザ定義）の場合で異なります。明示的な属性（identifier、mimeType、size、version、author、creationDate、modifiedBy、modifiedDate、lockedBy、description、または comments）を基準にしてソートする場合、そのソートはデータベース上で行われます。したがって、max="10" と sortBy を組み合わせると、システムはソートを行ってから最初の 10 項目を取得します。暗黙的な属性を基準にしてソートする場合、max 属性で指定した数の項目が選択された後でソートが行われます。

<pz:div>

| | |
|-----------------|--|
| タグライブラリ | pz.tld |
| インポート文 | <%@ taglib uri="pz.tld" prefix="pz" %> |
| 実装されるクラス | DivTag DivTagExtraInfo |

<pz:div> タグでは、ルール エンジンによる分類子ルールの実行結果として、条件に従ってコンテンツの断片を取り込むことができます。ユーザのプロファイルが E-Business Control Center で指定された分類に一致する場合、条件に合ったコンテンツが取り込まれます。このタグには、開始タグ、本体、および終了タグがあります。このタグは、ユーザが属する Classification オブジェクトのリストを返します。

タグの属性

表 3-11 は、<pz:div> タグの属性についての説明です。

表 3-11 <pz:div> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------|-----|--------|--|-----|
| rule | はい | String | E-Business Control Center で作成された顧客セグメント定義内での分類子ルールの名前。 | R |
| id | いいえ | String | 特定の分類子ルールに関してユーザに適用される Classification オブジェクトを格納するコレクション。 | C |

例

```

<%@ taglib uri="pz.tld" prefix="pz" %>
<%@ taglib uri="um.tld" prefix="um" %>

<um:getProfile profileKey="bob"
profileId="myProfile" scope="session"/>
<pz:div id="classifications" rule="PremierCustomer">

<%
// ユーザが「主要顧客」に分類される場合、1 つのエントリがあるリストが返される
//
    java.util.Iterator iterator=classifications.iterator();
    while (iterator.hasNext())
    {
        com.bea.pl3n.user. Classification
classification=(Classification) iterator.next();
        out.println (classification.getName());
    }
%>

    <p> 新しいお得意様優待プログラムを開始しました。 <p>
</pz:div>

```

プレースホルダ

プレースホルダ タグは、JSP 上で名前付きの位置として機能します。プレースホルダの動作は、E-Business Control Center を使って定義します。

以降の表の「必須」列では、属性が必須と省略可能のどちらであるかを「はい」または「いいえ」で示します。「R/C」列では、「C」は属性がコンパイル時表現であることを、「R」は属性が要求時表現とコンパイル時表現のいずれかであることを示します。

<ph:placeholder>

| | |
|----------|---|
| タグライブラリ | ph.tld |
| インポート文 | <%@ taglib uri="ph.tld" prefix="ph" %>> |
| 実装されるクラス | PlaceholderTag |

<ph:placeholder> タグは、プレースホルダを実装するタグです。プレースホルダは、JSP ページ上の特定の位置における動作を定義します。プレースホルダは、E-Business Control Center を使って定義します。

複数のプレースホルダ タグが同一のプレースホルダを参照していてもかまいません。プレースホルダ タグは、それぞれのプレースホルダ定義を、インスタンスごとに個別に呼び出します。プレースホルダ定義が複数のクエリを指定している場合、プレースホルダ タグの各インスタンスの定義が同一の場合でも、インスタンスごとに別々の広告を表示できます。

広告プレースホルダを持つ JSP のリクエストを WebLogic Portal が受信すると、プレースホルダ タグは広告サービス（表示する広告を決定するビジネス ロジックを呼び出すセッション EJB）に問い合わせます。

関連するタグの詳細については、「<ad:adTarget>」を参照してください。

タグとコンテンツ管理サービス プロバイダ インタフェースとのマッピング

以下の表は、タグが、コンテンツ管理サービス プロバイダ インタフェース (SPI) 内のメソッドにアクセスする方法を示したものです。この表では、呼び出しの順序についても示しています。

| EJB/ データ オブジェクト | SPI |
|---|---|
| 1. PlaceholderService.getContent() | 5. DocumentProvider.findDocumentMetadata() |
| 2. AdBucketService.getContent() | 6. DocumentMetadataDef.getID() |
| 3. AdService.getContent() | 7. DocumentMetadataDef.getName() ... (その他の get* メソッド) |
| 4. DocumentManager.getContent() | 8. DocumentMetadataDef.getAttributeNames() |
| 10. Document.getProperty() ... (その他の get* メソッド) | 9. DocumentMetadataDef.getAttribute() |
| 11. Document.getContent() | 13. DocumentProvider.getDocument() |
| 12. DocumentManager.getContentBlock() | 14. DocumentDef.openStream() |

タグの属性

表 3-12 は、<ph:placeholder> タグの属性についての説明です。

表 3-12 <ph:placeholder> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------|----|--------|--------------------|-----|
| name | はい | String | ブレースホルダ定義を参照する文字列。 | R |

表 3-12 <ph:placeholder> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------|-----|------|---|-----|
| height | いいえ | int | <p>ブラウザでドキュメントを表示するために必要な HTML を生成するときには、プレースホルダが占める高さをピクセル単位で指定する。</p> <p>プレースホルダでは、表示サイズが適用されるコンテンツタイプだけを対象に、他の属性によってそのドキュメントに対して定義されたサイズがまだない場合にのみこの値が使用される。</p> <p>この値を指定せず、他の属性によって定義されてもいない場合、ドキュメントの高さはブラウザの動作によって決定される。</p> | R |
| width | いいえ | int | <p>ブラウザでドキュメントを表示するために必要な HTML を生成するときには、プレースホルダが占める幅をピクセル単位で指定する。</p> <p>プレースホルダでは、表示サイズが適用されるコンテンツタイプだけを対象に、他の属性によってそのドキュメントに対して定義されたサイズがまだない場合にのみこの値が使用される。</p> <p>この値を指定せず、他の属性によって定義されてもいない場合、ドキュメントの幅はブラウザの動作によって決定される。</p> | R |

例

コードリスト 3-12 は、MainPageBanner プレースホルダで指定された広告を表示する例です。

コードリスト 3-12 <ph:placeholder> タグの使用例

```
<%@ taglib uri="ph.tld" prefix="ph" %>
.
.
```

```
<ph:placeholder name="/placeholders/MainPageBanner.pla"/>
```

プロパティ セット

プロパティ セット タグを使用すると、利用可能なプロパティとプロパティ セットのリストにアクセスできるようになります。プロパティ セットの操作は、E-Business Control Center を使用して行います。

すべてのプロパティ セットタグは、結果を同じファイルに送ります。結果を確認する場合、次のインポート ディレクティブをページの先頭に配置します。

```
<%@ page
import="com.bea.pl3n.property.servlets.jsp.taglib.PropertySetTag-
Constants" %>
```

以降の表の「必須」列では、属性が必須と省略可能のどちらであるかを「はい」または「いいえ」で示します。「R/C」列では、「C」は属性がコンパイル時表現であることを、「R」は属性が要求時表現とコンパイル時表現のいずれかであることを示します。

<ps:getPropertyNames>

| | |
|----------|--|
| タグ ライブラリ | ps.tld |
| インポート文 | <%@ taglib uri="ps.tld" prefix="ps" %> |
| 実装されるクラス | GetPropertyNamesTag GetPropertyNamesExtraInfo |

<ps:getPropertyNames> タグは、指定されたプロパティ セット内のプロパティ名のリストを取得するために使用します。

タグの属性

表 3-13 は、`<ps:getPropertyNames>` タグの属性についての説明です。

表 3-13 `<ps:getPropertyNames>`

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-----------------|-----|--------|--|-----|
| propertySetName | はい | String | 新しい検索の対象となるプロパティ セットの名前。 | R |
| propertySetType | はい | String | 検索するプロパティ セットのタイプ。 | R |
| id | はい | String | プロパティ名のリストを格納する変数 (String 配列) の識別子 (id)。 | C |
| result | いいえ | String | <p>操作の結果に応じて作成および初期化される Integer 型変数の識別子。</p> <p>この変数を取り得る値：</p> <p>クエリが正常に実行された場合： PropertySetTagConstants.PROPERTY_SEARCH_OK</p> <p>プロパティ名のリストを取得する際に問題があった場合： PropertySetTagConstants.PROPERTY_SEARCH_FAILED</p> <p>propertySetName と propertySetType で指定されたプロパティ セットが見つからなかった場合： PropertySetTagConstants.INVALID_PROPERTY_SET</p> | C |

例

コード リスト 3-13 は、`<ps:getPropertyNames>` タグの使用方法の例です。

コード リスト 3-13 `<ps:getPropertyNames>` タグの使用例

```
<%@ taglib uri="ps.tld" prefix="ps" %>
```

```

<%@ page import=
"com.bea.pl3n.property.servlets.jsp.taglib.PropertySetTagConstant
s"
%>

<% String myPropertySet="Demographics"; %>

<p> ----- <b>ps:getPropertyNames</b> -----

<br>

<ps:getPropertyNames propertySetName="<%= myPropertySet %%"
propertySetType="USER" id="propertyNames" result="myResult"/>

<% for (int i=0; i<propertyNames.length; i++)
out.println(propertyNames[i] + "<br>");
%>

```

<ps:getPropertySetNames>

| | |
|----------|--|
| タグライブラリ | ps.tld |
| インポート文 | <%@ taglib uri="ps.tld" prefix="ps" %> |
| 実装されるクラス | GetPropertySetNamesTag GetPropertySetNamesExtraInfo |

<ps:getPropertySetNames> タグ (表 3-14) は、指定されたプロパティ セットタイプのプロパティ セットのリストを取得するために使用します。

タグの属性

表 3-14 は、<ps:getPropertySetNames> タグの属性についての説明です。

表 3-14 <ps:getPropertySetNames>

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-----------------|----|--------|--------------------|-----|
| propertySetType | はい | String | 検索するプロパティ セットのタイプ。 | R |

表 3-14 <ps:getPropertySetNames> (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------|-----|--------|---|-----|
| id | はい | String | プロパティ名のリストを格納する変数 (String 配列) の識別子。 | C |
| result | いいえ | String | 操作の結果に応じて作成および初期化される Integer 型変数の識別子。 この変数を取り得る値: クエリが正常に実行された場合: PropertySetTagConstants.PROPERTY_SET_SEARCH_OK プロパティ名のリストを取得する際に問題があった場合: PropertySetTagConstants.PROPERTY_SET_SEARCH_FAILED propertySetName と propertySetType で指定されたプロパティセットが見つからなかった場合: PropertySetTagConstants.INVALID_PROPERTY_SET | C |

例

コード リスト 3-14 は、<ps:getPropertySetNames> タグの使用方法の例です。

コード リスト 3-14 <ps:getPropertySetNames> タグの使用例

```

<%@ taglib uri="ps.tld" prefix="ps" %>
.
.
.
<ps:getPropertySetNames propertySetType="USER"
id="userPropertySets" result="myResult" />
    
```

<ps:getRestrictedPropertyValues>

| | |
|----------|--|
| タグライブラリ | ps.tld |
| インポート文 | <%@ taglib uri="ps.tld" prefix="ps" %> |
| 実装されるクラス | GetRestrictedPropertyValuesTag GetRestrictedPropertyValuesExtraInfo |

<ps:getRestrictedPropertyValues> タグは、特定のプロパティ定義を対象に、文字列に変換された制限付き値のリストを返します。値は文字列 (String) の配列として返されます。

タグの属性

表 3-15 は、<ps:getRestrictedPropertyValues> タグの属性についての説明です。

表 3-15 <ps:getRestrictedPropertyValues>

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-----------------|----|--------|-------------------------------------|-----|
| propertySetName | はい | String | 対象のプロパティが含まれるプロパティセットの名前。 | R |
| propertySetType | はい | String | 対象のプロパティが含まれるプロパティセットのタイプ。 | R |
| propertyName | はい | String | 値を取得する対象のプロパティの名前。 | R |
| id | はい | String | プロパティ名のリストを格納する変数 (String 配列) の識別子。 | C |

表 3-15 <ps:getRestrictedPropertyValues> (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------|-----|--------|--|-----|
| result | いいえ | String | <p>操作の結果に応じて作成および初期化される Integer 型変数の識別子。</p> <p>この変数を取り得る値:</p> <p>クエリが正常に実行された場合: PropertySetTagConstants. PROPERTY_SEARCH_OK</p> <p>プロパティへのアクセスに問題がある場合: PropertySetTagConstants. PROPERTY_SEARCH_FAILED</p> <p>propertySetName と propertySetType で指定されたプロパティセットが見つからなかった場合: PropertySetTagConstants. INVALID_PROPERTY_SET</p> <p>指定されたプロパティが制限付きでない場合: PropertySetTagConstants. PROPERTY_NOT_RESTRICTED</p> | C |

例

コードリスト 3-15 は、<ps:getRestrictedPropertyValues> タグの使用方法の例です。

コードリスト 3-15 <ps:getRestrictedPropertyValues> タグの使用例

```
<%@ taglib uri="ps.tld" prefix="ps" %>
<%@ page import="com.bea.pl3n.property.servlets.jsp.taglib.PropertySetConstants" %>
<p> ---- <b>ps:getRestrictedPropertyValues</b> ----
<br>Possible values for PreferredLanguage:
```

```
<ps:getRestrictedPropertyValues propertySetName="Demographics"
  propertySetType="USER" propertyName="PreferredLanguage"
  id="values" result="myResult"/>

<ul>
<% if (myResult.intValue() ==
PropertySetTagConstants.PROPERTY_SEARCH_OK)
  {
      for ( int i=0; i<values.length; i++ ) {
          %><li><%=values[i]%>
<% }
  }
%>
</ul>
```

ユーザ管理： プロフィール管理タグ

ユーザ管理タグを使用すると、ユーザとグループのプロファイル情報にアクセスできるだけでなく、ユーザとグループの作成および削除や、ユーザとグループの関係の管理などの操作を行うこともできます。

以降の表の「必須」列では、属性が必須と省略可能のどちらであるかを「はい」または「いいえ」で示します。「R/C」列では、「C」は属性がコンパイル時表現であることを、「R」は属性が要求時表現とコンパイル時表現のいずれかであることを示します。

<um:getProfile>

| | |
|---------|--------|
| タグライブラリ | um.tld |
|---------|--------|

| | |
|-----------------|---|
| インポート文 | <pre><%@ taglib uri="um.tld" prefix="um" %></pre> <p>すべてのユーザ管理タグは、同じファイルに結果を送る。結果を確認する場合、次のインポート ディレクティブをページの先頭に配置する。</p> <pre><%@ page import="com.bea.pl3n.usermgmt. servlets.jsp.taglib.UserManagementTag- Constants" %></pre> |
| 実装されるクラス | <pre>GetProfileTag GetProfileExtraInfo</pre> |

`<um:getProfile>` タグ (表 3-16) は、指定されたプロファイル キーとプロファイル タイプに一致するプロファイルを検索します。このタグには本体はありません。取得されたプロファイルは

`com.bea.pl3n.usermgmt.profile.ProfileWrapper` として扱うことができます。プロファイル キーとプロファイルに加えて、`profileType` 属性で指定するものと同様の、明示的なサクセサ キーとサクセサ タイプを指定することができます。このサクセサは、取り出されたプロファイルとともに、その後で行われる `<um:getProperty>` タグの呼び出しで使われ、そのサクセサから必ずプロパティが継承されるようになります。サクセサが取り出されなければ、標準の `ConfigurableEntity` サクセサ検索パターンが、取り出されたプロパティに適用されることとなります。

タグの属性

表 3-16 は、`<um:getProfile>` タグの属性についての説明です。

表 3-16 `<um:getProfile>` タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|---------------------------|-----|--------|---|-----|
| <code>profileKey</code> | はい | String | 目的のプロファイルを検索するために使用できる固有の識別子。 例：“ <code><%=username%></code> ” | R |
| <code>successorKey</code> | いいえ | String | プロファイル サクセサを検索するために使用できる固有の識別子。 例：“ <code><%=defaultGroup%></code> ” | R |

表 3-16 <um:getProfile> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|--|-----|
| scope | いいえ | String | 取り出されたプロフィールの HTTP スコープ。「request」または「session」を値として指定する。 デフォルトは <i>session</i> 。 | C |
| groupOnly | いいえ | String | ユーザプロフィールではなく、 <i>profileKey</i> で指定された名前のグループプロフィールを検索するように指定する。この値が <i>true</i> の場合には、サクセサは検索されない。 デフォルトは <i>false</i> 。 | C |
| profileId | いいえ | String | JSP のページ スコープ内で、取り出されたプロフィールへのアクセスに使用できる変数名。 | C |
| successorId | いいえ | String | JSP のページ スコープ内で、取り出されたサクセサへのアクセスに使用できる変数名。 | C |
| result | いいえ | String | 操作の結果にアクセスするために使用できる変数名。 この変数が取り得る値： <i>操作が正常に実行された場合：</i> <code>UserManagementTagConstants.GET_PROFILE_OK</code> <i>エラーが発生した場合：</i> <code>UserManagementTagConstants.GET_PROFILE_FAILED</code> <code>UserManagementTagConstants.NO_SUCH_PROFILE</code> <code>UserManagementTagConstants.NO_SUCH_SUCCESSOR</code> | C |

例 1

コードリスト 3-16 は、サクセサを指定せず、*session* スコープを明示的に指定してプロフィールを取り出す方法の例です。

コードリスト 3-16 <um:getProfile> タグの使用例 (その1)

```
<%@ taglib uri="um.tld" prefix="um" %>
.
.
.
<um:getProfile profileKey="bob" profileType="AcmeUser"
profileId="myProfile" scope="session"/>
```

例 2

コードリスト 3-17 は、デフォルトのサクセサ タイプを使用し、*request* スコープを明示的に指定してデフォルトのユーザ プロファイル タイプを取り出す方法の例です。

コードリスト 3-17 <um:getProfile> タグの使用例 (その2)

```
<%@ taglib uri="um.tld" prefix="um" %>
.
.
.
<um:getProfile profileKey="bob" successorKey="engineering"
scope="request"/>
```

<um:getProperty>

| | |
|---------|--|
| タグライブラリ | um.tld |
| インポート文 | <pre><%@ taglib uri="um.tld" prefix="um" %> すべてのユーザ管理タグは、同じファイルに結果を送る。結果を確認する場合、次のインポートディレクトィブをページの先頭に配置する。 <%@ page import="com.bea.pl3n.usermgmt. servlets.jsp.taglib.UserManagementTag- Constants" %></pre> |

| | |
|----------|--|
| 実装されるクラス | GetPropertyTag GetPropertyExtraInfo |
|----------|--|

<um:getProperty> タグは、指定されたプロパティ セットとプロパティ名の組に対応するプロパティ値を検索します。このタグには本体はありません。返される値は Object です。<um:getProfile> タグが呼び出されて、セッションで使うためのプロフィールが取り出された後でこのタグが使われるのが典型的なケースです。検索対象のプロパティは、そのセッション プロフィールから取り出されません。<um:getProperty> タグの呼び出し時に <um:getProfile> タグがまだ使われていない場合、指定されたプロパティ値は匿名ユーザ プロフィールから取り出されます。

タグの属性

表 3-17 は、<um:getProperty> タグの属性についての説明です。

表 3-17 <um:getProperty> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------------|-----|--------|---|-----|
| propertySet | いいえ | String | プロパティの値を取得する対象のプロパティ セット。 例：“Demographics” 注意： プロパティ セットが指定されない場合、プロパティはプロフィールのデフォルト（スコープ制限なし）プロパティから取り出される。 | R |
| propertyName | はい | String | 取得するプロパティの名前。 例：“Date_of_Birth” | R |
| id | いいえ | String | id 属性を指定すると、id が割り当てられている変数名の中で、取得したプロパティの値を使用できる。この属性を指定しない場合、プロパティ値はインライン化される。 | C |

例 1

コードリスト 3-18 は、`<um:getProperty>` タグの使用方法の例です。

コードリスト 3-18 `<um:getProperty>` タグの使用例

```
<%@ taglib uri="um.tld" prefix="um" %>
.
.
.
<um:getProperty id="myBirthDate" propertySet="Demographics"
propertyName="Date_of_Birth"/>
My birthday is <%=myBirthDate%>.
```

`<um:getPropertyAsString>`

| | |
|-----------------|--|
| タグライブラリ | um.tld |
| インポート文 | <pre><%@ taglib uri="um.tld" prefix="um" %> すべてのユーザ管理タグは、同じファイルに結果を送る。結果を確認する場合、次のインポートディレクティブをページの先頭に配置する。 <%@ page import="com.bea.pl3n.usermgmt. servlets.jsp.taglib.UserManagementTag- Constants" %></pre> |
| 実装されるクラス | GetPropertyAsStringTag GetPropertyAsStringExtraInfo |

`<um:getPropertyAsString>` タグは、前に説明した `<um:getProperty>` タグと同じように機能しますが、取り出されるプロパティ値は必ず `String` になります。次の例では、`Collection` を返し、好きな色を列挙する多値プロパティを示しています。

タグの属性

表 3-18 は、`<um:getPropertyAsString>` タグの属性についての説明です。

表 3-18 `<um:getPropertyAsString>` タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------------|-----|--------|--|-----|
| propertySet | いいえ | String | プロパティの値を取得する対象のプロパティセット。 例：“Demographics” 注意： プロパティ セットが指定されない場合、プロパティはプロフィールのデフォルト（スコープ制限なし）プロパティから取り出される。 | R |
| propertyName | はい | String | 取得するプロパティの名前。 例：“Date_of_Birth” | R |
| id | いいえ | String | id 属性を指定すると、id が割り当てられている変数名の中で、取得したプロパティの値を使用できる。この属性を指定しない場合、プロパティ値はインライン化される。 | C |

例

コード リスト 3-19 は、`<um:getPropertyAsString>` タグの使用方法の例です。

コード リスト 3-19 `<um:getPropertyAsString>` タグの使用例

```
<%@ taglib uri="um.tld" prefix="um" %>
.
.
.
<um:getPropertyAsString id="myBirthDate"
propertySet="Demographics" propertyName="Date_of_Birth"/>
My birthday is <%=myBirthDate%>.
```

<um:removeProperty>

| | |
|-----------------|---|
| タグライブラリ | um.tld |
| インポート文 | <pre><%@ taglib uri="um.tld" prefix="um" %></pre> <p>すべてのユーザ管理タグは、同じファイルに結果を送る。結果を確認する場合、次のインポートディレクティブをページの先頭に配置する。</p> <pre><%@ page import="com.bea.p13n.usermgmt.servlets.jsp.taglib.UserManagementTagConstants" %></pre> |
| 実装されるクラス | RemovePropertyTag |

<um:removeProperty> タグ (表 3-19) は、指定されたプロパティを現在のセッションのプロファイルまたは匿名ユーザ プロファイルから削除します。このタグには本体はありません。削除されたプロパティに対して引き続き <um:getProperty> を呼び出すと、プロパティ セットで規定されているデフォルトのプロパティ値か、またはプロファイルのサクセサから継承されるデフォルト値が得られます。

タグの属性

表 3-19 は、<um:removeProperty> タグの属性についての説明です。

表 3-19 <um:removeProperty> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|--|-----|
| propertySet | いいえ | String | <p>プロパティの値を削除する対象のプロパティセット。</p> <p>例：“Demographics”</p> <p>注意： プロパティ セットが指定されない場合、プロパティはプロファイルのデフォルト（スコープ制限なし）プロパティから削除される。</p> | R |

表 3-19 <um:removeProperty> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------------|----|--------|-----------------------------------|-----|
| propertyName | はい | String | 削除するプロパティの名前。 例：“Income_Range” | R |

例

[コードリスト 3-20](#) は、<um:removeProperty> タグを使用してプロパティを削除する方法を示しています。

コード リスト 3-20 <um:removeProperty> タグの使用例

```
<%@ taglib uri="um.tld" prefix="um" %>
.
.
.
<um:removeProperty propertySet="<%=thePropertySet%>"
propertyName="<%=thePropertyName%>" />
```

<um:setProperty>

| | |
|----------|---|
| タグライブラリ | um.tld |
| インポート文 | <pre><%@ taglib uri="um.tld" prefix="um" %> すべてのユーザ管理タグは、同じファイルに結果を送る。結果を確認する場合、次のインポート ディレクティブをページの先頭に配置する。 <%@ page import="com.bea.pl3n.usermgmt. servlets.jsp.taglib.UserManagementTag- Constants" %></pre> |
| 実装されるクラス | <pre>SetPropertyTag ResultTagExtraInfo</pre> |

<um:setProperty> タグは、セッションの現在のプロファイルまたは匿名ユーザプロファイルのプロパティ値を更新します。このタグには本体はありません。

タグの属性

表 3-20 は、<um:setProperty> タグの属性についての説明です。

表 3-20 <um:setProperty> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------------|-----|--------|--|-----|
| propertySet | いいえ | String | プロパティ値を設定する対象のプロパティセット。 例：“Demographics” 注意： プロパティセットを指定しない場合、プロパティ値はプロファイルのデフォルト（スコープ制限なし）プロパティに対して設定される。 | R |
| propertyName | はい | String | 値を設定するプロパティの名前。 例：“Gender” | R |
| value | はい | Object | 新しいプロパティ値。 | R |
| result | いいえ | String | プロパティ設定操作の結果を格納する Integer 型オブジェクトの名前。 <i>操作が正常に実行された場合：</i> UserManagementTagConstants.SET_PROPERTY_OK <i>エラーが発生した場合：</i> UserManagementTagConstants.SET_PROPERTY_FAILED | C |

例

コード リスト 3-21 は、<um:setProperty> タグの使用方法の例です。

コードリスト 3-21 <um:setProperty> タグの使用例

```
<%@ taglib uri="um.tld" prefix="um" %>
.
.
.
<% String myGender = request.getParameter("gender"); %>
<um:setProperty propertySet="Demographics" propertyName="Gender"
value="<%=myGender%>" />
```

ユーザ管理： グループおよびユーザ管理タグ

ユーザ管理タグを使用すると、ユーザとグループのプロファイル情報にアクセスできるだけでなく、ユーザとグループの作成および削除や、ユーザとグループの関係の管理などの操作を行うこともできます。

以降の表の「必須」列では、属性が必須と省略可能のどちらであるかを「はい」または「いいえ」で示します。「R/C」列では、「C」は属性がコンパイル時表現であることを、「R」は属性が要求時表現とコンパイル時表現のいずれかであることを示します。

<um:addGroupToGroup>

| | |
|----------------|--|
| タグライブラリ | um.tld |
| インポート文 | <pre><%@ taglib uri="um.tld" prefix="um" %> すべてのユーザ管理タグは、同じファイルに結果を送る。結果を確認する場合、次のインポート ディレクティブをページの先頭に配置する。 <%@ page import="com.bea.pl3n.usermgmt.servlets.jsp.tags.UserManagementTagConstants" %></pre> |

| | |
|-----------------|--|
| 実装されるクラス | AddGroupToGroupTag ResultTagExtraInfo |
|-----------------|--|

<um:addGroupToGroup> タグ (表 3-21) は、childGroupName 属性で指定されたグループを、parentGroupName 属性で指定された親グループに追加します。グループは親を 1 つしか持てないので、そのグループが別の親に属していることを表す以前のデータベース レコードはすべて破棄されます。このタグが正常に動作するには、親グループも子グループもあらかじめ存在していなければなりません。このタグには本体はありません。

注意： このタグは、現在のレルムが `weblogic.security.acl.ManageableRealm` の実装である場合にのみ呼び出すことが推奨されます。このインタフェースはデフォルトの WebLogic Portal レルム (`com.bea.p13n.security.realm.RDBMSRealm`) によって実装されます。

タグの属性

表 3-21 は、<um:addGroupToGroup> タグの属性についての説明です。

表 3-21 <um:addGroupToGroup>

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-----------------|-----|--------|--|-----|
| childGroupName | はい | String | 子グループの名前。 例：“<%=childGroupName%>” | R |
| parentGroupName | いいえ | String | 親グループの名前。 例：“<%=parentGroupName%>” | R |
| result | はい | String | addGroupToGroup 操作の結果を格納する Integer 型変数の名前。 この変数を取り得る値： <i>操作が正常に実行された場合：</i> <code>UserManagementTagConstants.ADD_GROUP_OK</code> <i>エラーが発生した場合：</i> <code>UserManagementTagConstants.ADD_GROUP_FAILED</code> | C |

例

コードリスト 3-22 は、`<um:addGroupToGroup>` タグを使用して、既存のユーザグループに新しいユーザグループを追加する方法を示しています。

コードリスト 3-22 `<um:addGroupToGroup>` タグの使用例

```
<%@ taglib uri="um.tld" prefix="um" %>
.
.
.
<um:addGroupToGroup childGroupName="<%=childGroupName%>"
parentGroupName="<%=parentGroupName%>" result="result"/>
```

`<um:addUserToGroup>`

| | |
|----------|---|
| タグライブラリ | um.tld |
| インポート文 | <pre><%@ taglib uri="um.tld" prefix="um" %></pre> <p>すべてのユーザ管理タグは、同じファイルに結果を送る。結果を確認する場合、次のインポート ディレクティブをページの先頭に配置する。</p> <pre><%@ page import="com.bea.pl3n.usermgmt. servlets.jsp.tags.UserManagementTag- Constants" %></pre> |
| 実装されるクラス | AddUserToGroupTag ResultTagExtraInfo |

`<um:addUserToGroup>` タグは、`userName` 属性で名前が指定されたユーザを、`groupName` 属性で名前が指定されたグループに追加します。このタグが正常に動作するには、指定されたユーザも指定されたグループもあらかじめ存在していなければなりません。このタグには本体はありません。

注意： このタグは、現在のレルムが `weblogic.security.acl.ManageableRealm` の実装である場合にのみ呼び出すことが推奨されます。このインターフェースはデフォルトの WebLogic Portal レルム (`com.bea.pl3n.security.realm.RDBMSRealm`) によって実装されます。

タグの属性

表 3-22 は、`<um:addUserToGroup>` タグの属性についての説明です。

表 3-22 `<um:addUserToGroup>`

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------------------------|----|--------|--|-----|
| <code>userName</code> | はい | String | グループに追加するユーザの名前。 例：“ <code><%=username%></code> ” | R |
| <code>groupName</code> | はい | String | ユーザの追加先のグループの名前。 例：“ <code><%=groupName%></code> ” | R |
| <code>result</code> | はい | String | <code>addUserToGroup</code> 操作の結果を格納する Integer 型変数の名前。 この変数を取り得る値： <i>操作が正常に実行された場合：</i> <code>UserManagementTagConstants.ADD_USER_OK</code> <i>エラーが発生した場合：</i> <code>UserManagementTagConstants.ADD_USER_FAILED</code> | C |

例

コード リスト 3-23 は、`<um:addUserToGroup>` タグを使用して、既存のグループに新しいユーザを追加する方法を示しています。

コード リスト 3-23 <um:addUserToGroup> タグの使用例

```
<%@ taglib uri="um.tld" prefix="um" %>
.
.
<um:addUserToGroup userName="<%=userName%>"
groupname="<%=groupName%>" result="result"/>
```

<um:createGroup>

| | |
|-----------------|---|
| タグ ライブラリ | um.tld |
| インポート文 | <pre><%@ taglib uri="um.tld" prefix="um" %> すべてのユーザ管理タグは、同じファイルに結果を送る。結果を確認する場合、次のインポート ディレクティブをページの先頭に配置する。 <%@ page import="com.bea.p13n.usermgmt. servlets.jsp.tags.UserManagementTag- Constants" %></pre> |
| 実装されるクラス | <pre>CreateGroupTag CreateGroupExtraInfo</pre> |

<um:createGroup> タグ (表 3-23) は、新しいグループをレルム内に作成し、対応するグループ プロファイルをパーソナライゼーション データベース内に作成します。このタグには本体はありません。

注意： このタグは、現在のレルムが

weblogic.security.acl.ManageableRealm の実装である場合にのみ呼び出すことが推奨されます。このインタフェースはデフォルトの WebLogic Portal レルム (com.bea.p13n.security.realm.RDBMSRealm) によって実装されます。

タグの属性

表 3-23 は、`<um:createGroup>` タグの属性についての説明です。

表 3-23 `<um:createGroup>` タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------------|-----|--------|---|-----|
| groupName | はい | String | 新しいグループの名前。 例：“ <code><%=groupName%></code> ” | R |
| id | いいえ | String | 当該ページのスコープ内で、作成された Group オブジェクトへのアクセスに使用できる変数名。 | C |
| parentName | いいえ | String | 新しいグループの親として設定するグループの名前。 | R |
| result | はい | String | createGroup 操作の結果を格納する Integer 型変数の名前。 この変数が取り得る値： <i>操作が正常に実行された場合：</i> <code>UserManagementTagConstants.CRE ATE_GROUP_OK</code> <i>エラーが発生した場合：</i> <code>UserManagementTagConstants.CRE ATE_GROUP_FAILED</code> <i>指定された名前のグループがすでに存在する場合：</i> <code>UserManagementTagConstants.GRO UP_EXISTS</code> | C |

例

コード リスト 3-24 は、`<um:creategroup>` タグを使用して新しいグループを作成する方法を示しています。

コード リスト 3-24 <um:creategroup> タグの使用例

```

<%@ taglib uri="um.tld" prefix="um" %>
.
.
.
<um:creategroup groupName="<%=groupName%>" result="result"/>

```

<um:createUser>

| | |
|-----------------|---|
| タグ ライブラリ | um.tld |
| インポート文 | <pre> <%@ taglib uri="um.tld" prefix="um" %> </pre> <p>すべてのユーザ管理タグは、同じファイルに結果を送る。結果を確認する場合、次のインポート ディレクティブをページの先頭に配置する。</p> <pre> <%@ page import="com.bea.p13n.usermgmt. servlets.jsp.tags.UserManagementTag- Constants" %> </pre> |
| 実装されるクラス | <p>CreateUserTag</p> <p>CreateUserExtraInfo</p> |

<um:createUser> タグは、新しいユーザ プロファイルを作成します。このタグには本体はありません。このタグは、グループおよびユーザ管理タグとして分類されていますが、指定によっては、現在の匿名ユーザ プロファイルに関連付けられているあらゆるプロパティを永続化するので、実行時処理と組み合わせて使うことができます。

注意： このタグは、現在のレルムが

weblogic.security.acl.ManageableRealm の実装である場合にのみ呼び出すことが推奨されます。このインタフェースはデフォルトの WebLogic Portal レルム (com.bea.p13n.security.realm.RDBMSRealm) によって実装されます。

タグの属性

表 3-24 は、`<um:createUser>` タグの属性についての説明です。

表 3-24 `<um:createUser>` タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|---------------|-----|--------|--|-----|
| userName | はい | String | 新しいユーザの名前。 例：“ <code><%=username%></code> ” | R |
| password | はい | String | 新しいユーザのパスワード。 例：“ <code><%=password%></code> ” | R |
| profileType | いいえ | String | 拡張ユーザタイプ（例： WLCS_Customer）を指定して、そのタイプ のユーザを作成する。 | R |
| saveAnonymous | いいえ | String | 新しく作成されるユーザのプロファイル に、現在の匿名ユーザ プロファイルの属 性を永続的に保存するかどうかを指定す る。 デフォルトは false。 例：“false” | R |
| id | いいえ | String | 当該ページのスコープ内で、作成された User オブジェクトへのアクセスに使用で きる変数名。 | C |
| result | はい | String | createUser 操作の結果を格納する Integer 型変数の名前。 この変数が取り得る値： 操作が正常に実行された場合： UserManagementTagConstants.CRE ATE_USER_OK エラーが発生した場合： UserManagementTagConstants.CRE ATE_USER_FAILED 指定された名前のユーザがすでに存在す る場合： UserManagementTagConstants.USE R_EXISTS | C |

例

コードリスト 3-25 は、`<um:createUser>` タグを使用して新しいユーザを作成する方法を示しています。

コードリスト 3-25 `<um:createUser>` タグの使用例

```
<%@ taglib uri="um.tld" prefix="um" %>
.
<um:createUser userName="<%=username%>" password="<%=password%>"
result="result"/>
```

`<um:getChildGroupNames>`

| | |
|----------|--|
| タグライブラリ | um.tld |
| インポート文 | <pre><%@ taglib uri="um.tld" prefix="um" %></pre> <p>すべてのユーザ管理タグは、同じファイルに結果を送る。結果を確認する場合、次のインポートディレクティブをページの先頭に配置する。</p> <pre><%@ page import="com.bea.pl3n.usermgmt. servlets.jsp.tags.UserManagementTag- Constants" %></pre> |
| 実装されるクラス | <pre>GetChildGroupNamesTag GetChildGroupNamesExtraInfo</pre> |

`<um:getChildGroupNames>` タグは、指定されたグループの子に当たるすべてのグループの名前を返します。

タグの属性

表 3-25 は、`<um:getChildGroupNames>` タグの属性についての説明です。

表 3-25 <um:getChildGroupNames>

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-----------|----|--------|-------------------------------|-----|
| groupName | はい | String | 子グループを検索する対象のグループの名前。 | R |
| id | はい | String | 子グループ名の String 配列を格納する識別子の名前。 | C |

例

コードリスト 3-26 は、<um:getChildGroupNames> タグを使用して、グループ SomeGroup の子グループの名前を取り出す方法を示しています。

コードリスト 3-26 <um:getChildGroupNames> タグの使用例

```
<%@ taglib uri="um.tld" prefix="um" %>
.
.
.
<um:getChildGroupNames groupName="SomeGroup"
id="childrenOfSomeGroup" />
```

<um:getGroupNamesForUser>

| | |
|---------|---|
| タグライブラリ | um.tld |
| インポート文 | <%@ taglib uri="um.tld" prefix="um" %> すべてのユーザ管理タグは、同じファイルに結果を送る。結果を確認する場合、次のインポート ディレクティブをページの先頭に配置する。 <%@ page import="com.bea.pl3n.usermgmt.servlets.jsp.tags.UserManagementTagConstants" %> |

| | |
|----------|--|
| 実装されるクラス | GetGroupNamesForUserTag GetGroupNamesForUserExtraInfo |
|----------|--|

`<um:getGroupNamesForUser>` タグは、指定されたユーザが直接所属するグループに一致するグループ名を検索し、結果を `String` 配列に入れて返します。このタグには本体はありません。

タグの属性

表 3-26 は、`<um:getGroupNamesForUser>` タグの属性についての説明です。

表 3-26 `<um:getGroupNamesForUser>` タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-----------------------|----|---------------------|---|-----|
| <code>userName</code> | はい | <code>String</code> | 所属グループを検索する対象のユーザの名前。 例：“ <code><%=username%></code> ” | R |
| <code>id</code> | はい | <code>String</code> | グループ名の検索結果が格納される変数の名前。 例：“ <code>myGroups</code> ” | C |

例

コードリスト 3-27 は、`<um:getGroupNamesForUser>` タグを使用して、ユーザに適用するグループ名を取り出す方法を示しています。

コードリスト 3-27 `<um:getGroupNamesForUser>` タグの使用例

```
<%@ taglib uri="um.tld" prefix="um" %>
.
.
.
<um:getGroupNamesForUser userName="<%=username%>" id="myGroups" />
```

<um:getParentGroupName>

| | |
|-----------------|---|
| タグライブラリ | um.tld |
| インポート文 | <pre><%@ taglib uri="um.tld" prefix="um" %></pre> <p>すべてのユーザ管理タグは、同じファイルに結果を送る。結果を確認する場合、次のインポートディレクティブをページの先頭に配置する。</p> <pre><%@ page import="com.bea.p13n.usermgmt.servlets.jsp.tags.UserManagementTagConstants" %></pre> |
| 実装されるクラス | GetParentGroupNameTag GetParentGroupNameExtraInfo |

<um:getParentGroupName> タグは、`groupName` 属性で指定されたグループの親に当たるグループの名前を検索します。情報はレلمムから取得されます。このタグには本体はありません。

タグの属性

表 3-27 は、<um:getParentGroupName> タグの属性についての説明です。

表 3-27 <um:getParentGroupName>

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------------------------|----|--------|---|-----|
| <code>groupName</code> | はい | String | 親のグループ名を検索する対象のグループの名前。 例：“<%=groupName%>” | R |
| <code>id</code> | はい | String | 当該ページのスコープ内で、親の名前にアクセスするために使用できる変数名。 例：“parentGroupName” | C |

例

コードリスト 3-28 は、`<um:getParentGroupName>` タグを使用して親グループ名を取得する方法を示しています。

コードリスト 3-28 `<um:getParentGroupName>` タグの使用例

```
<%@ taglib uri="um.tld" prefix="um" %>
.
.
.
<um:getParentGroupName groupName="<%=groupName%>"
id="parentGroupName" />
```

`<um:getTopLevelGroups>`

| | |
|-----------------|---|
| タグライブラリ | um.tld |
| インポート文 | <pre><%@ taglib uri="um.tld" prefix="um" %></pre> <p>すべてのユーザ管理タグは、同じファイルに結果を送る。結果を確認する場合、次のインポート ディレクティブをページの先頭に配置する。</p> <pre><%@ page import="com.bea.pl3n.usermgmt. servlets.jsp.tags.UserManagementTag- Constants" %></pre> |
| 実装されるクラス | <pre>GetTopLevelGroupsTag GetTopLevelGroupsExtraInfo</pre> |

`<um:getTopLevelGroups>` タグは、親グループを持たないグループの名前を検索し、結果を配列に入れて返します。情報はレルムから取得されます。このタグには本体はありません。

タグの属性

表 3-28 は、`<um:getTopLevelGroups>` タグの属性についての説明です。

表 3-28 <um:getTopLevelGroups>

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------|----|--------|---|-----|
| id | はい | String | 当該ページのスコープ内で、トップレベルの Group オブジェクトにアクセスするために使用できる変数名。 例: "topLevelGroups" | C |

例

コード リスト 3-29 は、<um:getTopLevelGroups> タグの使用方法の例です。

コード リスト 3-29 <um:getTopLevelGroups> タグの使用例

```
<%@ taglib uri="um.tld" prefix="um" %>
.
.
.
<um:getTopLevelGroups id="topLevelGroups"/>
```

<um:getUsernames>

| | |
|----------|---|
| タグ ライブラリ | um.tld |
| インポート文 | <pre><%@ taglib uri="um.tld" prefix="um" %></pre> <p>すべてのユーザ管理タグは、同じファイルに結果を送る。結果を確認する場合、次のインポート ディレクティブをページの先頭に配置する。</p> <pre><%@ page import="com.bea.pl3n.usermgmt. servlets.jsp.tags.UserManagementTag- Constants" %></pre> |
| 実装されるクラス | <pre>GetUsernamesTag GetUsernamesExtraInfo</pre> |

`<um:getUsernames>` タグは、指定された検索式に一致するユーザ名を検索し、結果を `String` 配列に入れて返します。検索式では、ワイルドカード文字としてアスタリスク (*) だけを使用でき、大文字と小文字は区別されません。検索式では任意の数のアスタリスクを使用できます。このタグには本体はありません。

タグの属性

表 3-29 は、`<um:getUsernames>` タグの属性についての説明です。

表 3-29 `<um:getUsernames>` タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------------------------|-----|--|---|-----|
| <code>searchExp</code> | いいえ | <code>String</code> | ユーザ名の検索に適用される検索式。デフォルトは「*」。 例：“{*” | R |
| <code>userLimit</code> | いいえ | <code>String</code> (representing an Integer) | 検索結果として返されるユーザの最大数 (<code>Integer.valueOf</code> を個別に備える <code>String</code>)。デフォルトは 100。 ユーザ数が <code>userLimit</code> を超える場合、 <code>id</code> に返される配列の長さが <code>userLimit</code> の長さ にまで切り詰められる。 例：“500” | R |
| <code>id</code> | はい | <code>String</code> | ユーザ名の検索結果を格納する変数の名前。 例：“myUsers” | C |

表 3-29 <um:getUsernames> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------|-----|--------|---|-----|
| result | いいえ | String | <p>getUsernames 操作の結果を格納する Integer 型変数の名前。 この変数が取り得る値： 操作が正常に実行された場合： UserManagerTagConstants.USER_SEARCH_OK 一般エラーの場合（検索の間に発生したデータベース接続障害など）： UserManagerTagConstants.USER_SEARCH_FAILED</p> <p>注意： 検索条件に合致するユーザがない場合、結果は UserManagerTagConstants.USER_SEARCH_FAILED にはならず、「id」に返される配列の長さがゼロになる。</p> | C |

注意： USER_SEARCH_FAILED の値が返されるのは、ユーザの検索中に、データベース接続エラーなどの一般エラーが発生したときだけです。検索条件に合致するユーザがない場合、結果は UserManagerTagConstants.USER_SEARCH_FAILED にはならず、id に返される配列の長さがゼロになります。

例

コードリスト 3-30 は、<um:getUsernames> タグを使用してユーザ名を取得する方法を示しています。

コードリスト 3-30 <um:getUsernames> タグの使用例

```
<%@ taglib uri="um.tld" prefix="um" %>
.
.
.
```

```
<um:getUsernames userLimit="500" searchExp="t*" id="myUsers"/>
<%System.out.println("I found " + myUsers.length + " users.");%>
```

<um:getUsernamesForGroup>

| | |
|-----------------|--|
| タグライブラリ | um.tld |
| インポート文 | <pre><%@ taglib uri="um.tld" prefix="um" %></pre> <p>すべてのユーザ管理タグは、同じファイルに結果を送る。結果を確認する場合、次のインポートディレクティブをページの先頭に配置する。</p> <pre><%@ page import="com.bea.pl3n.usermgmt. servlets.jsp.tags.UserManagementTag- Constants" %></pre> |
| 実装されるクラス | <p>GetUsernamesForGroupTag</p> <p>GetUsernamesForGroupExtraInfo</p> |

<um:getUsernamesForGroup> タグは、指定された検索式に一致し、かつ指定されたグループに属するユーザ名を検索し、結果を `String` 配列に入れて返します。検索式では、ワイルドカード文字としてアスタリスク (*) だけを使用でき、大文字と小文字は区別されません。検索式では任意の数のアスタリスクを使用できます。このタグには本体はありません。

タグの属性

表 3-30 は、<um:getUsernamesForGroup> タグの属性についての説明です。

表 3-30 <um:getUsernamesForGroup> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-----------|-----|--------|--|-----|
| searchExp | いいえ | String | ユーザ名の検索に適用される検索式。デフォルトは「*」。 例: "t*" | R |

表 3-30 <um:getUsernamesForGroup> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-----------|-----|----------------------------|---|-----|
| groupName | はい | String | 検索式で取り出すユーザが属するグループの名前。 例：“engineering” | R |
| userLimit | いいえ | String (Integer を表す) | 検索結果として返されるユーザの最大数 (Integer.valueOf を個別に備える String)。デフォルトは 100。 ユーザ数が userLimit を超える場合、id に 返される配列の長さが userLimit の長さ にまで切り詰められる。 例：“500” | R |
| id | はい | String | ユーザ名の検索結果を格納する変数の 名前。 例：“myUsers” | C |
| result | いいえ | String | getUsernamesForGroup 操作の結果を格納 する Integer 型変数の名前。 この変数が取り得る値： 操作が正常に実行された場合： UserManagementTagConstants.USE R_SEARCH_OK 一般エラーの場合： UserManagementTagConstants.USE R_SEARCH_FAILED | C |

注意： USER_SEARCH_FAILED の値が返されるのは、ユーザの検索中に、データベース接続エラーなどの一般エラーが発生したときだけです。検索条件に合致するユーザがない場合、結果は UserManagementTagConstants.USER_SEARCH_FAILED にはならず、id に返される配列の長さがゼロになります。

例

コード リスト 3-31 は `<um:getUsernamesForGroup>` タグの使用法の例です。指定された検索式に一致し、かつ指定されたグループに属するユーザの名前を取得します。

コード リスト 3-31 `<um:getUsernamesForGroup>` タグの使用例

```
<%@ taglib uri="um.tld" prefix="um" %>
.
.
.
<um:getUsernamesForGroup groupName="engineering" userLimit="500"
searchExp="t*" id="myUsers"/>
<%System.out.println("I found " + myUsers.length + " users in my
group.");%>
```

`<um:removeGroup>`

| | |
|-----------------|---|
| タグ ライブラリ | um.tld |
| インポート文 | <pre><%@ taglib uri="um.tld" prefix="um" %></pre> <p>すべてのユーザ管理タグは、同じファイルに結果を送る。結果を確認する場合、次のインポート ディレクティブをページの先頭に配置する。</p> <pre><%@ page import="com.bea.pl3n.usermgmt. servlets.jsp.tags.UserManagementTag- Constants" %></pre> |
| 実装されるクラス | RemoveGroupTag ResultTagExtraInfo |

`<um:removeGroup>` タグは、`groupName` 属性で指定された名前のグループを削除します。このタグには本体はありません。

注意: このタグは、現在のレルムが `weblogic.security.acl.ManageableRealm` の実装である場合にのみ呼び出すことが推奨されます。このインターフェースはデフォルトの WebLogic Portal レルム (`com.bea.pl3n.security.realm.RDBMSRealm`) によって実装されます。

タグの属性

表 3-31 は、`<um:removeGroup>` タグの属性についての説明です。

表 3-31 `<um:removeGroup>`

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------------------------|----|--------|---|-----|
| <code>groupName</code> | はい | String | 削除するグループの名前。 例: “ <code><%=groupName%></code> ” | R |
| <code>result</code> | はい | String | <code>removeGroup</code> 操作の結果を格納する Integer 型変数の名前。 この変数を取り得る値: <i>操作が正常に実行された場合:</i> <code>UserManagementTagConstants.REM_OVE_GROUP_OK</code> <i>エラーが発生した場合:</i> <code>UserManagementTagConstants.REM_OVE_GROUP_FAILED</code> | C |

例

コード リスト 3-32 は、`<um:removeGroup>` タグを使用して、`groupName` 属性で指定された名前のグループを削除する方法を示しています。

コード リスト 3-32 `<um:removeGroup>` タグの使用例

```
<%@ taglib uri="um.tld" prefix="um" %>
.
.
.
<um:removeGroup groupName="<%=groupName%>" result="result"/>
```

<um:removeGroupFromGroup>

| | |
|-----------------|---|
| タグライブラリ | um.tld |
| インポート文 | <pre><%@ taglib uri="um.tld" prefix="um" %></pre> <p>すべてのユーザ管理タグは、同じファイルに結果を送る。結果を確認する場合、次のインポート ディレクティブをページの先頭に配置する。</p> <pre><%@ page import="com.bea.pl3n.usermgmt. servlets.jsp.tags.UserManagementTag- Constants" %></pre> |
| 実装されるクラス | RemoveGroupFromGroupTag ResultTagExtraInfo |

<um:removeGroupFromGroup> タグは、親グループから子グループを 1 つ削除します。

タグの属性

表 3-32 は、<um:removeGroupFromGroup> タグの属性についての説明です。

表 3-32 <um:removeGroupFromGroup>

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-----------------|----|--------|----------------------|-----|
| childGroupName | はい | String | 親から削除する子グループの名前。 | R |
| parentGroupName | はい | String | 子グループが削除される親グループの名前。 | R |

表 3-32 <um:removeGroupFromGroup> (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------|----|--------|---|-----|
| result | はい | String | removeGroupFromGroup 操作の結果を格納する Integer 型変数の名前。 この変数を取り得る値： <i>操作が正常に実行された場合：</i> UserManagementTagConstants.REM_OVE_GROUP_OK <i>操作の実行に失敗した場合：</i> UserManagementTagConstants.REM_OVE_GROUP_FAILED | C |

例

コード リスト 3-33 は、<um:removeGroupFromGroup> タグを使用して、親グループから子グループを削除する方法を示しています。

コード リスト 3-33 <um:removeGroupFromGroup> タグの使用例

```
<%@ taglib uri="um.tld" prefix="um" %>
.
.
.
<um:removeGroupFromGroup parentGroupName="SomeGroup"
childGroupName="ChildGroupToRemove" result="myResult" />
```

<um:removeUser>

| | |
|---------|--------|
| タグライブラリ | um.tld |
|---------|--------|

| | |
|-----------------|---|
| インポート文 | <pre><%@ taglib uri="um.tld" prefix="um" %> すべてのユーザ管理タグは、同じファイルに結果を送る。結果を確認する場合、次のインポート ディレクティブをページの先頭に配置する。 <%@ page import="com.bea.pl3n.usermgmt. servlets.jsp.tags.UserManagementTag- Constants" %></pre> |
| 実装されるクラス | <pre>RemoveUserTag ResultTagExtraInfo</pre> |

`<um:removeUser>` タグは、`userName` 属性で指定されたユーザを削除します。このタグでは、データベース内で `profileType` が設定されている任意のタイプの拡張ユーザを削除できます。このタグには本体はありません。

注意： このタグは、現在のレルムが

`weblogic.security.acl.ManageableRealm` の実装である場合にのみ呼び出すことが推奨されます。このインタフェースはデフォルトの WebLogic Portal レルム (`com.bea.pl3n.security.realm.RDBMSRealm`) によって実装されます。

タグの属性

表 3-33 は、`<um:removeUser>` タグの属性についての説明です。

表 3-33 `<um:removeUser>`

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-----------------------|----|--------|---|-----|
| <code>userName</code> | はい | String | 削除するユーザのユーザ名。 例：“ <code><%=username%></code> ” | R |

表 3-33 <um:removeUser> (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------|----|--------|---|-----|
| result | はい | String | removeUser 操作の結果を格納する Integer 型変数の名前。 この変数が取り得る値： <i>操作が正常に実行された場合：</i> UserManagementTagConstants.REM_OVE_USER_OK <i>エラーが発生した場合：</i> UserManagementTagConstants.REM_OVE_USER_FAILED | C |

例

コードリスト 3-34 は、<um:removeUser> タグを使用してユーザを削除する方法を示しています。

コードリスト 3-34 <um:removeUser> タグの使用例

```
<%@ taglib uri="um.tld" prefix="um" %>
.
.
.
<um:removeUser userName="<%=username%>" result="result"/>
```

<um:removeUserFromGroup>

| | |
|---------|--------|
| タグライブラリ | um.tld |
|---------|--------|

| | |
|-----------------|---|
| インポート文 | <pre><%@ taglib uri="um.tld" prefix="um" %> すべてのユーザ管理タグは、同じファイルに結果を送る。結果を確認する場合、次のインポートディレクティブをページの先頭に配置する。 <%@ page import="com.bea.pl3n.usermgmt.servlets.jsp.tags.UserManagementTagConstants" %></pre> |
| 実装されるクラス | <pre>RemoveUserFromGroupTag ResultTagExtraInfo</pre> |

`<um:removeUserFromGroup>` タグは、グループから 1 人のユーザを削除します。

注意： このタグは、現在のレルムが `weblogic.security.acl.ManageableRealm` の実装である場合にのみ呼び出すことが推奨されます。このインタフェースはデフォルトの WebLogic Portal レルム (`com.bea.pl3n.security.realm.RDBMSRealm`) によって実装されます。

タグの属性

表 3-34 は、`<um:removeUserFromGroup>` タグの属性についての説明です。

表 3-34 `<um:removeUserFromGroup>`

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------------------------|----|--------|--------------------------|-----|
| <code>userName</code> | はい | String | 指定されたグループから削除するユーザのユーザ名。 | R |
| <code>groupName</code> | はい | String | 削除されるユーザが属するグループの名前。 | R |

表 3-34 <um:removeUserFromGroup> (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------|----|--------|--|-----|
| result | はい | String | removeUserFromGroup 操作の結果を格納する Integer 型変数の名前。 この変数が取り得る値： <i>操作が正常に実行された場合：</i> UserManagementTagConstants.REM_OVE_USER_OK <i>操作の実行に失敗した場合：</i> UserManagementTagConstants.REM_OVE_USER_FAILED | C |

例

コードリスト 3-35 は、<um:removeUserFromGroup> タグを使用して、親グループから子グループを削除する方法を示しています。

コードリスト 3-35 <um:removeUserFromGroup> タグの使用例

```
<%@ taglib uri="um.tld" prefix="um" %>
.
.
.
<um:removeUserFromGroup username="UserToRemove"
groupname="SomeGroup" result="myResult" />
```

ユーザ管理：セキュリティ タグ

ユーザ管理タグを使用すると、ユーザとグループのプロファイル情報にアクセスできるだけでなく、ユーザとグループの作成および削除や、ユーザとグループの関係の管理などの操作を行うこともできます。

以降の表の「必須」列では、属性が必須と省略可能のどちらであるかを「はい」または「いいえ」で示します。「R/C」列では、「C」は属性がコンパイル時表現であることを、「R」は属性が要求時表現とコンパイル時表現のいずれかであることを示します。

<um:login>

| | |
|-----------------|---|
| タグライブラリ | um.tld |
| インポート文 | <pre><%@ taglib uri="um.tld" prefix="um" %></pre> <p>すべてのユーザ管理タグは、同じファイルに結果を送る。結果を確認する場合、次のインポート ディレクティブをページの先頭に配置する。</p> <pre><%@ page import="com.bea.pl3n.usermgmt.servlets.jsp.taglib.UserManagement-TagConstants" %></pre> |
| 実装されるクラス | LoginTag ResultTagExtraInfo |

<um:login> タグは、現在のセキュリティ レルムに対する簡易認証（ユーザ名、パスワード）を提供し、認証済みユーザを現在の WebLogic ユーザとして設定します。このタグには本体はありません。

注意： login タグが機能するには、username パラメータと password パラメータが HTTP リクエスト内に存在していなければなりません。

タグの属性

表 3-35 は、<um:login> タグの属性についての説明です。

表 3-35 <um:login> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------|----|--------|--|-----|
| result | はい | String | login 操作の結果を格納する Integer 型変数の名前。 この変数が取り得る値： <i>操作が正常に実行された場合：</i> UserManagementTagConstants.LOG_IN_OK <i>認証実行時に一般エラーが発生した場合：</i> UserManagementTagConstants.LOG_IN_ERROR <i>ユーザ名とパスワードの組み合わせが無効なために認証が失敗した場合</i> UserManagementTagConstants.LOG_IN_FAILED | C |

<um:logout>

| | |
|-----------------|--|
| タグライブラリ | um.tld |
| インポート文 | <pre><%@ taglib uri="um.tld" prefix="um" %></pre> <p>すべてのユーザ管理タグは、同じファイルに結果を送る。結果を確認する場合、次のインポート ディレクティブをページの先頭に配置する。</p> <pre><%@ page import="com.bea.pl3n.usermgmt.servlets.jsp.taglib.UserManagementTagConstants" %></pre> |
| 実装されるクラス | LogoutTag ResultTagExtraInfo |

<um:logout> タグは、現在のユーザの WebLogic Server セッションを終了します。このタグは <um:login> タグと組み合わせて使用することが推奨されます。

タグの属性

表 3-36 は、`<um:logout>` タグの属性についての説明です。

表 3-36 `<um:logout>`

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------|----|------|----|-----|
| 属性はない | | | | |

`<um:setPassword>`

| | |
|----------|---|
| タグライブラリ | um.tld |
| インポート文 | <pre><%@ taglib uri="um.tld" prefix="um" %></pre> <p>すべてのユーザ管理タグは、同じファイルに結果を送る。結果を確認する場合、次のインポート ディレクティブをページの先頭に配置する。</p> <pre><%@ page import="com.bea.pl3n.usermgmt. servlets.jsp.taglib.UserManagement- TagConstants" %></pre> |
| 実装されるクラス | SetPasswordTag ResultTagExtraInfo |

`<um:setPassword>` タグは、`userName` 属性でユーザ名が指定されたユーザのパスワードを更新します。

注意： このタグは、現在のレルムが

`weblogic.security.acl.ManageableRealm` の実装である場合にのみ呼び出すことが推奨されます。このインタフェースはデフォルトの WebLogic Portal レルム (`com.bea.pl3n.security.realm.RDBMSRealm`) によって実装されます。加えて、現在のレルムによって使われるユーザオブジェクトは `weblogic.security.acl.CredentialChanger` を実装する必要があります。

タグの属性

表 3-37 は、`<um:setPassword>` タグの属性についての説明です。

表 3-37 `<um:setPassword>` タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|----------|----|--------|--|-----|
| userName | はい | String | パスワードを変更する対象のユーザのユーザ名。 | R |
| password | はい | String | 新しいユーザパスワード。 | R |
| result | はい | String | setPassword 操作の結果を格納する Integer 型変数の名前。 この変数が取り得る値： <i>操作が正常に実行された場合：</i> <code>UserManagementTagConstants.SET_PASSWORD_OK</code> <i>操作の実行に失敗した場合：</i> <code>UserManagementTagConstants.SET_PASSWORD_FAILED</code> | C |

パーソナライゼーション ユーティリティ

`<es:jsptaglib>` タグ ライブラリには、JSP ページを作成するために使用できる汎用的なタグが用意されています。

以降の表の「必須」列では、属性が必須と省略可能のどちらであるかを「はい」または「いいえ」で示します。「R/C」列では、「C」は属性がコンパイル時表現であることを、「R」は属性が要求時表現とコンパイル時表現のいずれかであることを示します。

`<es:convertSpecialChars>`

| | |
|----------|---|
| タグ ライブラリ | es.tld |
| インポート文 | <code><%@ taglib uri="es.tld" prefix="es" %></code> |
| 実装されるクラス | ConvertSpecialCharsTag |

`<es:convertSpecialChars>` タグは、通常は HTML ブラウザにとって特別な意味を持つ文字を所定の文字列に変換し、作成者の意図どおりに表示できるようにします。

たとえば、次の文はブラウザに対してタグの開始と終了を知らせる文字である「`<`」と「`>`」を使用しているので、変換する必要があります。

Enter `<filename>` here:

タグの属性

表 3-38 は、`<es:convertSpecialChars>` タグの属性についての説明です。

表 3-38 `<es:convertSpecialChars>`

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------|----|--------|-----------|-----|
| string | はい | String | 変換対象の文字列。 | R |

例

コードリスト 3-36 は、小なり符号を含む文字列を HTML で表示できるようにする例です。

コード リスト 3-36 <es:convertSpecialChars> タグの使用例

```
<%@ taglib uri="es.tld" prefix="es" %>
.
.
.
<es:convertSpecialChars string="<thisString>" />
```

<es:counter>

| | |
|----------|--|
| タグ ライブラリ | es.tld |
| インポート文 | <%@ taglib uri="es.tld" prefix="es" %> |
| 実装されるクラス | CounterTag CounterTagExtraInfo |

<es:counter> タグは、for ループ構文を作成するために使用します。

タグの属性

表 3-39 は、<es:counter> タグの属性についての説明です。

表 3-39 <es:counter>

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------|-----|--------|---|-----|
| type | いいえ | String | カウンタのデータ型。取り得る値は int または long。デフォルトは int。 | R |
| id | はい | String | 変数の固有名。 | R |

表 3-39 <es:counter> (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|----------|----|------|-----------|-----|
| minCount | はい | Int | ループの開始位置。 | R |
| maxCount | はい | Int | ループの終了位置。 | R |

例

コードリスト 3-37 は、<es:counter> タグの使用方法の例です。

コードリスト 3-37 <es:counter> タグの使用例

```
<%@ taglib uri="es.tld" prefix="es" %>
.
.
.
<es:counter id="iterator" minCount="0" maxCount="10">
  <% System.out.println(iterator);%>
</es:counter>
```

<es:date>

| | |
|----------|--|
| タグライブラリ | es.tld |
| インポート文 | <%@ taglib uri="es.tld" prefix="es" %> |
| 実装されるクラス | DateTag |

<es:date> タグは、ユーザの時間帯設定に基づいて書式化された日時文字列 (String) を取得するために使用します。

タグの属性

表 3-40 は、<es:date> タグの属性についての説明です。

表 3-40 <es:date>

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------------|-----|--------|---|-----|
| timeZoneId | いいえ | String | デフォルトはサーバ上の時間帯。 | R |
| formatStr | いいえ | String | java.text.SimpleDateFormat に従う日時書式文字列。デフォルト値は「MM/dd/yyyy HH:mm:ss:z」。 | R |

例

コード リスト 3-38 では、<es:date> タグを使用して日時を取得する方法を示しています。

コード リスト 3-38 <es:date> タグの使用例

```
<%@ taglib uri="es.tld" prefix="es" %>
.
.
.
<es:date formatStr="MMMM dd yyyy" timeZoneId="MST" />
```

<es:forEachInArray>

| | |
|----------|--|
| タグライブラリ | es.tld |
| インポート文 | <%@ taglib uri="es.tld" prefix="es" %> |
| 実装されるクラス | ForEachInArrayTag ForEachInArrayTagExtraInfor |

<es:forEachInArray> タグは、配列に対して反復処理を行うために使用しません。

タグの属性

表 3-41 は、`<es:forEachInArray>` タグの属性についての説明です。

表 3-41 `<es:forEachInArray>`

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-----------|-----|------------|----------------|-----|
| id | はい | String | 配列内の各値を格納する変数。 | R |
| type | はい | String | 配列内の各値のデータ型。 | R |
| array | はい | Object [] | 反復処理の対象となる配列。 | R |
| counterId | いいえ | String | 配列内の現在位置。 | R |

例

コード リスト 3-39 は、`<es:forEachInArray>` タグを使用して、配列に対して反復処理を行う方法を示しています。

コード リスト 3-39 `<es:forEachInArray>` タグの使用例

```
<es:forEachInArray id="item" array="<%=items%>" type="String"
counterId="i">
  <% System.out.println("items[" + i + "]: " + item);%>
</es:forEachInArray>
```

`<es:isNull>`

| | |
|----------|---|
| タグライブラリ | es.tld |
| インポート文 | <code><%@ taglib uri="es.tld" prefix="es" %></code> |
| 実装されるクラス | IsNullTag |

`<es:isNull>` タグは、値が `null` かどうかを調べるために使用します。対象の値が `String` 型の場合、`<es:isNull>` タグは、その `String` が `null` かそれとも値を持っているかを調べるために使われます。空の文字列の場合、`isNull` は `false` になります（空の文字列は `null` ではありません）。

タグの属性

表 3-42 は、`<es:isNull>` タグの属性についての説明です。

表 3-42 `<es:isNull>` タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------------|----|---------------------|---------|-----|
| <code>item</code> | はい | <code>Object</code> | 評価する変数。 | R |

例

コードリスト 3-40 は、`<es:isNull>` タグを使用して、値が `null` かどうかを調べる方法を示しています。

コード リスト 3-40 `<es:isNull>` タグの使用例

```
<%@ taglib uri="es.tld" prefix="es" %>
.
.
.
<es:isNull item="<%=value%%">
    Error: the value is null.
</es:isNull>
```

`<es:notNull>`

| | |
|---------|---|
| タグライブラリ | <code>es.tld</code> |
| インポート文 | <code><%@ taglib uri="es.tld" prefix="es" %></code> |

| | |
|----------|------------|
| 実装されるクラス | NotNullTag |
|----------|------------|

`<es:notNull>` タグは、値が `null` でないかどうかを調べるために使われます。対象の値が `String` 型の場合、`<es:notNull>` タグは、その `String` が `null` でないか、それとも値を持っているかを調べるために使われます。空の文字列の場合、`notNull` は `true` になります（空の文字列は値として扱われます）。

タグの属性

表 3-43 は、`<es:notNull>` タグの属性についての説明です。

表 3-43 `<es:notNull>`

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------|----|--------|---------|-----|
| item | はい | Object | 評価する変数。 | R |

例

コードリスト 3-41 は、`<es:notNull>` タグを使用して、値が `null` でないかどうかを調べる方法を示しています。

コードリスト 3-41 `<es:notNull>` タグの使用例

```
<%@ taglib uri="es.tld" prefix="es" %>
.
.
.
<es:notNull item="<%=value%>">
    The value is not null.
</es:notNull>
```

<es:transposeArray>

| | |
|----------|---|
| タグライブラリ | es.tld |
| インポート文 | <%@ taglib uri="es.tld" prefix="es" %> |
| 実装されるクラス | TransposeArrayTag TransposeArrayTagExtraInfo |

<es:transposeArray> タグは、標準の [行][列] 配列を [列][行] 配列に転置するために使用します。

タグの属性

表 3-44 は、<es:transposeArray> タグの属性についての説明です。

表 3-44 <es:transposeArray>

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------|----|------------|-------------------------------------|-----|
| id | はい | String | [列][行] 配列を格納する変数。 | R |
| type | はい | String | [行][列] 配列内の変数のデータ型 (例: String)。 | R |
| array | はい | Object[][] | [行][列] 配列を格納する変数。 | R |

例

コード リスト 3-42 は、<es:transposeArray> タグの使用方法の例です。

コード リスト 3-42 <es:transposeArray> タグの使用例

```
<%@ taglib uri="es.tld" prefix="es" %>
.
.
.
```

```
<es:transposeArray id="byColumnRow" array="<%=byRowColumn%>"
type="String">
    ...
</es:transposeArray>
```

<es:uriContent>

| | |
|-----------------|---|
| タグライブラリ | es.tld |
| インポート文 | <%@ taglib uri="es.tld" prefix="es" %> |
| 実装されるクラス | UriContentTag UriContentTagExtraInfo |

<es:uriContent> タグは、URL からコンテンツを取り出すために使用します。このタグは、大量のテキストが含まれるページを取得する用途に最も適しています。

タグの属性

表 3-45 は、<es:uriContent> タグの属性についての説明です。

表 3-45 <es:uriContent>

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------|----|--------|------------------------------|-----|
| id | はい | String | URI からダウンロードされたコンテンツを格納する変数。 | R |
| uri | はい | String | コンテンツの取得先の完全修飾 URI。 | R |

例

コード リスト 3-43 は、<es:uriContent> タグを使用して URL からコンテンツを取得する方法を示しています。

コード リスト 3-43 <es:uriContent> タグの使用例

```
<%@ taglib uri="es.tld" prefix="es" %>
.
.
.
<es:uriContent id="uriContent"
uri="http://www.beasys.com/index.html">
<%
    out.print(uriContent);
%>
</es:uriContent>
```

注意： HTML ページを相対 URL と結合する場合には、それらのページを各 URL 内の正確なホスト名まで完全修飾しなければなりません。そうしないと、(他のリソース上の) 画像をブラウザから正しく取り出すことはできません。

WebLogic ユーティリティ

`<wl:jsptaglib>` タグ ライブラリには、WebLogic Server プラットフォームの一部として提供されるカスタム JSP 拡張タグが用意されています。

以降の表の「必須」列では、属性が必須と省略可能のどちらであるかを「はい」または「いいえ」で示します。「R/C」列では、「C」は属性がコンパイル時表現であることを、「R」は属性が要求時表現とコンパイル時表現のいずれかであることを示します。

`<wl:cache>`

| | |
|----------|---|
| タグ ライブラリ | wl.tld |
| インポート文 | <code><%@ taglib uri="weblogic.tld" prefix="wl" %></code> |
| 実装されるクラス | CacheTag CacheTagInfo |

`<wl:cache>` タグでは、表示のたびにそのコンテンツが更新される必要がないことを指定します。

タグの属性

表 3-46 は、`<wl:cache>` タグの属性についての説明です。

表 3-46 `<wl:cache>`

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|---------|-----|---------|--|-----|
| timeout | いいえ | Integer | データの生存時間 (TTL)、つまりそのデータを、他のすべての制御に関係なく更新しなければならない頻度を制御する。単位は秒。 | R |

表 3-46 <wl:cache>

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------|-----|---------|--|-----|
| scope | いいえ | String | データの生存時間 (TTL)、つまりそのデータを、他のすべての制御に関係なく更新しなければならない頻度を制御する。単位は秒。 | C |
| name | いいえ | String | このキャッシュを一意に識別する。名前を指定しない場合、ランダムに生成された名前が付けられる。 | C |
| size | いいえ | Integer | キャッシュ中のエントリの最大数。デフォルトは無制限。関連付けられているキーがあるときだけ適用される。 | R |
| vars | いいえ | String | 変換されたキャッシュ出力をキャッシュするほかに、ブロック内部の計算値をキャッシュすることもできる。これらの変数は、キャッシュ キーと同様にして指定される。このタイプのキャッシュは「入力キャッシュ」と呼ばれる。 | C |
| key | いいえ | String | 対象のデータが依存している現在のページからアクセス可能な値のリスト (CSV 形式) を指定する。これらの値はキャッシュへの補助キーとして機能する。 | C |
| async | いいえ | String | async パラメータが true に設定されている場合、可能であればキャッシュは非同期的に更新される。キャッシュ検索を開始するユーザには古いデータが提示される。 | C |

<wl:process>

| | |
|----------|--|
| タグライブラリ | wl.tld |
| インポート文 | <%@ taglib uri="weblogic.tld" prefix="wl" %> |
| 実装されるクラス | ProcessTag |

`<wl:process>` タグは、クエリ属性ベースのフロー制御を行うために使われます。[コードリスト 3-44](#) に示すように、4つの属性の組み合わせを使って、`<wl:process>` タグと `</wl:process>` タグで囲まれる文を選択的に実行できます。

タグの属性

[表 3-47](#) は、`<wl:process>` タグの属性についての説明です。

表 3-47 `<wl:process>`

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|----------|-----|--------|-----------|-----|
| name | いいえ | String | クエリ属性の名前。 | R |
| notName | いいえ | String | クエリ属性の名前。 | R |
| value | いいえ | String | クエリ属性の値。 | R |
| notValue | いいえ | String | クエリ属性の値。 | R |

`<wl:process>` タグで囲まれる文は、次の表の内容に従って実行されます。

| 属性 | value | notValue | value と notValue のどちらでもない |
|----------|------------------------|--------------------------|------------------------------|
| name | 名前付き属性が value に等しい。 | 名前付き属性が value に等しくない。 | name 属性の値が null でない。 |
| not Name | | | notName 属性の値が null。 |

例

[コードリスト 3-44](#) は、`<wl:process>` タグの使用法の例です。lastBookRead が存在し、lastBookRead の値が「A Man in Full」である場合に実行されます。

コードリスト 3-44 <wl:process> タグの使用例

```
<%@ taglib uri="weblogic.tld" prefix="wl" %>
.
.
.
<wl:process name="lastBookRead" value="A Man in Full">
<!-- この部分のコードは、
      lastBookRead が存在し、lastBookRead の値が
      "A Man in Full" である場合に実行される。 -->
</wl:process>
```

<wl:repeat>

| | |
|-----------------|--|
| タグライブラリ | wl.tld |
| インポート文 | <%@ taglib uri="weblogic.tld" prefix="wl" %> |
| 実装されるクラス | RepeatTag RepeatTagInfo |

<wl:repeat> タグは、set 属性で指定されるさまざまな種類の Java オブジェクトに対して反復処理を行うために使われます。

タグの属性

表 3-48 は、<wl:repeat> タグの属性についての説明です。

表 3-48 <wl:repeat> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------|-----|--------|---|-----|
| set | いいえ | Object | 以下に示すオブジェクト セット。 <ul style="list-style-type: none"> ◆ 列挙値 ◆ イテレータ ◆ コレクション ◆ 配列 ◆ ベクタ ◆ 結果セット ◆ 結果セット メタデータ ◆ ハッシュテーブルキー | R |
| count | いいえ | Int | 指定されたセットの先頭から、この属性で指定された個数のエントリに対して反復処理を行う。 | R |
| id | いいえ | String | 現在反復処理の対象となっているオブジェクトが格納される変数。 | C |
| type | いいえ | String | 指定されたセットに対する反復処理の結果として得られるオブジェクトのデータ型。デフォルトは Object。このデータ型は完全修飾形式でなければならない。 | C |

第4章 ナビゲーション (Webflow) 用 JSP タグ

WebLogic Portal 製品スイートには、Webflow を使った JSP の開発を容易にするように設計された一連の JSP タグが含まれています。これらの定義済みタグを使うと、JSP に Webflow 関連の Java コードを入れる必要がありません。この章では、こうしたタグのセットを Web ページにインポートする方法や、個々のタグの目的を説明します。

このトピックは、以下の節で構成されています。

- [URL 作成タグ](#)
- [フォーム タグ](#)
- [検証機能付きフォーム タグ](#)
- [Pipeline セッション タグ](#)

URL 作成タグ

この節では、動的または静的な URL リンクや、JSP 内のその他のリソースを作成するのに使われる Webflow URL タグについて説明します。

注意： 以降の表の「必須」列では、属性が必須と省略可能のどちらであるかを「はい」または「いいえ」で示します。「R/C」列では、「C」は属性がコンパイル時表現であることを、「R」は属性が要求時表現とコンパイル時表現のいずれかであることを示します。

<webflow:createWebflowURL>

<webflow:createWebflowURL> タグ (表 4-1) は JSP 内で使われ、Webflow URL を JSP 内に動的に作成します。Webflow URL には、プロトコル、ドメイン名、ポート、Web アプリケーション URI、WebflowServlet URI、およびクエリ文字列を入れることができます。

| | |
|-----------------|---|
| タグライブラリ | webflow.tld |
| インポート文 | <%@ taglib uri="webflow.tld" prefix="webflow" %> |
| 実装されるクラス | CreateWebflowURLTag |

表 4-1 は、<webflow:createWebflowURL> タグの属性についての説明です。

表 4-1 <webflow:createWebflowURL>

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------------|-----|--------|---|-----|
| domainName | いいえ | String | URL のドメイン名や IP アドレスを変更するのに使う。Webflow が別のマシン上のプロキシ サーバを経由する場合に、この属性を使用できる。 | R |
| doRedirect | いいえ | String | WebflowServlet にプレゼンテーションノードへの転送ではなくリダイレクトを行わせる。有効な値は true と false。デフォルトは false (リダイレクトしない)。 | R |
| encode | いいえ | String | Webflow 側で URL をコード化するように指定する。セッションの状態を保持したいが、ブラウザがクッキーを受け付けない場合には、URL をコード化する必要がある。有効な値は true と false。デフォルト値は true。ただし、ブラウザがクッキーを受け付けない場合にしか、URL をコード化する必要はない。 | R |

表 4-1 <webflow:createWebflowURL> (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|--|-----|
| event | はい | String | Webflow は、これを起点と組み合わせて使って、指定されたネームスペース XML ファイル内の遷移先を決定する。 | R |
| extraParams | いいえ | String | リクエスト パラメータを名前と値のペアとして追加指定するのに使われる。 | R |
| httpsInd | いいえ | String | Webflow 側でプロトコルを推定するか、あるいは HTTPS または HTTP を使うように指定する。このため、有効な値は calculate、http、および https。デフォルト値は http。 Calculate を指定すると、起点/イベントの分岐連鎖のリストに含まれているいずれかのノードが、アプリケーションの WEB-INF/web.xml ファイル内の HTTPS_URL_PATTERNS コンテキスト パラメータで指定されているパターンと一致する場合には、HTTPS が選択される。Calculate のほうがより動的で処理コストがかかるが、プロトコルを強制する必要がある場合には、それをここで指定することができる。 | R |
| namespace | いいえ | String | 起点およびイベントが定義されている Webflow コンフィグレーション ファイルを示す。省略されている場合は、現在の (最後に成功した) ネームスペースが使われる。 | R |
| origin | いいえ | String | イベントの出発点となるノード (起点) で、<ノード名>.<ノードタイプ>の形式に従う。ページ名と同じ場合も異なる場合もある。省略された場合、JSP ページ名が使われる。 | R |

表 4-1 <webflow:createWebflowURL> (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------------|-----|--------|--|-----|
| pathPrefix | いいえ | String | <p>パス情報の接頭辞として使われる。この文字列は、Web アプリケーション URI の前に置かれる。プロキシ サーバが同一のマシン上にある場合に使われる。</p> <p>注意: プロキシ サーバ側では、このパス接頭辞を取り除いてから Webflow にリクエストを転送する必要がある。</p> | R |
| pathSuffix | いいえ | String | <p>パス情報の接尾辞として使われる。WebflowServlet URI に続けて、追加のパス情報が置かれる。これにより、この情報は、request.getPathInfo() メソッドから取得可能になる。</p> | R |

例

コードリスト 4-1 に、`<webflow:createWebflowURL>` JSP タグの使用法とその属性の例を示します。

注意： Web ページ上には値を直接埋め込まないようにお勧めします。

コードリスト 4-1 `<webflow:createWebflowURL>` タグと属性の使用例

```
<a href="<webflow:createWebflowURL event='link.yo' pathSuffix='/morepath'
/>">パス接尾辞 </a>
<br>

<a href="<webflow:createWebflowURL event='link.yo' pathPrefix='/pathprefix' />">
パス接頭辞 </a>
<br>

<a href="<webflow:createWebflowURL event='link.yo' pathPrefix='/pathprefix'
pathSuffix='/suffix' />">パス接頭辞とパス接尾辞 </a>
<br>

<a href="<webflow:createWebflowURL event='link.yo' domainName='123.123.123.123'
/>">ドメイン名 </a>
<br>

<a href="<webflow:createWebflowURL event='link.yo' pathSuffix='/morepath'
extraParams='sex=male' />">パス接尾辞と 1 個の追加パラメータ </a>
<br>

<a href="<webflow:createWebflowURL event='link.yo' pathSuffix='/morepath'
extraParams='sex=male&animal=dog' />">パス接尾辞と 2 個の追加パラメータ </a>
<br>

<a href="<webflow:createWebflowURL event='link.yo' httpsInd='https' />">常に
HTTPS を使う </a>
<br>

<a href="<webflow:createWebflowURL event='link.yo' httpsInd='http' />">常に HTTP
を使う </a>
<br>

<a href="<webflow:createWebflowURL event='link.yo' httpsInd='calculate'
/>">HTTPS を推定 </a>
<br>
```

第4章 ナビゲーション (Webflow) 用 JSP タグ

```
<a href="<webflow:createWebflowURL event='link.yo' encode='false' />">URL をエン  
コードしない</a>  
<br>
```

```
<a href="<webflow:createWebflowURL event='link.yo' doRedirect='true' />">転送では  
なくリダイレクト</a>  
<br>
```

<webflow:createResourceURL>

<webflow:createResourceURL> タグ (表 4-2) は JSP 内で使われ、アプリケーションの WEB-INF/web.xml ファイルにある P13N_STATIC_ROOT コンテキストパラメータの値を使ってリソースの静的 URL を作成します。このタグは、GIF 画像を別のサーバからロードするのに使われることもあります。

| | |
|-----------------|---|
| タグライブラリ | webflow.tld |
| インポート文 | <%@ taglib uri="webflow.tld" prefix="webflow" %> |
| 実装されるクラス | CreateResourceURLTag |

表 4-2 は、<webflow:createResourceURL> タグの属性についての説明です。

表 4-2 <webflow:createResourceURL>

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|----------|-----|---------|--|-----|
| encode | いいえ | Boolean | URLをコード化するかどうかを指定する。有効な値は true と false。デフォルト値は true。 | R |
| resource | いいえ | String | ファイルまたは画像への相対パス。 | R |

例

コードリスト 4-2 は、`<webflow:createResourceURL>` JSP タグを使って特定のリソース、この場合は .gif ファイルをポイントする方法を示しています。

コードリスト 4-2 `<webflow:createResourceURL>` タグの使用例

```
" border="0" alt="チェック
アウトに進む" border="0"></a>
```

フォーム タグ

この節では、フォーム アクションへの簡単な動的リンクを作成するのに使う Webflow JSP フォーム タグについて説明します。

注意： 以降の表の「必須」列では、属性が必須と省略可能のどちらであることを「はい」または「いいえ」で示します。「R/C」列では、「C」は属性がコンパイル時表現であることを、「R」は属性が要求時表現とコンパイル時表現のいずれかであることを示します。

`<webflow:form>`

`<webflow:form>` タグは JSP 内で使われ、HTML フォーム タグを動的に生成します。このタグは `<webflow:validatedForm>` タグほど複雑ではなく、もっと単純です。

| | |
|----------|---|
| タグライブラリ | webflow.tld |
| インポート文 | <pre><%@ taglib uri="webflow.tld" prefix="webflow" %></pre> |
| 実装されるクラス | WebflowFormTag |

注意: このタグは、`<webflow:validatedForm>` とは異なり、埋め込み Webflow フォーム タグをサポートしていません。

表 4-3 は、`<webflow:form>` タグの属性についての説明です。

表 4-3 `<webflow:form>`

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------------------------|-----|--------|--|-----|
| <code>domainName</code> | いいえ | String | URL のドメイン名や IP アドレスを変更するのに使う。Webflow が別のマシン上のプロキシ サーバを経由する場合に、この属性を使用できる。 | R |
| <code>doRedirect</code> | いいえ | String | WebflowServlet にプレゼンテーション ノードへの転送ではなくリダイレクトを行わせる。有効な値は <code>true</code> と <code>false</code> 。デフォルトは <code>false</code> (リダイレクトしない)。 | R |
| <code>encode</code> | いいえ | String | Webflow 側で URL をコード化するように指定する。セッションの状態を保持したいが、ブラウザがクッキーを受け付けない場合には、URL をコード化する必要がある。有効な値は <code>true</code> と <code>false</code> 。デフォルト値は <code>true</code> 。ただし、ブラウザがクッキーを受け付けない場合にしか、URL をコード化する必要はない。 | R |
| <code>event</code> | はい | String | Webflow は、これを起点と組み合わせて使って、指定されたネームスペース XML ファイル内の遷移先を決定する。 | R |
| <code>extraParams</code> | いいえ | String | リクエスト パラメータを名前と値のペアとして追加指定するのに使われる。 | R |
| <code>hide</code> | いいえ | String | <code>false</code> に設定した場合、ネームスペース、起点、およびイベントは、隠しフォーム フィールドとしてではなく、コマンドライン上に表示される。有効な値は <code>true</code> と <code>false</code> 。デフォルト値は <code>true</code> 。 | R |

表 4-3 <webflow:form> (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-----------|-----|--------|--|-----|
| httpsInd | いいえ | String | <p>Webflow 側でプロトコルを推定するか、あるいは HTTPS または HTTP を使うように指定する。このため、有効な値は calculate、http、および https。デフォルト値は http。</p> <p>Calculate を指定すると、起点/イベントの分岐連鎖のリストに含まれているいずれかのノードが、アプリケーションの WEB-INF/web.xml ファイル内の HTTPS_URL_PATTERNS コンテキストパラメータで指定されているパターンと一致する場合には、HTTPS が選択される。Calculate のほうがより動的で処理コストがかかるが、プロトコルを強制する必要がある場合には、それをここで指定することができる。</p> | R |
| method | いいえ | String | <p>フォーム用に使われるメソッド。有効な値は get と post。デフォルト値は post。</p> | R |
| name | いいえ | String | <p>フォームの名前。</p> | R |
| namespace | いいえ | String | <p>起点およびイベントが定義されている Webflow コンフィグレーション ファイルを示す。省略されている場合は、現在の (最後に成功した) ネームスペースが使われる。</p> | R |
| origin | いいえ | String | <p>イベントの出発点となるノード (起点) で、<ノード名>.<ノードタイプ>の形式に従う。ページ名と同じ場合も異なる場合もある。省略された場合、JSP ページ名が使われる。</p> | R |

表 4-3 `<webflow:form>` (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------------------|-----|--------|---|-----|
| <code>pathPrefix</code> | いいえ | String | パス情報の接頭辞として使われる。この文字列は、Web アプリケーション URI の前に置かれる。プロキシ サーバが同一のマシン上にある場合に使われる。 注意: プロキシ サーバ側では、このパス接頭辞を取り除いてから Webflow にリクエストを転送する必要がある。 | R |
| <code>pathSuffix</code> | いいえ | String | パス情報の接尾辞として使われる。WebflowServlet URI に続けて、追加のパス情報が置かれる。これにより、この情報は、 <code>request.getPathInfo()</code> メソッドから取得可能になる。 | R |

例

コード リスト 4-3 に、`<webflow:form>` JSP タグの使用方法を示します。

コード リスト 4-3 `<form>` の使用例

```
<webflow:form event="button.go" >
  <input type="text" name="username" >
</webflow:form>
```

検証機能付きフォーム タグ

Webflow の検証機能付きフォーム JSP タグは、検証可能な HTML フォームを動的に生成するのに使います。これらのタグは、入力プロセッサおよび `ValidatedValues` クラスと組み合わせられて機能します。詳細については、『*Javadoc*』を参照してください。Web サイトへの訪問者が無効な情報を入力すると、訪問者の入力はいったん保存され、適切なエラー メッセージを付けて再表示されます。

<webflow:validatedForm>

<webflow:validatedForm> タグは JSP 内で使われ、HTML フォーム内でアクションを定義する URL を動的に作成します。このタグは `com.bea.p13n.appflow.webflow.forms.*` パッケージ、および `webflow.tld` ファイル内で定義されている、その他のネストされたフォーム タグ（この節の後の方で説明）との組合せで使われるものです。

| | |
|-----------------|--|
| タグライブラリ | <code>webflow.tld</code> |
| インポート文 | <pre><%@ taglib uri="webflow.tld" prefix="webflow" %></pre> |
| 実装されるクラス | <code>WebflowValidatedFormTag</code> <code>WebflowValidatedFormTagExtralInfo</code> |

表 4-4 は、<webflow:validatedForm> タグの属性についての説明です。

表 4-4 <webflow:validatedForm>

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------|-----|--------|--|-----|
| applyStyle | いいえ | String | フィールドのステータスに関連付けられている CSS スタイルをメッセージまたはフィールドに適用するか、あるいはそのいずれにも適用しない。このため、有効な値は message、field、および none。デフォルト値は message。 関連項目： invalidStyle styleId validStyle unspecifiedStyle | R |
| domainName | いいえ | String | URL のドメイン名や IP アドレスを変更するのに使う。Webflow が別のマシン上のプロキシ サーバを経由する場合に、この属性を使用できる。 | R |
| doRedirect | いいえ | String | WebflowServlet にプレゼンテーションノードへの転送ではなくリダイレクトを行わせる。有効な値は true と false。デフォルトは false (リダイレクトしない)。 | R |
| encode | いいえ | String | Webflow 側で URL をコード化するように指定する。セッションの状態を保持したいが、ブラウザがクッキーを受け付けない場合には、URL をコード化する必要がある。有効な値は true と false。デフォルト値は true。ただし、ブラウザがクッキーを受け付けない場合にしか、URL をコード化する必要はない。 | R |
| event | はい | String | Webflow は、これを起点と組み合わせて使って、指定されたネームスペース XML ファイル内の遷移先を決定する。 | R |
| extraParams | いいえ | String | リクエスト パラメータを名前と値のペアとして追加指定するのに使われる。 | R |

表 4-4 <webflow:validatedForm> (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------------|-----|--------|---|-----|
| hide | いいえ | String | false に設定した場合、ネームスペース、起点、およびイベントは、隠しフォームフィールドとしてではなく、コマンドライン上に表示される。 有効な値は true と false。デフォルト値は true。 | R |
| httpsInd | いいえ | String | Webflow 側でプロトコルを推定するか、あるいは HTTPS または HTTP を使うように指定する。このため、有効な値は calculate、http、および https。デフォルト値は http。 Calculate を指定すると、起点/イベントの分岐連鎖のリストに含まれているいずれかのノードが、アプリケーションの WEB-INF/web.xml ファイル内の HTTPS_URL_PATTERNS コンテキストパラメータで指定されているパターンと一致する場合には、HTTPS が選択される。Calculate のほうがより動的で処理コストがかかるが、プロトコルを強制する必要がある場合には、それをここで指定することができる。 | R |
| invalidStyle | いいえ | String | HTML フィールドが無効な場合に、そのフィールドまたはメッセージを書式化するのに使用される CSS スタイル。 | R |
| messageAlign | いいえ | String | エラーメッセージの表示位置をフィールドの上、フィールドの右、フィールドの下のいずれかに指定する。このため、有効な値は top、right、および bottom。デフォルト値は right。 | R |
| method | いいえ | String | フォーム用に使われるメソッド。有効な値は get と post。デフォルト値は post。 | R |
| name | いいえ | String | フォームの名前。 | R |

表 4-4 <webflow:validatedForm> (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------------------|-----|--------|---|-----|
| namespace | いいえ | String | 起点およびイベントが定義されている Webflow コンフィグレーション ファイルを示す。省略されている場合は、現在の (最後に成功した) ネームスペースが使われる。 | R |
| origin | いいえ | String | イベントの出発点となるノード (起点) で、<ノード名>.<ノードタイプ>の形式に従う。ページ名と同じ場合も異なる場合もある。省略された場合、JSP ページ名が使われる。 | R |
| pathPrefix | いいえ | String | パス情報の接頭辞として使われる。この文字列は、Web アプリケーション URI の前に置かれる。プロキシ サーバが同一のマシン上にある場合に使われる。 注意: プロキシ サーバ側では、このパス接頭辞を取り除いてから Webflow にリクエストを転送する必要がある。 | R |
| pathSuffix | いいえ | String | パス情報の接尾辞として使われる。WebflowServlet URI に続けて、追加のパス情報が置かれる。これにより、この情報は、request.getPathInfo() メソッドから取得可能になる。 | R |
| styleId | いいえ | String | フィールドのステータスが valid、invalid、unspecified のいずれであるかに応じて、invalidStyle、unspecifiedStyle、validStyle のいずれか 1 つに設定されるスクリプト変数。HTML フォームの書式化をより細かく制御するのに使用できる。 | R |
| unspecifiedStyle | いいえ | String | HTML フィールドの検証前の初期 CSS スタイルを指定するのに使われる。 | R |
| validStyle | いいえ | String | 有効な HTML フィールドを書式化するのに使用される CSS スタイル。 | R |

<webflow:text>

<webflow:text> タグは JSP 内で使われ、HTML テキスト フィールドを検証します。このタグは、<webflow:validatedForm> タグ内にネストする必要があります。

| | |
|----------|---|
| タグライブラリ | webflow.tld |
| インポート文 | <%@ taglib uri="webflow.tld" prefix="webflow" %> |
| 実装されるクラス | TexTag |

表 4-5 は、<webflow:text> タグの属性についての説明です。

表 4-5 <webflow:text>

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|----------------|-----|--------|--|-----|
| htmlAttributes | いいえ | String | 追加の HTML 属性。直接サポートされていない属性をここで提供する。 | R |
| maxLength | いいえ | String | テキスト フィールドの最大長。 | R |
| name | はい | String | テキスト フィールドの名前。 | R |
| retainValue | いいえ | String | テキスト フィールドが再表示される時に入力内容も再表示するかどうかを指定する。有効な値は true と false。 | R |
| size | いいえ | String | テキスト フィールドのサイズ。 | R |
| style | いいえ | String | そのテキスト フィールドに関連付けられている HTML CSS スタイル。 | R |
| value | いいえ | String | テキスト フィールドの初期値。 | R |

<webflow:password>

<webflow:password> タグは <webflow:text> タグと同様ですが、フィールドの入力内容がアスタリスク (****) でマスクされます。このタグは、<webflow:validatedForm> タグ内にネストする必要があります。

| | |
|-----------------|---|
| タグライブラリ | webflow.tld |
| インポート文 | <%@ taglib uri="webflow.tld" prefix="webflow" %> |
| 実装されるクラス | PasswordTag |

表 4-6 <webflow:password>

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|----------------|-----|--------|--|-----|
| htmlAttributes | いいえ | String | 追加の HTML 属性。直接サポートされていない属性をここで提供する。 | R |
| maxLength | いいえ | String | パスワード フィールドの最大長。 | R |
| name | はい | String | パスワード フィールドの名前。 | R |
| retainValue | いいえ | String | パスワード フィールドが再表示されるときに入力内容も再表示するかどうかを指定する。有効な値は true と false。デフォルト値は false。 | R |
| size | いいえ | String | パスワード フィールドのサイズ。 | R |
| style | いいえ | String | そのパスワード フィールドに関連付けられている HTML CSS スタイル。 | R |
| value | いいえ | String | パスワード フィールドの初期値。 | R |

<webflow:radio>

<webflow:radio> タグは JSP 内で使われ、HTML ラジオ ボタンを表しますが、入力内容を保存します。このタグは、<webflow:validatedForm> タグ内にネストする必要があります。

| | |
|----------|---|
| タグライブラリ | webflow.tld |
| インポート文 | <%@ taglib uri="webflow.tld" prefix="webflow" %> |
| 実装されるクラス | RadioTag |

表 4-7 は、<webflow:radio> タグの属性についての説明です。

表 4-7 <webflow:radio>

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|----------------|-----|--------|---|-----|
| checked | いいえ | String | 初期状態でラジオ ボタンが選択されているかどうかを指定する。有効な値は true と false。デフォルト値は false。 | R |
| htmlAttributes | いいえ | String | 追加の HTML 属性。直接サポートされていない属性をここで提供する。 | R |
| name | はい | String | ラジオ ボタン フィールドの名前。 | R |
| value | はい | String | ラジオ ボタン フィールドの初期値。 | R |

<webflow:checkbox>

<webflow:checkbox> タグは JSP 内で使われ、チェックボックスに必要な HTML を生成します。このタグは、InputProcessor が InvalidFormFieldException を送出したときに、入力内容を保存します。このタグは、<webflow:validatedForm> タグ内にネストする必要があります。

| | |
|---------|-------------|
| タグライブラリ | webflow.tld |
|---------|-------------|

| | |
|----------|---|
| インポート文 | <code><%@ taglib uri="webflow.tld" prefix="webflow" %></code> |
| 実装されるクラス | CheckboxTag |

表 4-8 は、`<webflow:checkbox>` タグの属性についての説明です。

表 4-8 `<webflow:checkbox>`

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|----------------|-----|--------|--|-----|
| checked | いいえ | String | 初期状態でチェックボックス フィールドが選択されているかどうかを指定する。有効な値は true と false。デフォルト値は false。 | R |
| htmlAttributes | いいえ | String | 追加の HTML 属性。直接サポートされていない属性をここで提供する。 | R |
| name | はい | String | チェックボックス フィールドの名前。 | R |
| value | はい | String | チェックボックス フィールドの初期値。 | R |

`<webflow:textarea>`

`<webflow:textarea>` タグは JSP 内で使われ、HTML テキスト領域を表しますが、入力内容を保存します。このタグは、`<webflow:validatedForm>` タグ内にネストする必要があります。

| | |
|----------|---|
| タグライブラリ | webflow.tld |
| インポート文 | <code><%@ taglib uri="webflow.tld" prefix="webflow" %></code> |
| 実装されるクラス | TextareaTag |

表 4-9 は、`<webflow:textarea>` タグの属性についての説明です。

表 4-9 `<webflow:textarea>`

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------------------------|-----|--------|--|-----|
| <code>cols</code> | いいえ | String | テキスト領域のカラム数。 | R |
| <code>name</code> | はい | String | テキスト領域の名前。 | R |
| <code>retainValue</code> | いいえ | String | テキスト領域が再表示されるときに入力内容も再表示するかどうかを指定する。有効な値は <code>true</code> と <code>false</code> 。 | R |
| <code>rows</code> | いいえ | String | テキスト領域の行数。 | R |
| <code>style</code> | いいえ | String | そのテキスト領域に関連付けられている HTML CSS スタイル。 | R |
| <code>value</code> | いいえ | String | テキスト領域の初期値。 | R |
| <code>wrap</code> | いいえ | String | テキスト領域で入力内容を右端で折り返すかどうかを指定する。有効な値は <code>true</code> と <code>false</code> 。デフォルト値は <code>true</code> 。 | R |

`<webflow:select>`

`<webflow:select>` タグは JSP 内で使われ、リスト ボックスを表しますが、入力内容を保存します。このタグは、`<webflow:validatedForm>` タグ内にネストする必要があります。

| | |
|----------|---|
| タグライブラリ | <code>webflow.tld</code> |
| インポート文 | <pre><%@ taglib uri="webflow.tld" prefix="webflow" %></pre> |
| 実装されるクラス | <code>SelectTag</code> |

表 4-10 は、`<webflow:select>` タグの属性についての説明です。

表 4-10 `<webflow:select>`

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-----------------------------|-----|--------|---|-----|
| <code>htmlAttributes</code> | いいえ | String | 追加の HTML 属性。直接サポートされていない属性をここで提供する。 | R |
| <code>multiple</code> | いいえ | String | リスト ボックスで複数選択を可能にするかどうかを指定する。有効な値は <code>true</code> と <code>false</code> 。デフォルト値は <code>false</code> 。 | R |
| <code>name</code> | はい | String | リスト ボックスの名前。 | R |
| <code>size</code> | いいえ | String | リスト ボックスのサイズ。 | R |
| <code>style</code> | いいえ | String | HTML CSS スタイル属性。 | R |

`<webflow:option>`

`<webflow:option>` タグ (表 4-11) は JSP 内で使われ、HTML オプションを表しますが、入力内容を保存します。オプションはリスト ボックス内の 1 つの値です。このタグは、`<webflow:select>` タグ内にネストする必要があります。

| | |
|----------|---|
| タグライブラリ | <code>webflow.tld</code> |
| インポート文 | <code><%@ taglib uri="webflow.tld" prefix="webflow" %></code> |
| 実装されるクラス | <code>OptionTag</code> |

表 4-11 は、`<webflow:option>` タグの属性についての説明です。

表 4-11 `<webflow:option>`

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-----------------------|-----|--------|--|-----|
| <code>selected</code> | いいえ | String | 初期状態でそのオプションが選択されているかどうかを指定する。有効な値は <code>true</code> と <code>false</code> 。デフォルト値は <code>false</code> 。 | R |

表 4-11 <webflow:option> (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------|-----|--------|--------------|-----|
| style | いいえ | String | HTML スタイル属性。 | R |
| value | はい | String | そのオプションが表す値。 | R |

例

コードリスト 4-4 では、それぞれの検証機能付きフォーム タグを、Web サイトの訪問者から情報を入手する HTML フォーム ページで使っています。

コードリスト 4-4 webflow.tld フォーム タグの使用例

```
<%@ taglib uri="webflow.tld" prefix="webflow" %>

<% String validStyle="background: white; color: black; font-family: Arial"; %>
<% String invalidStyle="background: white; color: red; font-style: italic"; %>

<!-- InvalidFormException が送出されたらメッセージを表示する --%>

<font size="5" color="green"><webflow:getException/></font>
<br>

<webflow:validatedForm event="button.go" applyStyle="message"
  messageAlign="right" validStyle="<%=validStyle%>"
  invalidStyle="<%=invalidStyle%>" unspecifiedStyle="<%=validStyle%>" >
<p>

ユーザ名 :
<webflow:text name="username" value="start" size="15" maxlength="30"
htmlAttributes="onMouseOver=\\\"self.status='User ID';return true\\\" " />
<br>

パスワード :
<webflow:password name="password" size="15" retainValue="true" maxlength="30"
  htmlAttributes="onMouseOver=\\\"self.status='Secret Password';return true\\\" " />
<br>

性別 :
<webflow:radio name="sex" checked="true" value="M"/> 男
<webflow:radio name="sex" value="F" /> 女
<br>

好きなペット (複数選択可) :
<webflow:checkbox name="cat" value="cat" /> 猫 <br>
<webflow:checkbox name="dog" checked="true" value="dog" /> 犬 <br>
<webflow:checkbox name="bird" value="bird" /> 鳥
<p>
```

コメント :

```
<webflow:textarea name="comment" cols="20" rows="3" value="hello" />
<br>
```

趣味 (複数選択可) :

```
<webflow:select name="hobbies" size="3" multiple="true">
  <webflow:option value="Running" /> ランニング
  <webflow:option value="Skiing" /> スキー
  <webflow:option value="Motocross" /> バイク
  <webflow:option value="Rugby" /> ラグビー
</webflow:select>
<br>
<input type="submit" name="Submit" />
</webflow:validatedForm>
```

Pipeline セッション タグ

Pipeline セッションは、入力プロセッサ、Pipeline コンポーネント、プレゼンテーション ノードの間で情報を共有するのに使います。Pipeline セッション JSP タグは、Pipeline セッションで、プロパティを取得したり設定したりするのに使います。プレゼンテーション ノード (JSP など) は、通常、Pipeline セッションから情報を取得するのに使い、入力プロセッサと Pipeline コンポーネントは、Pipeline セッションにプロパティを配置するのに使います。ただし、Pipeline セッションでプロパティを設定するための JSP タグがあります。

<webflow:setProperty>

`<webflow:setProperty>` タグは、Pipeline セッション内でプロパティを設定します。

| | |
|----------|---|
| タグ ライブラリ | webflow.tld |
| インポート文 | <pre><%@ taglib uri="webflow.tld" prefix="webflow" %></pre> |

| | |
|----------|----------------|
| 実装されるクラス | setPropertyTag |
|----------|----------------|

表 4-12 は、`<webflow:setProperty>` タグの属性についての説明です。

表 4-12 `<webflow:setProperty>`

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-----------|-----|--------|---|-----|
| namespace | いいえ | String | namespace 属性を使用して、特別な起点およびイベントを定義している特定の webflow コンフィグレーション ファイルを webflow に強制的に使用させる。省略されている場合は、現在の (最後に成功した) ネームスペースが使われる。 | R |
| property | はい | String | 特定のプロパティが関連付けられている名前またはキー。 | R |
| scope | いいえ | String | プロパティのスコープ。有効な値は session と request。デフォルト値は session。 | R |
| value | はい | Object | プロパティに関連付ける値。オブジェクト名または Java 式として指定される。 | R |

例

コード リスト 4-5 では、`<webflow:setProperty>` JSP タグを使って、Pipeline セッション (Request スコープ) 内で任意のオブジェクトを設定する方法を示しています。

コード リスト 4-5 `<webflow:setProperty>` タグの使用例

```
<% SomeObject so = new SomeObject("TWO"); %>
<webflow:setProperty property="myobject" value="<%= so %>"
scope="request" />
```

<webflow:getProperty>

<webflow:getProperty> タグ (表 4-13) は、Pipeline セッションから名前の付いたプロパティを取得します。このタグは、インラインに埋め込むこともでき、スクリプト変数を返すようにもできます。

| | |
|-----------------|---|
| タグライブラリ | webflow.tld |
| インポート文 | <%@ taglib uri="webflow.tld" prefix="webflow" %> |
| 実装されるクラス | GetPropertyTag |

表 4-13 は、<webflow:getProperty> タグの属性についての説明です。

表 4-13 <webflow:getProperty>

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-----------|-----|--------|--|-----|
| namespace | いいえ | String | namespace 属性を使用して、特別な起点およびイベントを定義している特定の webflow コンフィグレーション ファイルを webflow に強制的に使用させる。省略されている場合は、現在の (最後に成功した) ネームスペースが使われる。 | R |
| id | いいえ | Object | 返されたオブジェクトのインスタンスを受け取る Java スクリプト変数。省略された場合、オブジェクト上で toString() メソッドが呼ばれ、結果がブラウザに表示される。 | R |
| property | はい | String | Pipeline セッションから取得するプロパティの名前またはキー。 | R |
| scope | いいえ | String | プロパティの範囲。request と session のいずれか。Request スコープのプロパティを使うと、複製の必要がないので、特にクラス内で性能を向上させることが可能。有効な値は session と request。デフォルト値は session。 | R |

表 4-13 `<webflow:getProperty>` (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------|-----|--------|---------------------------|-----|
| type | いいえ | String | 例外をキャストするのに使える Java クラス名。 | R |

例 1

以下のコード例では、`<webflow:getProperty>` JSP タグをインラインで使っています。SomeObject のインスタンス上で toString() メソッドが呼ばれます。

```
result = <webflow:getProperty property="myobject" scope="request" />
```

例 2

26 ページのコードリスト 4-6 では、`<webflow:getProperty>` JSP タグを使って SomeObject 型のスクリプト変数を返しています。

コード リスト 4-6 `<webflow:getProperty>` タグの使用例 2

```
<webflow:getProperty id="myObj" property="myobject"
type="com.bea.test.SomeObject" scope="request" />
result = <%= myObj.getValue() %>
```

`<webflow:setValidatedValue>`

`<webflow:setValidatedValue>` タグ (表 4-14) は、JSP 内で、Web サイト訪問者が修正する必要のあるフォーム内のフィールドの表示をコンフィグレーションするのに使用します。通常、この設定は入力プロセッサ内で行いますが、このタグを使って JSP で設定することもできます。`<webflow:setValidatedValue>` タグは、`<webflow:getValidatedValue>` タグと組み合わせて使われます。

タグ ライブラリ

webflow.tld

| | |
|----------|---|
| インポート文 | <code><%@ taglib uri="webflow.tld" prefix="webflow" %></code> |
| 実装されるクラス | <code>setValidatedValueTag</code> |

注意： `<webflow:validatedForm>` タグは以前のリリースの `validatedValues` クラスをサポートするので、`<webflow:setValidatedValue>` の代わりに使用することを考慮するとよいでしょう。ただし、下位レベル機能にアクセスする必要がある場合は、ここで紹介するタグも有効です。

表 4-14 は、`<webflow:setValidatedValue>` タグの属性についての説明です。

表 4-14 `<webflow:setValidatedValue>`

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------------------------|-----|--------|--|-----|
| <code>fieldName</code> | いいえ | String | ステータスが要求されるフィールドの名前。HTML フォームのフィールド名と一致する必要がある。 | R |
| <code>fieldStatus</code> | いいえ | String | フィールドの処理ステータス。有効な値は以下のいずれか。 <code>unspecified</code> (未指定) フィールドは空欄。Web サイト訪問者が何かデータを入力する必要がある。 <code>invalid</code> (無効) 入力されたデータが不正。 <code>valid</code> (有効) 正しいデータを入力済み。 | R |
| <code>fieldValue</code> | いいえ | String | フィールドの新しい値。 | R |

例

このサンプル コードを JSP 内で使うと、`<field_name>` というフォーム フィールドの現在値と処理ステータスが取得されます。

```
<webflow:setValidatedValue fieldName="<field_name>"
fieldValue="<field_value>" fieldStatus="status" />
```

<webflow:getValidatedValue>

<webflow:getValidatedValue> タグは、JSP 内で使われ、Web サイト訪問者が修正する必要のあるフォーム内のフィールドを表示します。

<webflow:getValidatedValue> タグは、<webflow:setValidatedValue> タグと組み合わせて使われます。

| | |
|----------|---|
| タグライブラリ | webflow.tld |
| インポート文 | <%@ taglib uri="webflow.tld" prefix="webflow" %> |
| 実装されるクラス | getValidatedValueTag |

注意： <webflow:validatedForm> タグは以前のリリースの ValidatedValues クラスをサポートするので、<webflow:getValidatedValue> の代わりに使用することを考慮するとよいでしょう。ただし、下位レベル機能にアクセスする必要がある場合は、ここで紹介するタグも有効です。

表 4-15 は、<webflow:getValidatedValue> タグの属性についての説明です。

表 4-15 <webflow:getValidatedValue>

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------------------|-----|--------|---|-----|
| fieldColor | いいえ | String | ステータスに応じて、invalidColor、validColor、unspecifiedColor のいずれかに設定されるスクリプト変数。フィールドやメッセージの色を変えるのに使うことが可能。 | R |
| fieldDefault Value | いいえ | String | fieldValue がいないときに使用されるデフォルト値。 | R |
| fieldMessage | いいえ | String | 現在のフィールドに特定のメッセージを提供するのに使うスクリプト変数。 | R |
| fieldName | はい | String | ステータスが要求されるフィールドの名前。HTML フォームのフィールド名と一致する必要がある。 | R |

表 4-15 <webflow:getValidatedValue> (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------------------|-----|--------|---|-----|
| fieldStatus | はい | String | フィールドのステータス。有効な値は以下のいずれか。 unspecified (未指定) フィールドは空欄。Web サイト訪問者が何かデータを入力する必要がある。 invalid (無効) 入力されたデータが不正。 valid (有効) 正しいデータを入力済み。 | R |
| fieldValue | はい | String | フォーム フィールドの値を表すスクリプト変数。 | R |
| invalidColor | いいえ | String | 無効なフィールドを示すラベルに使う色。デフォルトは red。 | R |
| unspecifiedColor | いいえ | String | Web サイト訪問者が必須フィールドを空欄のまま残した場合、そのフィールドのラベルがこの色になる。デフォルトは red。 | R |
| validColor | いいえ | String | 有効なフィールドを示すラベルに使う色。デフォルトは black。 | R |

これらのフィールドは、入力プロセッサによって検証作業のあとで決定され、マークが付けられます。あらゆる `InputProcessor` は `ValidatedValues` オブジェクトを使って、無効と決定されたフィールドだけでなく、どのフィールドが正常に処理されたかも伝達します。詳細については、『*Javadoc*』を参照してください。

例 1

このサンプル コードを JSP 内で使うと、<field_name> というフォーム フィールドの現在値と処理ステータスが取得されます。

```
<webflow:getValidatedValue fieldName="<field_name>"
fieldValue="<field_value>" fieldStatus="status" />
```

例 2

`<webflow:getValidatedValue>` タグは、`webflow.tld` タグ ライブラリを参照して、使用可能な要素や属性を取得します。この例では、HTTP セッションから以下の値を取得するリクエストが行われています。

```
fieldName
fieldValue
fieldStatus
validColor
invalidColor
unspecifiedColor
fieldColor
```

これらの属性は表示目的で使われます。この例では、そのフィールドが必須であることを示し、赤くマーキングしています。コードリスト 4-7 に示すように、最終目標は、入力プロセッサから返されたときにどの部分が有効または無効かを Web サイト訪問者に対して示すことです。

コード リスト 4-7 `<webflow:getValidatedValue>` タグの使用例 2

```
<webflow:getValidatedValue
fieldName="<%=HttpRequestConstants.CUSTOMER_FIRST_NAME%>"
fieldValue="customerFirstName" fieldStatus="status"
validColor="black"
invalidColor="red" unspecifiedColor="black" fieldColor="fontColor"
/>
```

`<webflow:getException>`

`<webflow:getException>` タグは、Webflow プロセッサにより送出された例外やメッセージを取得するのに使います。これは `InvalidFormFieldException` 例外か、`ProcessingException` 例外に関連付けられたメッセージです。このタグはインラインに埋め込むことも（その場合、例外上で `getMessage()` メソッドを呼びます）、例外を表すスクリプト変数を返すようにもできます。

| | |
|----------|---|
| タグライブラリ | webflow.tld |
| インポート文 | <%@ taglib uri="webflow.tld" prefix="webflow" %> |
| 実装されるクラス | getExceptionTag |

表 4-16 は、<webflow:getException> タグの属性についての説明です。

表 4-16 <webflow:getException> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------|-----|-----------|-----------------------------------|-----|
| id | はい | Exception | 例外のインスタンスを取り出すのに使える Java スクリプト変数。 | R |
| type | いいえ | String | 例外をキャストするのに使える Java クラス名。 | R |

例

コードリスト 4-8 に、<webflow:getException> JSP タグの使用方法を示します。

コードリスト 4-8 <webflow:getException> タグの使用例

```
<%-- InvalidFormDateException が送出されたらメッセージを表示する --%>
<font size="5" color="red"><webflow:getException/></font>
```

第5章 カタログ開発用 JSP タグ

WebLogic Portal 製品スイートに含まれている Commerce サービス には、使用頻度の高い Web ベースのプロダクト カタログ機能を実装する JavaServer Page (JSP) テンプレートと JSP タグが用意されています。プロダクト カタログ用の JSP テンプレートにより、顧客は商品アイテムを検索したり、カテゴリをたどってアイテムにたどり着くことができます。JSP タグは、こうした機能の実装に使用します。

このトピックは、以下の節で構成されています。

- [カタログ用 JSP タグ](#)
- [e ビジネス用 JSP タグ](#)

カタログ用 JSP タグ

この節では、プロダクト カタログ用 JSP タグ ライブラリを構成するタグをまとめています。これらのタグは、デフォルトの Product Catalog を構成する JSP テンプレートで使われます。必要な書式に合わせて、JSP テンプレートの使用時にタグを追加したり、削除したりできます。

注意： 以降の表の「必須」列では、属性が必須と省略可能のどちらであるかを「はい」または「いいえ」で示します。「R/C」列では、「C」は属性がコンパイル時表現であることを、「R」は属性が要求時表現とコンパイル時表現のいずれかであることを示します。

<catalog:getProperty>

| | |
|----------|--|
| タグライブラリ | cat.tld |
| インポート文 | <%@ taglib uri="cat.tld" prefix="catalog" %> |
| 実装されるクラス | GetPropertyTag GetPropertyTagExtraInfo |

表示用のプロパティを `ProductItem` または `Category` から読み込むには、`<catalog:getProperty>` タグ (表 5-1) を使います。プロパティは、明示的プロパティ (Catalog 項目で `get` メソッドを使って読み込めるプロパティ) または暗黙的プロパティ (Catalog 項目で `ConfigurableEntity` `getProperty` メソッドを介して利用できるプロパティ) のどちらでもかまいません。このタグは、まず指定のプロパティを明示的プロパティとして読み込めるかをチェックします。読み込めない場合は、指定のプロパティを暗黙的プロパティとして読み込みます。

表 5-1 は、`<catalog:getProperty>` タグの属性についての説明です。

表 5-1 `<catalog:getProperty>` タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-----------------------------|-----|--------|---|-----|
| <code>getterArgument</code> | いいえ | String | <p>明示的プロパティ <code>getter</code> メソッドの引数として指定するオブジェクトへの参照を示す。</p> <p>暗黙的プロパティ、またはプロパティセットフレームワークを使用して定義されたカスタムプロパティを取り込むために使用する場合がある。その場合、<code>getterArgument</code> はプロパティセットのスコープ名となる (下の例 2 を参照)。オブジェクトは、<code><%=getterArgumentReference %></code> の形式で指定し、実行時表現にする必要がある。</p> | R |

表 5-1 <catalog:getProperty> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------------|-----|--------------|--|-----|
| id | いいえ | String | id="newInstance" id 属性を指定すると、id が割り当てられている変数名の中で、読み込んだプロパティの値が使える。id 属性を指定しないと、プロパティの値がインライン化される。 | C |
| object | はい | Catalog item | <%= objectReference %> の形式で指定される ProductItem または Category オブジェクトへの参照を示す。 | R |
| propertyName | はい | String | propertyName="propertyName" 読み込むプロパティの名前。明示的プロパティの場合は、表 5-2 に示すいずれかの値が使える。 | C |
| returnType | いいえ | String | returnType="returnType" id 属性を指定すると、id 属性で指定されている変数のデータ型が宣言される。 | C |

表 5-2 propertyName の値

| プロパティ名 | カタログ項目のタイプ |
|---|-----------------------------|
| "contributor coverage creationDate creator description image key language modifiedDate name publisher relation rights source" | Catalog Item (一般的なプロパティ) |
| "jsp" | Category |
| "availability currentPrice format jsp msrp shippingCode taxCode type visible" | ProductItem |

例 1

コードリスト 5-1 では、既存の `ProductItem` から詳細 JSP 情報を読み込んでいます。

コードリスト 5-1 `<catalog:getProperty>` タグの使用例 1

```
<%@ taglib uri="cat.tld" prefix="catalog" %>

<catalog:getProperty
object="<%= item %>"
  propertyName="Jsp"
  getterArgument=
    "<%= new Integer(ProductItem.DETAILED_DISPLAY_JSP_INDEX) %>"
    id="detailJspInfo"
returnType="com.beasys.commerce.ebusiness.catalog.JspInfo"
/>
```

例 2

以下の例では、次の特徴をもつプロパティ セットまたはスキーマに対し、`getterArgument` 属性を使って暗黙的プロパティまたはカスタム プロパティを取得する方法を示しています。

- 名前 :MyCatalog
- プロパティ名 :color

注意： `getterArgument` は実行時表現でなければならないため、`MyCatalog` を String 変数に割り当て、その変数を `getterArgument` に対する値として使用します。

コードリスト 5-2 `<catalog:getProperty>` タグの使用例 2

```
<%@ taglib uri="cat.tld" prefix="catalog" %>

<%
String myPropertySetName = "MyCatalog";
ProductItem myProductItem = .....; // ProductItem への参照
%>
<catalog:getProperty
  object="<%=myProductItem%>"
```

```

    propertyName="color"
    getterArgument="<%=myPropertySetName%>"
/>

```

<catalog:iterateViewIterator>

| | |
|----------|---|
| タグライブラリ | cat.tld |
| インポート文 | <%@ taglib uri="cat.tld" prefix="catalog" %> |
| 実装されるクラス | IterateViewIteratorTag IterateViewIteratorTagExtraInfo |

<catalog:iterateViewIterator> タグは、ViewIterator を通して反復するときに使います。ViewIterator は、一連の固定サイズの Views に分割された大規模なリモート データの集合に対するイテレータです。ViewIterators は、多数の ProductItems または Categories を返す可能性がある Catalog サービスのすべての API メソッドから返されます。このタグを使えば、一度に 1 つのアイテム (ProductItem または Category) に対して ViewIterator を反復することも、一度に View 全体 (固定サイズの ProductItems または Categories のセット) に対して反復することもできます。このタグは、処理の完了時に ViewIterator の状態をリセットしません。

表 5-3 は、<catalog:iterateViewIterator> タグの属性についての説明です。

表 5-3 <catalog:iterateViewIterator> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|----------|----|--------------|--|-----|
| id | はい | String | id="newInstance" id が割り当てられている変数名の中で、現在反復されているオブジェクトの値が使える。 | C |
| iterator | はい | ViewIterator | ViewIterator オブジェクトへの参照を示す。指定する形式は、<%= iteratorReference %>。 | R |

表 5-3 <catalog:iterateViewIterator> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|---------------|-----|--------|---|-----|
| iterateByView | いいえ | String | iterateByView="{true false}" View または Catalog 項目のどちらで ViewIterator を反復するかを指定す る。指定がないと、ViewIterator は Catalog 項目で反復する。 | C |
| returnType | いいえ | String | returnType="returnType" id 属性で指定している変数のデータ型を 宣言する。デフォルトは java.lang.Object。 iterateByView が true の場合、デー タ型は com.beasys.commerce.ebusiness. catalog.View とみなされる。 | C |

例 1

コードリスト 5-3 では、ViewIterator にあるすべての Categories のキーを表示しています。

コードリスト 5-3 <catalog:iterateViewIterator> タグの使用例 1

```
<%@ taglib uri="cat.tld" prefix="catalog" %>

<catalog:iterateViewIterator
  iterator="<%= myIterator %>"
  id="category"
  returnType="com.beasys.commerce.ebusiness.catalog.Category">
  <%= category.getKey().toString() %>
</catalog:iterateViewIterator>
```

例 2

以下の例では、ViewIterator に含まれているすべての Views を表示していま
す。

コードリスト 5-4 <catalog:iterateViewIterator> タグの使用例 2

```
<%@ taglib uri="cat.tld" prefix="catalog" %>

<catalog:iterateViewIterator
  iterator="<%= myIterator %>"
  id="view"
  returnType="com.beasys.commerce.ebusiness.catalog.ViewIterator"
  iterateByView="true">
  <%= view.toString() %>
</catalog:iterateViewIterator>
```

<catalog:iterateThroughView>

| | |
|----------|---|
| タグライブラリ | cat.tld |
| インポート文 | <%@ taglib uri="cat.tld" prefix="catalog" %> |
| 実装されるクラス | IterateThroughViewTag IterateThroughViewTagExtraInfo |

<catalog:iterateThroughView> タグ (表 5-4) により、指定の ViewIterator を View 内で反復することができます。View の最後に達するまで、一度に 1 つの Catalog 項目に対して View を反復します。反復対象の View の (インデックスによる) 指定がないと、ViewIterator の現在の View が使われます。このタグは、処理の完了時に ViewIterator の状態をリセットしません。

表 5-4 は、<catalog:iterateThroughView> タグの属性についての説明です。

表 5-4 <catalog:iterateThroughView> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------|----|--------|--|-----|
| id | はい | String | id="newInstance" id が割り当てられている変数名の中で、 現在反復されているオブジェクトの値が 使える。 | C |

表 5-4 <catalog:iterateThroughView> タグの属性 (続き)

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|------------|-----|--------------|---|-----|
| iterator | はい | ViewIterator | ViewIterator オブジェクトへの参照を示す。指定する形式は、 <%= iteratorReference %>。 | R |
| returnType | いいえ | String | returnType="returnType" id 属性で指定している変数のデータ型を宣言する。デフォルトは java.lang.Object。 | C |
| viewIndex | いいえ | Integer | 反復対象の View のインデックス (ViewIterator の先頭との相対位置) を指定する。参照されるオブジェクトは、 <%= viewIndexIntegerReference %> の形式で指定する必要がある。 | R |

例 1

コードリスト 5-5 では、指定した ViewIterator の現在の View に含まれるすべての ProductItems のキーを表示しています。

コードリスト 5-5 <catalog:iterateThroughView> タグの使用例 1

```
<%@ taglib uri="cat.tld" prefix="catalog" %>
<catalog:iterateThroughView
  iterator="<%= myIterator %>"
  id="item"
  returnType="com.beasys.commerce.ebusiness.catalog.ProductItem">
<%= item.getKey().toString() %>
</catalog:iterateThroughView>
```

例 2

以下の例では、指定した ViewIterator の最初の View に含まれるすべての ProductItems のキーを表示しています。

コードリスト 5-6 <catalog:iterateThroughView> タグの使用例 2

```
<%@ taglib uri="cat.tld" prefix="catalog" %>

<catalog:iterateThroughView
  iterator="<%= myIterator %>"
  id="item"
  returnType="com.beasys.commerce.ebusiness.catalog.ProductItem"
  viewIndex="new Integer(0)">
  <%= item.getKey().toString() %>
</catalog:iterateThroughView>
```

e ビジネス用 JSP タグ

この節では、e ビジネス用 JSP タグ ライブラリを構成するタグをまとめています。

<eb:smnav>

| | |
|----------|--|
| タグ ライブラリ | eb.tld |
| インポート文 | <%@ taglib uri="eb.tld" prefix="eb" %> |
| 実装されるクラス | ScrollableModelTag |

スクロール可能なモデルを使用して、値オブジェクトを取得する際、表示されているところだけが取り込まれるようにします。<eb:smnav> タグ (表 5-5) を使用すると、値オブジェクトの一覧に含まれる要素の表示方法が制御でき、その前後のページへリンクできます。

接頭辞「<eb:」は、e-business (e ビジネス) の略です。スクロール可能なモデルを e ビジネス パッケージ全体で使用して、オブジェクトの一覧を通して反復させることができます。トランザクションやショッピング カート、注文履歴、発送サービスと組み合わせて使用できます。

このタグは、`PipelineSessionConstants.SCROLLABLE_MODEL` キーにある `ScrollableModel` オブジェクトを含む Pipeline セッションに依存しています。

表 5-5 は、`<eb:smnav>` タグの属性についての説明です。

表 5-5 `<eb:smnav>` タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-------------------------|-----|--------|---|-----|
| <code>event</code> | いいえ | String | 訪問者が [次] または [前] をクリックすることにより、Webflow のなかでコンフィグレーションが可能なリンク名。 | C |
| <code>nextstring</code> | いいえ | String | [次] に対応するローカライズ名。“>” のように単純なものでもよい。 | C |
| <code>origin</code> | いいえ | String | 現在の JSP ページ。 | C |
| <code>pageindex</code> | いいえ | String | 表示するページのインデックス。 | R |
| <code>prevstring</code> | いいえ | String | [前] に対応するローカライズ名。“<” のように単純なものでもよい。 | C |

例

Commerce サービスの JSP テンプレートの一部である `orderhistory.jsp` により、訪問者は注文内容をページからページへとたどることができます。一度に表示できる注文は 10 件だけです。前後のページに移動するには、訪問者はタグによって表示される [次] または [前] のハイパーリンクをクリックします。コードリスト 5-7 では、訪問者に 40 件の注文があり、2 ページ目を表示している場合、タグが [前 | 20-29 | 次] のように表示されます。

コードリスト 5-7 `<eb:smnav>` タグの使用例

```
<%@ taglib uri="eb.tld" prefix="eb" %>

<!-- [ 前 / 10-19 / 次 ] の移動用文字列を表示 -->

<eb:smnav origin="orderhistory.jsp" event="link.viewOrderHistory"
  prevstring="Previous" nextstring="Next"
  pageindex="<%=pageIndexString%" />
```

第6章 イベントおよび行動追跡用 JSP タグ

このタグライブラリには、BEA WebLogic Portal で使われるさまざまな拡張タグが定義されています。それらのタグは、特に Events and Behavior Tracking (イベントおよび行動追跡) サービスで使用します。

イベント タグは、販売促進やキャンペーンのシナリオ アクションに利用できます。行動追跡をアクティブにすると、ユーザのサイト内での行動パターンを記録することができます。行動追跡によってリレーショナル データベースに記録されたイベントは、サードパーティのツールを使って分析できます。

イベント タグは、コンテンツ トラッキングおよび商品トラッキングの2つの領域に大別されます。コンテンツおよび商品のトラッキング用タグは、任意の Web またはポータル アプリケーションで利用できます。

このトピックは、以下の節で構成されています。

- [コンテンツ タグ](#)
- [プロダクト タグ](#)

コンテンツ タグ

注意： 以降の表の「必須」列では、属性が必須と省略可能のどちらであるかを「はい」または「いいえ」で示します。「R/C」列では、「C」は属性がコンパイル時表現であることを、「R」は属性が要求時表現とコンパイル時表現のいずれかであることを示します。

<tr:clickContentEvent>

| | |
|----------|--|
| タグライブラリ | tracking.tld |
| インポート文 | <%@ taglib uri="tracking.tld" prefix="tr" %> |
| 実装されるクラス | ClickContentTag ClickContentExtraInfo |

<tr:clickContentEvent> タグは、ユーザが広告インプレッションをクリックしたときの動作イベントを生成するのに使います。イベントパラメータが入った URL クエリ文字列が返されます。そのため、ハイパーリンクでコンテンツと結ばれる完結した URL を形成するのに使われます。

表 6-1 は、<tr:clickContentEvent> タグの属性についての説明です。

表 6-1 <tr:clickContentEvent> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------------|-----|--------|---|-----|
| documentId | いいえ | String | 表示される項目の ID (画像の URL またはバナー広告の ID、取得可能な場合)。 | R |
| documentType | いいえ | String | 表示される項目のタイプまたはカテゴリ (取得可能な場合)。 | R |
| id | いいえ | String | このタグの出力を保持するページ変数。 | C |
| userId | いいえ | String | コンテンツが検索される対象のユーザの名前。この値は省略可能で、未指定の場合は <code>request.getRemoteUser()</code> の値に設定される。 | R |

例

以下に、Webflow サブレットに移動するクリックスルーの例を示します。このリンクはクリックスルー コンテンツ イベントを生成し、指示されたコンテンツを表示します。ユーザが商品の解説リンクをクリックした後、クリック コンテンツ イベントが生成される様子わかります。アプリケーションの `web.xml`

ファイルで指定された、デフォルト Webflow サブレットの `<filter>` タグで、`ClickThroughEventFilter.doFilter()` メソッドの呼び出しが生成されます。このメソッドでは、`HttpRequest` を調べ、`ClickThroughConstants.EVENT_TYPE` があればクリック イベントを生成します。

`ClickThroughConstants.EVENT_TYPE` は、[コード リスト 6-1](#) に示すように、JSP に `<tr:clickContentEvent>` タグを追加すると生成されます。

コード リスト 6-1 `<tr:clickContentEvent>` タグの使用例

```
<tr:clickContentEvent documentId="<%= documentId %>"
  documentType="<%= documentType %>"
  userId="<%= userId %>"
  id="outputFromTag"
/>
```

以下の例では、目的のコンテンツと、上記のタグからの出力を参照するリンクとを関連付けています。

コード リスト 6-2 コンテンツと [コード リスト 6-1](#) からの出力を参照するリンクとの関連付け

```
<A HREF="<webflow:createWebflowURL event="link.clickContent"
  namespace="trackingWebApp_main" extraParams="<%= outputFromTag
  %>" />" />">ここをクリックすると clickContentEvent を生成 </A>
```

注意： ユーザを別のサイトにリダイレクトするには、`createWebflowURL` タグで `redirect="true"` に設定します。

<tr:displayContentEvent>

<tr:displayContentEvent> タグでは、ユーザが広告を受信した（ユーザに広告が表示された）ときに動作イベントを生成します。「広告」は、画像、テキスト、PDF ファイル、Web 対応マルチメディア コンテンツなど、任意の HTML コンテンツにすることができます。

| | |
|-----------------|--|
| タグライブラリ | tracking.tld |
| インポート文 | <%@ taglib uri="tracking.tld" prefix="tr" %> |
| 実装されるクラス | DisplayContentEventTag |

表 6-2 は、<tr:displayContentEvent> タグの属性についての説明です。

表 6-2 <tr:displayContentEvent> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|--------------|-----|--------|---|-----|
| documentId | いいえ | String | 表示される項目の ID（画像の URL またはパ ナー広告の ID、取得可能な場合） | R |
| documentType | いいえ | String | 表示される項目のタイプまたはカテゴリ（取 得可能な場合） | R |

例

ここでは、<cm:select> の呼び出しに続く処理を行うコードの例を示します。<tr:displayContentEvent> タグは、ドキュメントが返されるたびにイベントを生成して、ドキュメントの ID およびタイプを渡します。この例ではドキュメントの表示は行いません。

コード リスト 6-3 <tr:displayContentEvent> タグの使用例

```
<%@ taglib uri="tracking.tld" prefix="tr" %>
.
.
.
<es:forEachInArray id="nextRow" array="<%=headlines%>"
```

```
type="com.bea.pl3n.content.Content">
<es:NotNull item="<%=nextRow%>">
  <tr:displayContentEvent
    documentId="<%=nextRow.getIdentifier()%>"
    documentType="<%=headingProp%>" />
</es:NotNull>
</es:forEachInArray>
```

プロダクト タグ

注意： 以降の表の「必須」列では、属性が必須と省略可能のどちらであるかを「はい」または「いいえ」で示します。「R/C」列では、「C」は属性がコンパイル時表現であることを、「R」は属性が要求時表現とコンパイル時表現のいずれかであることを示します。

<tr:clickProductEvent>

| | |
|----------|--|
| タグライブラリ | productTracking.tld |
| インポート文 | <%@ taglib uri="productTracking.tld" prefix="trp" %> |
| 実装されるクラス | ClickProductEventTag |

<tr:clickProductEvent> タグは、ユーザがプロダクト インプレッションをクリック（クリックスルー）したときの動作イベントを生成するのに使います。イベント パラメータが入った URL クエリ文字列が返されます。そのため、ハイパーリンクでコンテンツと結ばれる完結した URL を形成するのに使われます。

sku、categoryId、または documentId のうち少なくとも 1 つの属性を指定します。

表 6-3 は、<tr:clickProductEvent> タグの属性についての説明です。

表 6-3 <trp:clickProductEvent> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-----------------|---------------|-------------------------------|---|-----|
| applicationName | いいえ | String | webApp またはアプリケーションの名前（取得可能な場合）。同じサーバ上に展開されている（または、同じデータベースにデータが永続化される）複数の店舗があるときに、データを区別するために使用できる。 | R |
| categoryId | いいえ | String、または Category オブジェクト | 表示されるコンテンツと関連付けられた商品のカテゴリ（取得可能な場合）。 | R |
| documentId | はい | String | 表示される項目の名前（画像の URL または バナー広告の ID、取得可能な場合）。 | R |
| documentType | いいえ | String | 表示される項目のタイプまたはカテゴリ（取得可能な場合）。 | R |
| sku | いいえ （解説参照） | String、または ProductItem オブジェクト | 表示されるコンテンツ項目と関連付けられた商品の ID（取得可能な場合）。通常、sku は、少なくとも categoryId と documentId のいずれかが指定されていなければならない。 | R |
| userId | いいえ | String | コンテンツが検索される対象のユーザの名前。この値は省略可能で、未指定の場合は <code>request.getRemoteUser()</code> の値に設定される。 | R |

例

以下に、Webflow サブレットに移動するクリックスルーの例を示します。このリンクはクリックスルー コンテンツ イベントを生成し、指示されたコンテンツを表示します。この例では、商品名 (`productItem.getName()`) および商品 ID の SKU を使用して、ドキュメント ID を持つ `ClickProductEvent` を生成する方法を示します。

コード リスト 6-4 <trp:clickProductEvent> タグの使用例

```
<%@ taglib uri="productTracking.tld" prefix="trp" %>
.
.
.
<%
detailsUrl = WebflowJSPHelper.createWebflowURL(pageContext,
"itemssummary.jsp", "link(" + detailsLink + ")",
"&" + HttpRequestConstants.CATALOG_ITEM_SKU + "=" +
productItem.getKey().getIdentifier() + "&" +
HttpRequestConstants.CATALOG_CATEGORY_ID + "=" +
category.getKey().getIdentifier() + "&" +
HttpRequestConstants.DOCUMENT_TYPE + "=" + detailsLink, true);
%>

<trp:clickProductEvent
  id="url"
  documentId="<%= productItem.getName() %>"
  sku="<%= productItem.getKey().getIdentifier() %>" />

<%
detailsUrl = detailsUrl + "&" + url;
%>

<a href="<%= detailsUrl %>">
```

<trp:displayProductEvent>

| | |
|-----------------|--|
| タグ ライブラリ | productTracking.tld |
| インポート文 | <%@ taglib uri="productTracking.tld" prefix="trp" %> |
| 実装されるクラス | DisplayProductEventTag |

<trp:displayProductEvent> タグは、ユーザに向けて商品（通常は gif 画像）が表示されたときに動作イベントを生成するために使われます。

sku、categoryId、または documentId のうち少なくとも 1 つの属性を指定します。

表 6-4 は、<trp:displayProductEvent> タグの属性についての説明です。

表 6-4 <trp:displayProductEvent> タグの属性

| タグ属性 | 必須 | データ型 | 解説 | R/C |
|-----------------|---------------|-------------------------------|--|-----|
| applicationName | いいえ | String | webApp またはアプリケーションの名前（取得可能な場合）、同じサーバ上に展開されている（または、同じデータベースにデータが永続化される）複数の店舗があるときに、データを区別するために使用できる。 | R |
| categoryId | いいえ | String、または Category オブジェクト | 表示されるコンテンツと関連付けられた商品のカテゴリ（取得可能な場合）。 | R |
| documentId | いいえ | String | 表示される項目の名前（画像の URL または バナー広告の ID、取得可能な場合）。 | R |
| documentType | いいえ | String | 表示される項目のタイプまたはカテゴリ（取得可能な場合）。 例： DisplayProductEvent.CATEGORY_BROWSE DisplayProductEvent.ITEM_BROWSE DisplayProductEvent.CATEGORY_VIEW DisplayProductEvent.BANNER_AD_PROMOTION | R |
| sku | いいえ （解説参照） | String、または ProductItem オブジェクト | 表示されるコンテンツ項目と関連付けられた商品の ID（取得可能な場合）。通常、sku は、少なくとも categoryId と documentId のいずれかが指定されていれば必要ない。 | R |

例

コードリスト 6-5 に、カタログ項目の取得に続く処理を行うコードの例を示します。<tr:displayProductEvent> タグではイベントを生成して、ドキュメントの ID と、商品アイテムの種類および SKU 番号を渡します。

コードリスト 6-5 <tr:displayProductEvent> タグの使用例

```
<%@ taglib uri="productTracking.tld" prefix="trp" %>
...
<trp:displayProductEvent
    documentId="<%= item.getName() %>"
    documentType="<%= DisplayProductEvent.ITEM_BROWSE %>"
    sku="<%= item.getKey().getIdentifier() %>" />
```

付録 A JSP テンプレート

WebLogic Portal には、アプリケーションで利用できる定義済みの JSP テンプレート セットが用意されています。これらのテンプレートは、BEA が提供している形でそのまま使うことも、独自のスキンやレイアウトを適用するなど個別のニーズに合わせてカスタマイズすることもできます。この節には、利用できるテンプレートの一覧があります。

この節では、以下の内容について説明します。

- テンプレートの場所
- Commerce テンプレート
- Campaign テンプレート
- Catalog テンプレート
- Catalog および Campaign の include テンプレート
- Order テンプレート
- Registration テンプレート
- User テンプレート

テンプレートの場所

テンプレートはすべて、次のディレクトリに置かれています。

```
<BEA_HOME>\weblogic700\portal\samples\portal\wlcsDomain\  
beaApps\wlcsApp\wlcs\commerce
```

Commerce テンプレート

表 A-1 は、Commerce サービス開発に使用するテンプレートの一覧です。

表 A-1 /commerce テンプレート

| | |
|------------------------|---|
| main.jsp | 匿名ユーザ向けのメイン ホーム（開始）ページ。 |
| login.jsp | アカウント、および新規ユーザ作成用のリンクにアクセスするためのユーザ名とパスワードをフォームベースで発行する。 |
| badlogin.jsp | login.jsp と同様で、login.jsp に加えてエラー メッセージを表示する。 |
| configurationerror.jsp | Webflow のコンフィグレーションまたは Pipeline でエラーが発生したときに使われる。 |
| newuserforward.jsp | user/newusercreation.jsp にリダイレクトする。 |
| servererror.jsp | 403、404、405、500 番のサーバエラーに使われる汎用ページ。 |
| sessiontimeout.jsp | セッション タイムアウト ページ。 |
| shoppingcart.jsp | 注文品が提示される。ユーザは、数量の変更や商品の削除を行ったり、ショッピングを続けたりすることができる。 |

Campaign テンプレート

表 A-2 は、Campaign サービスの設定に利用できる JSP テンプレートの一覧です。

表 A-2 /commerce/campaign テンプレート

| | |
|------------------------|-----------------------------|
| contractor_journal.jsp | キャンペーンの例に関係するサンプルのオフサイトページ。 |
|------------------------|-----------------------------|

表 A-2 /commerce/campaign テンプレート (続き)

| | |
|-------------------|----------------------------|
| fixup_channel.jsp | キャンペーンの例に関するサンプルのオフサイトページ。 |
|-------------------|----------------------------|

Catalog テンプレート

表 A-3 は、Catalog サービスの設定に利用できる JSP テンプレートの一覧です。

表 A-3 /Catalog テンプレート

| | |
|-------------------|--|
| browse.jsp | カタログのブラウズによって、連続的にドリルダウンする。サブカテゴリとサブサブカテゴリにも同じテンプレートが使われる。 |
| details.jsp | 単一商品のメタデータのうち選択されたものをすべて表示する。親カテゴリへのバックトラック ナビゲーションも可能。 |
| search.jsp | 詳細検索 (パワーサーチ) ページ。詳細検索結果も表示する。 |
| searchresults.jsp | サイト内のどこでも行われる簡易キーワード検索の結果。 |

Catalog および Campaign の include テンプレート

表 A-4 は、Catalog または Campaign サービスの設定に利用できる JSP include テンプレートの一覧です。

表 A-4 Catalog および Campaign include テンプレート

| | |
|-----------------|---|
| category.jsp | サブカテゴリの一覧を作成し、親（現在の）カテゴリレベルにあるあらゆる商品の商品要約を提示する。商品数が所定の数を越える場合には、一覧の続きは、[次] ページおよび [前] ページで参照できる。 |
| itemdetails.jsp | 商品の詳細情報を表示するインクルード モジュール。 |
| itemsummary.jsp | 検索ページやブラウズ ページで複数の商品をまとめて表示できるインクルード モジュール。 |
| navigation.jsp | ブラウズ インタフェースにおける「バックトラック」ナビゲーション機能を提供するインクルード モジュール。 |
| navigation2.jsp | ブラウズ インタフェースにおける「バックトラック」ナビゲーション機能を提供するインクルード モジュール。 |
| admin.inc | すべてのページで使用されて、上部の赤と黒のパナーを表示する。メインの管理画面、このテンプレートのインデックス、および現在のテンプレートの *.jsp.html ファイルに対するリンクが表示される。 |
| header.inc | すべてのページで使用されて、[ホーム] へのリンクを表示する。header.inc をインクルードするページによっては、[サーチ]、[カートの中を見る]、[ログイン] (ユーザーがまだログインしていない場合)、[ログアウト] (認証済みユーザーに対して) などのリンクも表示される。 |
| footer.inc | すべてのページで使用される。 |
| leftside.inc | 簡易検索機能とプロモーション用広告が表示される。認証済みユーザーの場合は、ユーザーに対するパーソナライズされたメッセージ、顧客プロファイルへのリンク、注文履歴へのリンク、および支払い履歴へのリンクも表示される。 |
| stylesheet.inc | すべてのページで使用される。サイト全体に適用されるパラグラフやテキストのスタイルを定義するカスケーディング スタイルシート。 |

表 A-4 Catalog および Campaign include テンプレート (続き)

| | |
|-----------------------------|---|
| states.inc | 入力ポップアップ ウィンドウに選択肢として米国の州名が列挙されるあらゆるフォームで使われる。 |
| countries.inc | 入力ポップアップ ウィンドウに選択肢として国名が列挙されるあらゆるフォームで使われる。 |
| editaddressstemplate.inc | ユーザが登録済みの届け先住所を編集できるプロフィールおよび注文用フォームで使われる。 |
| editccstemplate.inc | ユーザが登録済みのクレジットカード情報を編集できるプロフィールおよび注文用フォームで使われる。 |
| newaddressstemplate.inc | ユーザが新しい届け先住所を指定できるプロフィールおよび注文用フォームで使われる。 |
| newccstemplate.inc | ユーザが支払用に新しいクレジットカードを指定できるプロフィールおよび注文用フォームで使われる。 |
| newdemographicstemplate.inc | register/newuser.jsp および user/editdemographics.jsp にインクルードされる。ユーザがデモグラフィック情報を入力または変更できるフォームを提示する。提供される詳細なデモグラフィック情報は、キャンペーン関連の機能で使用できる。 |

Order テンプレート

表 A-5 は、Order サービスの設定に利用できるテンプレートの一覧です。

表 A-5 /Order テンプレート

| | |
|-------------------|---|
| selectaddress.jsp | 登録済みの届け先住所（複数の場合あり）を表示する。ユーザは、届け先住所の選択、編集、および新規住所の追加を行える。 |
| addaddress.jsp | selectaddress.jsp で、ユーザが希望する届け先住所が見つからない場合に表示される。newaddressstemplate.jsp をインクルード。 |

表 A-5 /Order テンプレート (続き)

| | |
|----------------------|---|
| shipping.jsp | 出荷方法の選択肢として、速達便、航空便、UPS などが提示される。ユーザは出荷に関して特別な指示をすることもできる。 |
| selecttaxaddress.jsp | 特定の税務管轄区域に依存する。ユーザは郡などを指定し、チェックアウトを続行できる。 |
| payment.jsp | 現在の取引に使用する (クレジットカード) アカウントの選択。ユーザは既存アカウントの選択、編集、新規アカウントの追加を行うことができる。 |
| paymentnewcc.jsp | ユーザは、新しいクレジットカード情報を追加できる。newcctemplate.inc をインクルード。 |
| paymenteditcc.jsp | ユーザは、クレジットカード情報を payment.jsp での選択に合わせて変更できる。editcctemplate.inc をインクルード。 |
| checkout.jsp | 請求の最終的承認を得るために、商品、届け先住所、出荷方法の選択、クレジットカード、税金、および総額を明示して取引内容全体が表示される。 |
| confirmorder.jsp | チェックアウトが正常に実行されたあと、注文が受理されたことを確認する要約が表示される。 |
| orderhistory.jsp | ユーザは、過去または進行中の注文の一覧を表示し、そのうちの 1 つを選んで、内容、出荷日、その他の詳細を表示できる。 |
| paymenthistory.jsp | ユーザは、過去の注文の支払い情報の一覧を表示できる。 |
| orderstatus.jsp | 個々の注文が、そのアイテム、費用、状況、届け先住所、その他の詳細と共に表示される。 |
| systemerror.jsp | トランザクション コンポーネントでのサービス システム障害のために発生したエラーを表示する。 |

Registration テンプレート

表 A-6 は、Registration サービスの設定に利用できる JSP テンプレートの一覧です。

表 A-6 Register テンプレート

| | |
|-------------|--|
| newuser.jsp | プロファイルの個人情報フィールドをすべて空白にする。ユーザ プロファイル情報を記録するためのデモグラフィック調査も含まれる。WebLogic Portal の完全ライセンス（Campaign サービスが含まれる）がある場合は、このデータをキャンペーンに利用できる。 |
|-------------|--|

User テンプレート

表 A-7 は、ユーザの設定に利用できる JSP テンプレートの一覧です。

表 A-7 User テンプレート

| | |
|----------------------|--|
| viewprofile.jsp | ログイン ユーザの既存の個人情報を、情報の変更および追加用のボタンと共に表示する。 |
| editdemographics.jsp | 登録済みユーザは、以前に入力したデモグラフィック情報を変更できる。 |
| editprofile.jsp | 個人情報（名前、連絡先、電話番号、電子メール）を変更する。 |
| changepassword.jsp | ユーザは、パスワードを変更できる。 |
| profilenewcc.jsp | ユーザは、新しいクレジット アカウント情報を追加できる。newcctemplate.inc をインクルード。 |
| profileeditcc.jsp | ユーザは、クレジット アカウント情報を payment.jsp での選択に合わせて変更できる。editcctemplate.inc をインクルード。 |

表 A-7 User テンプレート (続き)

| | |
|-------------------------|--|
| profilenewaddress.jsp | ユーザは、新しい届け先住所を追加できる。 newaddresstemplate.inc をインクルード。 |
| profileeditaddress.jsp | ユーザはプロフィールに登録済みの届け先住所を変更できる。 editaddresstemplate.inc をインクルード。 |
| newusercreation.jsp | ユーザ登録の成功後に表示され、ショッピング カートを表示したり、チェックアウトに進んだり、ホームに戻るためのリンクを表示する。 |
| secureMain.jsp | ログインに成功したユーザを main.jsp に戻すだけの安全なページ。 |
| usercreationforward.jsp | ユーザの作成後、newusercreation.jsp ページをリダイレクトするのに使用する。これは、応答オブジェクト経由で適切な URL にリダイレクトしない限り、要求にユーザ情報が含まれないため。この usercreationforward.jsp により、認証済みユーザの情報を持つ新規要求が作成され、キャンペーンの開始が可能になる。 |

付録 B JSP パフォーマンスのチューニング

顧客が e ビジネスの Web サイトのページを要求すると、WebLogic Portal は対応する JSP をコンパイルしてサーブレットを生成します。コンパイル済みの JSP から生成されるサーブレットだけでなく、WebLogic Portal には、システムのさまざまなコンポーネント間で情報を交換するためのサーブレットが用意されています。

この節では、JSP のチューニング、コンパイル、および更新についての情報を示します。説明する内容は以下のとおりです。

- ステップ 1: JSP のプリコンパイル
- ステップ 2: Web アプリケーションに対する Java コンパイラの指定
- ステップ 3: JSP およびサーブレットの変更チェック間隔の調整

ステップ 1: JSP のプリコンパイル

デプロイする Web アプリケーションごとに、WebLogic Portal が JSP をコンパイルするタイミングを指定できます。

- `weblogic.jspc` を使用して JSP を手動でプリコンパイルします。
- サーバの起動時にアプリケーション サーバがすべての JSP をプリコンパイルするように指定できます。プリコンパイル オプションを有効にすると、サーバの起動プロセスは Web アプリケーションに新しい JSP または変更された JSP があるかどうかを検査し、あればそれをコンパイルします。
- 要求されたときにだけアプリケーション サーバが JSP をコンパイルするように指定できます。このオプションが無効になっていると、サーバは短時間で起動しますが、新しい JSP または変更された JSP にユーザがはじめてアクセスした時点で JSP をコンパイルしなければならず、長い遅延が発生します。

WebLogic Portal のサンプル アプリケーションは、インストール時にプリコンパイルされます。

以下の手順に従います。詳細については、WebLogic Server の XML 要素リファレンス (http://e-docs.beasys.co.jp/e-docs/wls/docs70/webapp/weblogic_xml.html) および『WebLogic Server 管理者ガイド』 (<http://e-docs.beasys.co.jp/e-docs/wls/docs70/adminguide/overview.html#1036999>) を参照してください。

1. Web アプリケーションが WAR ファイルとしてデプロイされる場合、ファイルを展開 (unjar) します。
2. WebLogic Server コンソールで、対象の Web アプリケーションを選択し、[コンフィグレーション] タブをクリックし、[precompile] チェック ボックスをオンにします。 (weblogic.xml ファイルにおけるパラメータ名は `weblogic.jsp.precompile` です)
3. ファイルを保存します。
4. 必要であれば、WAR ファイルを再度アーカイブします。
5. サーバを再起動します。

ステップ 2: Web アプリケーションに対する Java コンパイラの指定

WebLogic Server Administration Console では、各サーバ コンフィグレーションに対する Java コンパイラが示されます。サーバにデプロイされるすべてのアプリケーションは、Web アプリケーションの `weblogic.xml` ファイルで別のコンパイラが指定されていない限り、このコンパイラを使用します。

サーバに対する現在の Java コンパイラを確認するには、WebLogic Server Administration Console の左側のペインで対象のサーバをクリックします。右ペインで [コンフィグレーション] タブをクリックし、[コンパイラ] サブタブをクリックします。Web アプリケーションで使用する Java コンパイラのパス名を入力して、新しい名前を入力します。変更内容を有効にするには、サーバを再起動します。

詳細については、WebLogic Server の XML 要素リファレンス (http://e-docs.beasys.co.jp/e-docs/wls/docs70/webapp/weblogic_xml.html)、および『WebLogic Server 管理者ガイド』 (<http://e-docs.beasys.co.jp/e-docs/wls/docs70/adminguide/overview.html#1036999>) を参照してください。

ステップ 3: JSP およびサーブレットの変更 チェック間隔の調整

Web アプリケーションの JSP および他のサーブレットのソース ファイルに対する変更をサーバが検査する頻度を指定できます。

サンプル Web アプリケーションでは、Web ブラウザが JSP を要求するたびに JSP に対する変更をチェックします。同様に、サーバは、サンプル Web アプリケーションのサーブレットにリクエストを送信するたびに、サーブレットのクラス ファイルに対する変更をチェックします。

プロダクション環境の Web サイトでは、サーバが変更のチェックを行う間隔を長くすることで、WebLogic Portal が JSP を提供したりサーブレットに対するリクエストを処理したりする時間を短縮できます。

変更チェック間隔の値を大きくするほどサーバの動作は速くなりますが、ソース ファイルの変更に対する感度は低くなります。たとえば、JSP の変更をサーバがチェックする間隔を 10 分に設定できます。JSP を変更した後で、変更がサーバに認識されるまでに 10 分間かかります。

この節では、以下の内容について説明します。

- ページチェック間隔のプロパティ
- 間隔の調節

ページ チェック間隔のプロパティ

`pageCheckSeconds` 属性では、Web アプリケーションの JSP ファイルが変更されていて再コンパイルが必要かどうかをサーバがチェックする間隔を設定します。このプロパティは、Web アプリケーションごとに `WEB-INF\weblogic.xml` ファイルの中で個別に定義されます。

```
<jsp-param>
  <param-name>pageCheckSeconds</param-name>
  <param-value>1</param-value>
</jsp-param>
```

データベースやコンテンツ管理システムに格納されているコンテンツの更新をサーバがチェックする頻度は、ページ チェック間隔では指定されません。コンテンツの更新周期は、さまざまなキャッシュに対する TTL（生存時間）の設定によって決まります。たとえば、ページ チェック間隔を毎秒 1 回に設定し、コンテンツ キャッシュに対する TTL を 10 分に設定すると、サーバによる新しい JSP ソース コードのチェックは 1 秒ごとに行われますが、サーバが新しいコンテンツを認識するには最大で 10 分かかかる可能性があります。TTL プロパティの設定について、詳しくは『開発者ガイド』の「パフォーマンス チューニング」（<http://edocs.beasys.co.jp/e-docs/wlp/docs70/dev/x10ding.htm#1040480>）を参照してください。

間隔の調節

プロダクション環境の Web サイトに最適なページ チェック間隔およびサーブレット再ロード間隔を設定するには、次の手順に従います。

1. 間隔を `-1`（サーバが変更チェックを行わないようにする設定値）に設定して WebLogic Portal のパフォーマンスをテストし、パフォーマンスの基準を設定します。
2. 間隔にさまざまな秒数を設定し、パフォーマンスをテストします。たとえば、間隔を `600` 秒（10 分）に設定して、パフォーマンスをテストします。次に `900` 秒に設定してテストします。
3. JSP ファイルとサーブレット クラスの変更を十分な頻度でチェックしながら最善のパフォーマンスが得られる間隔を選択します。

付録 C JSP タグ リファレンス（名前順）

この章では、ポータル開発に利用できるすべての JSP タグをタグ名順に示します。タグ属性の説明や、タグの使用方法を示すサンプルコードなど、タグについてより詳しい情報を確認するには、表 C-1 中のタグ名をクリックしてください。

表 C-1 ポータル開発 JSP タグ（名前順）

| タグ名 | 解説 | タグライブラリ |
|--|---|---------|
| <ad:adTarget> | コンテンツ管理システムに広告クエリを送る。<ph:placeholder> タグとは異なり、<ad:adTarget> タグは広告ブレースホルダ内の他のクエリと競合しない。 | ad.tld |
| <catalog:getProperty> | 表示用のプロパティを ProductItem または Category のどちらかから読み込む。プロパティは明示のプロパティ（Catalog 項目の get メソッドを使用して取得できるプロパティ）または暗黙のプロパティ（Catalog 項目の ConfigurableEntity getProperty メソッドを通じて参照できるプロパティ）のどちらでもかまわない。 | cat.tld |
| <catalog:iterateThroughView> | 指定された ViewIterator の View に対して反復処理を行う。このタグは、View の最後に達するまで、一度に 1 つの Catalog 項目に対して View を反復する。 | cat.tld |

表 C-1 ポータル開発 JSP タグ (名前順) (続き)

| タグ名 | 解説 | タグライブラリ |
|---|--|---------|
| <catalog:iterateViewIterator> | ViewIterator に対して反復処理を行う。ViewIterator は、一連の固定サイズの View に分割された、リモート データの大規模な集合に対するイテレータである。 | cat.tld |
| <cm:getProperty> | 指定されたコンテンツ メタデータ プロパティの値を取り出して、resultId で指定された変数に入れる。このタグには本体はない。resultId が指定されない場合、<cm:printProperty> タグと同様に、値はページにインラインで埋め込まれる。 | cm.tld |
| <cm:printDoc> | Document オブジェクトの未処理バイト データを JSP 出力ストリームにインラインで埋め込む。このタグは本体をサポートしておらず、また Document オブジェクトしかサポートしない。このタグでは、テキストとバイナリ データは区別されない。 | cm.tld |
| <cm:printProperty> | 指定されたコンテンツ メタデータ プロパティの値を文字列としてインラインに埋め込む。このタグには本体はない。このタグは、Content オブジェクトだけでなく、任意の ConfigurableEntity に作用する。ただし、ConfigurableEntity サクセサはサポートされない。 | cm.tld |

表 C-1 ポータル開発 JSP タグ (名前順) (続き)

| タグ名 | 解説 | タグ ライブラリ |
|---|---|----------|
| <code><cm:select></code> | 検索式のクエリ構文だけを使用してコンテンツを選択する。このタグは本体をサポートせず、また使用することもない。このタグが <code><es:forEachInArray></code> タグを返した後で、Content オブジェクトの配列の反復処理にゼロを使用できる。このタグは、ContentManager インタフェースを通じて、汎用の Content をサポートする。 | cm.tld |
| <code><cm:selectById></code> | Content の一意な識別子を使ってコンテンツを取得する。このタグには本体はない。このタグは基本的には select タグのラッパーである。文字列で指定可能な主キーを持つ、すべての Content オブジェクトに対して機能する。 | cm.tld |
| <code><eb:smnav></code> | 参照される対象だけが取り出されるように、値オブジェクトを取得する。このタグを使用すると、参照の対象となっている値オブジェクトのリストに含まれる要素の表示形式を制御でき、その前後のページへのリンクが提供される。 | eb.tld |
| <code><es:convertSpecialChars></code> | 通常は HTML ブラウザにとって特別な意味を持つ文字を、作者の意図どおりに表示できるように変換する。 | es.tld |
| <code><es:counter></code> | for ループ構文を作成する。 | es.tld |
| <code><es:date></code> | ユーザの時間帯設定に基づいて書式化された日時文字列を取得する。 | es.tld |

表 C-1 ポータル開発 JSP タグ (名前順) (続き)

| タグ名 | 解説 | タグライブラリ |
|--|---|----------|
| <code><es:forEachInArray></code> | 配列に対して反復処理を行う。 | es.tld |
| <code><es:isNull></code> | 値がヌル (null) かどうか調べる。対象の値が String 型の場合、 <code><es:isNull></code> タグは、その String が null かそれとも値を持っているかを調べるために使われる。空文字列の場合には、 <code>isNull</code> の結果は <code>false</code> になる (空文字列は null ではない)。 | es.tld |
| <code><es:notNull></code> | <code><es:notNull></code> タグは、値が null でないかどうかを調べるために使われる。対象の値が String 型の場合、 <code><es:notNull></code> タグは、その String が null でないか、それとも値を持っているかを調べるために使われる。空文字列の場合、 <code>notNull</code> の結果は <code>true</code> になる (空文字列は値として扱われる)。 | es.tld |
| <code><es:transposeArray></code> | <code><es:transposeArray></code> タグは、標準の [行] [列] 配列を [列] [行] 配列に転置するために使われる。 | es.tld |
| <code><es:uriContent></code> | URL からコンテンツを取り出す。このタグは、大量のテキストが含まれるページを取得する用途に最も適している。 | es.tld |
| <code><i18n:getMessage></code> | <code><i18n:localize></code> タグと組み合わせ、ローカライズされた静的テキストまたはメッセージを <code>JspMessageBundle</code> から取り出すために使われる。 | i18n.tld |

表 C-1 ポータル開発 JSP タグ (名前順) (続き)

| タグ名 | 解説 | タグライブラリ |
|--|--|-------------------------|
| <i18n:localize> | <code><i18n:getmessage></code> タグを通じてリソースバンドルにアクセスするときにページ内で使われる言語コード、国別コード、ロケール変化形、および基本バンドル名を定義する。 | <code>i18n.tld</code> |
| <ph:placeholder> | JSP ページ上の特定の位置の動作を定義するプレースホルダを実装する。プレースホルダは E-Business Control Center を使って定義する。 「 <code><ad:adTarget></code> 」も参照。 | <code>ph.tld</code> |
| <portal:createPortalPageChangeURL> | ページ変更イベントの Webflow URL を生成する。 | <code>portal.tld</code> |
| <portal:createWebflowURL> | JSP 内で Webflow URL を動的に作成する。Webflow URL には、プロトコル、ドメイン名、ポート、Web アプリケーション URI、WebflowServlet URI、およびクエリ文字列を入れることができる。 | <code>portal.tld</code> |
| <portal:form> | HTML フォーム タグを生成する。このタグは <code><portal:validatedForm></code> タグよりも機能が制限された単純なものである。 <code><portal:validatedForm></code> タグの詳細については、次の項目を参照。 | <code>portal.tld</code> |
| <portal:validatedForm> | 検証可能な HTML フォームを動的に生成する。Web サイトへの訪問者が無効な情報を入力すると、訪問者の入力はいったん保存され、適切なエラーメッセージを付けて再表示される。 | <code>portal.tld</code> |

表 C-1 ポータル開発 JSP タグ (名前順) (続き)

| タグ名 | 解説 | タグライブラリ |
|---|---|-------------|
| <code><portlet:createPortletEditURL></code> | ポートレットの編集を表す Webflow URL を生成する。 | portlet.tld |
| <code><portlet:createPortletFloatURL></code> | 「移動可能な」ポートレット (独立したウィンドウに表示されるポートレット) の作成を表す Webflow URL を生成する。 | portlet.tld |
| <code><portlet:createPortletMaximizeURL></code> | ポートレットの最大化を表す Webflow URL を生成する。 | portlet.tld |
| <code><portlet:createPortletMinimizeURL></code> | ポートレットの最小化を表す Webflow URL を生成する。 | portlet.tld |
| <code><portlet:createPortletUneditURL></code> | 編集 URL から別のページへの移動を表す Webflow URL を生成する。 | portlet.tld |
| <code><portlet:createPortletUnmaximizeURL></code> | ポートレットの最大化解除を表す Webflow URL を生成する。 | portlet.tld |
| <code><portlet:createPortletUnminimizeURL></code> | ポートレットの最小化解除を表す Webflow URL を生成する。 | portlet.tld |
| <code><portlet:createWebflowURL></code> | JSP 内で Webflow URL を動的に作成する。Webflow URL には、プロトコル、ドメイン名、ポート、Web アプリケーション URI、WebflowServlet URI、およびクエリ文字列を入れることができる。 | portlet.tld |
| <code><portlet:form></code> | HTML フォーム タグを動的に生成する。このタグは <code><portlet:validatedForm></code> タグよりも機能が限定された単純なものである。 | portlet.tld |

表 C-1 ポータル開発 JSP タグ (名前順) (続き)

| タグ名 | 解説 | タグライブラリ |
|---|---|--------------------------|
| <code><portlet:getException></code> | Webflow プロセッサから送出された例外またはメッセージを取り出す。取り出し可能なのは、 <code>InvalidFormFieldException</code> または <code>ProcessingException</code> と関連付けられたメッセージ。このタグは、例外に対して <code>getMessage()</code> を呼び出す箇所にインラインに埋め込むことも、あるいは例外自体を表すスクリプト変数を返すようにすることもできる。 | <code>portlet.tld</code> |
| <code><portlet:validatedForm></code> | 検証可能な HTML フォームを動的に生成する。Web サイトへの訪問者が無効な情報を入力すると、訪問者の入力はいったん保存され、適切なエラーメッセージを付けて再表示される。 | <code>portlet.tld</code> |
| <code><ps:getPropertyNames></code> | 指定されたプロパティ セットのプロパティ名のリストを取り出す。 | <code>ps.tld</code> |
| <code><ps:getPropertySetNames></code> | 指定されたプロパティ セットタイプのプロパティ セットのリストを取り出す。 | <code>ps.tld</code> |
| <code><ps:getRestrictedPropertyValues></code> | 特定のプロパティ定義について、制限付きの値のリストを <code>String</code> 型に変換して返す。値は文字列 (<code>String</code>) の配列として返される。 | <code>ps.tld</code> |

表 C-1 ポータル開発 JSP タグ (名前順) (続き)

| タグ名 | 解説 | タグライブラリ |
|---|--|--------------|
| <code><pz:contentQuery></code> | コンテンツ マネージャ内のコンテンツを対象にコンテンツ属性検索を実行する。useCache 属性が true に設定されている場合、コンテンツ管理クエリの結果はキャッシュされる。このタグには開始タグしかなく、本体と終了タグはない。このタグは、コンテンツクエリの実行結果としてコンテンツ マネージャから返された Content オブジェクトの配列を返す。 | cm.tld |
| <code><pz:contentSelector></code> | コンテンツ セレクタ ルールに基づいて、パーソナライゼーションされた任意のコンテンツを推奨する。 | cm.tld |
| <code><pz:div></code> | ルール エンジンによる分類子 ルールの実行結果として、条件に従ってコンテンツの断片を取り込む。ユーザのプロファイルが E-Business Control Center で指定された分類に一致する場合、条件に合ったコンテンツが取り込まれる。このタグには開始タグ、本体、終了タグがある。このタグはユーザが属する Classification オブジェクトのリストを返す。 | cm.tld |
| <code><tr:clickContentEvent></code> | ユーザが広告をクリックしたときに動作イベントを生成する。イベント パラメータが入った URL クエリ文字列を返す。この文字列はその後、コンテンツへのハイパーリンクとなる完全な URL を生成するときに使われる。 | tracking.tld |

表 C-1 ポータル開発 JSP タグ (名前順) (続き)

| タグ名 | 解説 | タグライブラリ |
|---|---|---------------------|
| <code><tr:displayContentEvent></code> | ユーザに向けて広告 (通常は gif 画像) が表示されたときに動作イベントを生成する。 | tracking.tld |
| <code><tr:clickProductEvent></code> | ユーザが画面上の商品をクリックしたときに動作イベントを生成する。イベントパラメータが入った URL クエリ文字列を返す。この文字列はその後、コンテンツへのハイパーリンクとなる完全な URL を生成するときに使われる。 | productTracking.tld |
| <code><tr:displayProductEvent></code> | ユーザに向けて商品 (通常は gif 画像) が表示されたときに動作イベントを生成する。sku、categoryId、または documentId のうち少なくとも 1 つの属性を指定する。 | productTracking.tld |
| <code><um:addGroupToGroup></code> | 指定された childGroupName に一致するグループを、指定された groupName に一致するグループに追加する。グループは親を 1 つしか持てないので、そのグループが別の親に属していることを表す以前のデータベースレコードはすべて破棄される。このタグが正常に動作するには、親グループも子グループもあらかじめ存在していなければならない。このタグには本体はない。 | um.tld |
| <code><um:addUserToGroup></code> | 指定された username に一致するユーザを、指定された groupName に一致するグループに追加する。このタグが正常に動作するには、指定されたユーザも指定されたグループもあらかじめ存在していなければならない。このタグには本体はない。 | um.tld |

表 C-1 ポータル開発 JSP タグ (名前順) (続き)

| タグ名 | 解説 | タグライブラリ |
|--|---|---------|
| <code><um:createGroup></code> | 新しいグループをレルムに作成し、対応するグループ プロファイルをパーソナライゼーション データベースに作成する。このタグには本体はない。 | um.tld |
| <code><um:createUser></code> | 新しいユーザ プロファイルを作成する。このタグには本体はない。このタグはグループおよびユーザ管理タグとして分類されているが、指定によっては、現在の匿名ユーザ プロファイルに関連付けられているあらゆるプロパティを永続化するので、実行時処理と組み合わせて使用できる。 | um.tld |
| <code><um:getChildGroupNames></code> | 指定されたグループの子に当たるすべてのグループの名前を返す。 | um.tld |
| <code><um:getGroupNamesForUser></code> | 指定されたユーザが直接所属するグループに一致するグループ名を検索し、結果を String 配列に入れて返す。このタグには本体はない。 | um.tld |
| <code><um:getParentGroupName></code> | 指定された groupName に関連付けられているグループの親の名前を取得する。情報はレルムから取得される。このタグには本体はない。 | um.tld |
| <code><um:getProfile></code> | 指定されたプロファイル キーとプロファイル タイプに一致するプロファイルを取得する。このタグには本体はない。取得されたプロファイルは <code>com.bea.pl3n.usermgmt.profile.ProfileWrapper</code> として扱うことができる。 | um.tld |

表 C-1 ポータル開発 JSP タグ (名前順) (続き)

| タグ名 | 解説 | タグ ライブラリ |
|--|--|----------|
| <code><um:getProperty></code> | プロパティ セットとプロパティ の名前の組を指定してプロパティ 値を取得する。このタグには本体はない。返される値は Object。 | um.tld |
| <code><um:getPropertyAsString></code> | <code><um:getProperty></code> タグと同じ動作をするが、取得されるプロパティ 値は必ず String になる。次の例では、Collection を返し、好きな色を列挙する多値プロパティを示している。 | um.tld |
| <code><um:getTopLevelGroups></code> | 親グループのないグループのグループ名の配列を取得する。情報はレルムから取得される。このタグには本体はない。 | um.tld |
| <code><um:getUsernames></code> | 指定された検索式に一致するユーザ名を格納した String 配列を取得する。検索式では、ワイルドカード文字としてアスタリスク (*) だけを使用でき、大文字と小文字は区別されない。検索式では任意の数のアスタリスクを使用できる。このタグには本体はない。 | um.tld |
| <code><um:getUsernamesForGroup></code> | 指定された検索式に一致するユーザ名のうち、指定されたグループのメンバーに一致するものを格納した String 配列を取得する。検索式では、ワイルドカード文字としてアスタリスク (*) だけを使用でき、大文字と小文字は区別されない。検索式では任意の数のアスタリスクを使用できる。このタグには本体はない。 | um.tld |

表 C-1 ポータル開発 JSP タグ (名前順) (続き)

| タグ名 | 解説 | タグライブラリ |
|--|---|---------|
| <code><um:login></code> | 現在のセキュリティ レルムに対する簡易認証 (ユーザ名、パスワード) を提供し、認証済みユーザを現在の WebLogic ユーザとして設定する。このタグには本体はない。 | um.tld |
| <code><um:logout></code> | 現在のユーザの WebLogic Server セッションを終了する。このタグは <code><um:login></code> タグと組み合わせることで使うことが推奨される。 | um.tld |
| <code><um:removeGroup></code> | <code>groupName</code> で指定されたグループ名に一致するグループを削除する。このタグには本体はない。 | um.tld |
| <code><um:removeGroupFromGroup></code> | 親グループから子グループを 1 つ削除する。 | um.tld |
| <code><um:removeProperty></code> | 現在のセッションのプロファイル、または匿名ユーザ プロファイルから指定されたプロパティを削除する。このタグには本体はない。削除されたプロパティに対して引き続き <code><um:getProperty></code> を呼び出すと、プロパティ セットで規定されているデフォルトのプロパティ値か、またはプロファイルのサクセサから継承されるデフォルト値が得られる。 | um.tld |
| <code><um:removeUser></code> | <code>username</code> で指定されたユーザ名に一致するユーザを削除する。データベース内で <code>profileType</code> が設定されている任意のタイプの拡張ユーザを削除できる。このタグには本体はない。 | um.tld |

表 C-1 ポータル開発 JSP タグ (名前順) (続き)

| タグ名 | 解説 | タグライブラリ |
|--|--|----------|
| <um:removeUserFromGroup> | グループから 1 名のユーザを削除する。 ** このタグは、現在のレルムが <code>weblogic.security.acl.ManageableRealm</code> の実装である場合にのみ呼び出すことが推奨される。このインタフェースはデフォルトの WebLogic Portal レルム (<code>com.bea.pl3n.security.realm.RDBMSRealm</code>) によって実装される。 | um.tld |
| <um:setPassword> | 指定されたユーザ名に一致するユーザのパスワードを更新する。 ** このタグは、現在のレルムが <code>weblogic.security.acl.ManageableRealm</code> の実装である場合にのみ呼び出すことが推奨される。 | um.tld |
| <um:setProperty> | セッションの現在のプロフィールまたは匿名ユーザプロフィールのプロパティ値を更新する。このタグには本体はない。 | um.tld |
| <util:invalidURL> | 指定された URL が null であるか、または 4 文字未満である場合にタグの本体が処理される。 | util.tld |
| <util:validURL> | 指定された URL が null でなく、かつ 4 文字以上である場合にタグの本体が処理される。 | util.tld |

表 C-1 ポータル開発 JSP タグ (名前順) (続き)

| タグ名 | 解説 | タグライブラリ |
|---|--|-------------|
| <webflow:checkbox> | チェックボックスに必要な HTML を生成する。このタグは、InputProcessor が InvalidFormFieldException を送出したときに入力内容を保存する。このタグは <webflow:validatedForm> タグ内にネストする必要がある。 | webflow.tld |
| <webflow:createResourceURL> | アプリケーションの WEB-INF/web.xml ファイル内のコンテキストパラメータ P13N_STATIC_ROOT の値を使用して、リソースの静的 URL を作成する。このタグは、GIF 画像を別のサーバからロードするために使われる場合がある。 | webflow.tld |
| <webflow:createWebflowURL> | JSP 内で Webflow URL を動的に作成する。Webflow URL には、プロトコル、ドメイン名、ポート、Web アプリケーション URI、WebflowServlet URI、およびクエリ文字列を入れることができる。 | webflow.tld |
| <webflow:form> | HTML フォーム タグを動的に生成する。このタグは <webflow:validatedForm> タグよりも機能が限定された単純なものである。 | webflow.tld |

表 C-1 ポータル開発 JSP タグ (名前順) (続き)

| タグ名 | 解説 | タグ ライブラリ |
|--|--|-------------|
| <code><webflow:getException></code> | Webflow プロセッサから送出された例外またはメッセージを取り出す。取り出し可能なのは、 <code>InvalidFormFieldException</code> 例外または <code>ProcessingException</code> 例外と関連付けられたメッセージ。このタグはインラインに埋め込む (その場合、例外に対して <code>getMessage()</code> メソッドを呼び出す) ことも、あるいは例外を表すスクリプト変数を返すようにすることもできる。 | webflow.tld |
| <code><webflow:getProperty></code> | Pipeline セッションから名前付きのプロパティを取得する。このタグはインラインに埋め込むことも、スクリプト変数を返すようにすることもできる。 | webflow.tld |
| <code><webflow:getValidatedValue></code> | Web サイトの訪問者が訂正しなければならないフォーム内のフィールドを表示する。 <code><webflow:getValidatedValue></code> タグは <code><webflow:setValidatedValue></code> タグと組み合わせて使われる。 | webflow.tld |
| <code><webflow:option></code> | HTML のオプションを表すが、入力内容を保持する。オプションはリストボックス内の 1 つの値である。このタグは <code><webflow:select></code> タグ内にネストする必要がある。 | webflow.tld |

表 C-1 ポータル開発 JSP タグ (名前順) (続き)

| タグ名 | 解説 | タグライブラリ |
|--|---|-------------|
| <code><webflow:password></code> | このタグは <code><webflow:text></code> タグと同様に動作するが、フィールドの入力内容がアスタリスク (****) でマスクされる。このタグは <code><webflow:validatedForm></code> タグ内にネストする必要がある。 | webflow.tld |
| <code><webflow:radio></code> | HTML のラジオ ボタンを表すが、入力内容を保持する。このタグは <code><webflow:validatedForm></code> タグ内にネストする必要がある。 | webflow.tld |
| <code><webflow:select></code> | リスト ボックスを表すが、その入力内容を保持する。このタグは <code><webflow:validatedForm></code> タグ内にネストする必要がある。 | webflow.tld |
| <code><webflow:setProperty></code> | Pipeline セッション内でプロパティを設定する。 | webflow.tld |
| <code><webflow:setValidatedValue></code> | Web サイトの訪問者が訂正しなければならないフォーム内のフィールドの表示形式を設定する。この設定は通常入力プロセッサ内で行うが、このタグを使って JSP で設定することもできる。 | webflow.tld |
| <code><webflow:text></code> | HTML のテキスト フィールドを検証する。このタグは <code><webflow:validatedForm></code> タグ内にネストする必要がある。 | webflow.tld |
| <code><webflow:textarea></code> | HTML のテキスト領域を表すが、入力内容を保持する。このタグは <code><webflow:validatedForm></code> タグ内にネストする必要がある。 | webflow.tld |

表 C-1 ポータル開発 JSP タグ (名前順) (続き)

| タグ名 | 解説 | タグ ライブラリ |
|--|--|--------------------------|
| <code><webflow:validatedForm></code> | HTML フォーム内でのアクションを定義する URL を動的に作成する。このタグは <code>com.bea.pl3n.appflow.webflow.forms.*</code> パッケージ、および、 <code>webflow.tld</code> ファイル内で定義されている、その他のネストされたフォーム タグとの組み合わせで使われる。 | <code>webflow.tld</code> |
| <code><wl:cache></code> | 表示のたびに更新する必要がないコンテンツであることを指定する。 | <code>wl.tld</code> |
| <code><wl:process></code> | 属性ベースのフロー制御のクエリを実行する。4 つの属性の組み合わせを使って、 <code><wl:process></code> タグと <code></wl:process></code> タグで囲まれる文を選択的に実行することができる。 | <code>wl.tld</code> |
| <code><wl:repeat></code> | <code>set</code> 属性で指定されるさまざまな Java オブジェクトに対して反復処理を行う。 | <code>wl.tld</code> |

