



Web アプリケーション フレームワーク タグライブラリリファレンス

Sun Java™ Studio Enterprise 7 2004Q4

Sun Microsystems, Inc.
www.sun.com

Part No. 819-1296-10
2004 年 12 月, Revision A

Copyright 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements. Use is subject to license terms.

この製品には第三者によって開発された成果物が含まれている場合があります。フロントテクノロジーを含むサードパーティ製のソフトウェアの著作権およびライセンスは、Sun Microsystems, Inc. のサプライヤが保有しています。

Sun、Sun Microsystems、Sun のロゴ、Java、JavaHelp、docs.sun.com、および Solaris は、米国および他の各国における Sun Microsystems, Inc. の商標または登録商標です。すべての SPARC の商標はライセンス規定に従って使用されており、米国および他の各国における SPARC International, Inc. の商標または登録商標です。SPARC の商標を持つ製品は、Sun Microsystems, Inc. によって開発されたアーキテクチャに基づいています。

UNIX は、X/Open Company Limited が独占的にライセンスしている米国ならびに他の国における登録商標です。

本製品はライセンス規定に従って配布され、本製品の使用、コピー、配布、逆コンパイルには制限があります。本製品のいかなる部分も、その形態および方法を問わず、Sun Microsystems, Inc. およびそのライセンサーの事前の書面による許可なく複製することを禁じます。

本製品は、米国輸出管理法の対象となっています。また、他国においても輸出入管理法の対象となっている場合があります。お客様は、それらのすべての法令および規制を厳守することに同意し、納品後に輸出、再輸出、または輸入の許可が必要となった場合には、お客様にそれらを取得する責任があるものとします。本製品を米国輸出規制法に指定されている各国または団体に提供することを禁じます。お客様は、本ソフトウェアが、核施設の設計、建設、運転または保守で使用するように設計、ライセンス、および意図されていないことを認識するものとします。Sun Microsystems, Inc. は、そのような目的の適合性に関して、明示的、黙示的を問わずいかなる保証も致しません。

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含み、明示的であるか黙示的であるかを問わず、あらゆる説明および保証は、法的に無効である限り、拒否されるものとします。

原典:	<i>Web Application Framework Tag Library Reference</i> Part No: 819-0730-10 Revision A
-----	--



Please
Recycle



Adobe PostScript

目次

はじめに ix

1. 概要 1

ライブラリの概要 1

タグライブラリと Web アプリケーションフレームワークの対話の仕組み 1

アプリケーションからのタグライブラリの利用 2

非 HTML JSP からのタグライブラリの利用 2

JSP コンテンツの取り込み : ページレット 3

タグの概要 5

コンテキストタグ 6

値タグ 6

可視 (HTML) タグ 6

ツリータグ 8

その他のタグ 8

タグリファレンス 8

2. containerView 9

<jato:containerView> 9

3. tiledView 11

<jato:tiledView> 11

4. treeView 13
 <jato:treeView> 13
5. useViewBean 15
 <jato:useViewBean> 15
6. getDisplayFieldValue 17
 <jato:getDisplayFieldValue> 17
7. getModelFieldValue 19
 <jato:getModelFieldValue> 19
8. setDisplayFieldValue 21
 <jato:setDisplayFieldValue> 21
9. setModelFieldValue 23
 <jato:setModelFieldValue> 23
10. button 25
 <jato:button> 25
11. checkbox 27
 <jato:checkbox> 27
12. combobox 31
 <jato:combobox> 31
13. fileUpload 33
 <jato:fileUpload> 33
14. form 37
 <jato:form> 37
15. frameSrc 41

- `<jato:frameSrc>` 41
- 16. hidden 43
 - `<jato:hidden>` 43
- 17. href 45
 - `<jato:href>` 45
- 18. image 49
 - `<jato:image>` 49
- 19. listBox 53
 - `<jato:listBox>` 53
- 20. password 57
 - `<jato:password>` 57
- 21. radioButton 61
 - `<jato:radioButtons>` 61
- 22. text 65
 - `<jato:text>` 65
- 23. textArea 69
 - `<jato:textArea>` 69
- 24. textField 73
 - `<jato:textField>` 73
- 25. validatingTextArea 77
 - `<jato:validatingTextArea>` 77
- 26. validatingTextField 81
 - `<jato:validatingTextField>` 81

- 27. content 85
 - <jato:content> 85
- 28. pagelet 89
 - <jato:pagelet> 89
- 29. treeNode 91
 - <jato:treeNode> 91
- 30. treeLevel 93
 - <jato:treeLevel> 93
- 31. treeNodeHandle 95
 - <jato:treeNodeHandle> 95
- 32. datasetLocator 97
 - <jatox:datasetLocator> 97
- 33. datasetNavigator 99
 - <jatox:datasetNavigator> 99
- 34. dateTimeView 101
 - <jatox:dateTimeView> 101
- 35. dateView 103
 - <jatox:dateView> 103
- 36. goToPage 105
 - <jatox:goToPage> 105
- 37. maskedTextField 107
 - <jatox:maskedTextField> 107
- 38. menu 111

	<jatox:menu>	111
39.	staticBreadCrumb	113
	<jatox:staticBreadCrumb>	113
40.	timeView	115
	<jatox:timeView>	115
	索引	117

はじめに

このマニュアルはタグライブラリに関する簡潔な入門書となっているばかりでなく、Web アプリケーションフレームワークのライブラリに用意されているタグに関する包括的なリファレンスにもなっています。

お読みになる前に

このマニュアルを読み始める前に、サーブレットや **JavaServlet™** ページ (**JSP™** ページ) などの既存の **J2EE Web** テクノロジーを利用した **Web** アプリケーションの構築で用いられている概念を理解しておくことを推奨します。

詳しい情報は、以下のリソースから得ることができます。

- **Java 2 Platform, Enterprise Edition Specification**
<http://java.sun.com/j2ee/download.html#platformspec>
- **J2EE Tutorial**
<http://java.sun.com/j2ee/tutorial>
- **Java Servlet Specification バージョン 2.3**
<http://java.sun.com/products/servlet/download.html#specs>
- **JavaServer Pages Specification バージョン 1.2**
<http://java.sun.com/products/jsp/download.html#specs>

注 - Sun では、本マニュアルに掲載されている第三者の Web サイトのご利用に關しましては責任はなく、保証するものでもありません。また、これらのサイトあるいはリソースに関する、あるいはこれらのサイト、リソースから利用可能であるコンテンツ、広告、製品、あるいは資料に關しても一切の責任を負いません。Sun は、これらのサイトあるいはリソースに関する、あるいはこれらのサイトから利用可能であるコンテンツ、製品、サービスの利用あるいはそれらのものを信頼することによって、あるいはそれに関連して発生するいかなる損害、損失、申し立てに対する一切の責任を負いません。

書体と記号について

書体または記号*	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例。	.login ファイルを編集します。 ls -a を実行します。 % You have mail.
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して表します。	% su Password:
<i>AaBbCc123</i> またはゴシック	コマンド行の可変部分。実際の名前や値と置き換えてください。	rm <i>filename</i> と入力します。 rm ファイル名 と入力します。
『 』	参照する書名を示します。	『Solaris ユーザーマニュアル』
「 」	参照する章、節、または、強調する語を示します。	第 6 章「データの管理」を参照。 この操作ができるのは「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅をこえる場合に、継続を示します。	% grep <code>``^#define \</code> <code>XV_VERSION_STRING</code>

* 使用しているブラウザにより、これら設定と異なって表示される場合があります。

関連マニュアル

Java Studio Enterprise のマニュアルとしては、Acrobat Reader (PDF) 形式のマニュアル、チュートリアルと、HTML 形式のリリースノート、オンラインヘルプ、チュートリアルが提供されています。

オンラインで入手可能なマニュアル

ここで紹介しているマニュアルは、docs.sun.comSM Web サイトおよび Sun Java Studio Enterprise Developers Source ポータルサイト (<http://developers.sun.com/jsenterprise>) のドキュメントリンクから入手できます。

docs.sun.com Web サイト (<http://docs.sun.com>) では、インターネットで Sun のマニュアルを参照、印刷、購入することができます。

- 『Sun Java Studio Enterprise 7 2004Q4 リリースノート』 - Part No. 819-1302-10
最新のリリースの変更点や技術的な注意事項を説明しています。

- 『Sun Java Studio Enterprise 7 インストールガイド』 (PDF 形式)
- Part No. 819-1300-10

サポートしている各プラットフォームへの Sun Java Studio Enterprise 7 統合開発環境 (IDE) のインストール方法を説明しています。システム要件やアップグレード方法、サーバー情報、コマンド行スイッチ、インストールされるサブディレクトリ、データベースの統合、アップデートセンターの使用方法などの関連情報も記載されています。

- 『J2EE アプリケーションのプログラミング』 - Part No. 819-1298-10

EJB モジュールや Web モジュールを J2EE にアSEMBルする方法を説明しています。また、J2EE アプリケーションの配備や実行についても説明しています。

- Web アプリケーションフレームワークのマニュアル (PDF 形式)

- 『Web アプリケーションフレームワーク コンポーネント作成ガイド』
- Part No. 819-1284-10

Web アプリケーションフレームワークのコンポーネントアーキテクチャと新しいコンポーネントの設計、作成、配布工程を説明しています。

- 『Web アプリケーションフレームワーク コンポーネントリファレンスガイド』
- Part No. 819-1286-10

Web アプリケーションフレームワークライブラリに提供されているコンポーネントを説明しています。

- 『Web アプリケーションフレームワーク 概要』 - Part No. 819-1288-10

Web アプリケーションフレームワークとその位置づけ、仕組み、他のアプリケーションフレームワークと異なる点を説明しています。

- 『Web アプリケーションフレームワーク チュートリアル』
- Part No. 819-1290-10

Web アプリケーションフレームワークを使用して Web アプリケーションを構築する際の仕組みとその手法を紹介しています。

- 『Web アプリケーションフレームワーク 開発ガイド』 - Part No. 819-1292-10

Web アプリケーションフレームワークを使用し、開発するアプリケーションの構成要素として使用可能なアプリケーションコンポーネントの作成および使用の手順と、そのアプリケーションを大部分の J2EE コンテナに配備する方法を説明しています。

- 『Web アプリケーションフレームワーク IDE ガイド』 - Part No. 819-1294-10

Sun Java Studio Enterprise 7 2004Q4 IDE の各部の概要、および Web アプリケーションフレームワークアプリケーションを開発するためのビジュアルツールの使用方法を重点的に説明しています。

- 『Web アプリケーションフレームワーク タグライブラリリファレンス』
- Part No. 819-1296-10

Web アプリケーションフレームワークのタグライブラリを簡単に紹介し、タグライブラリに提供されているタグに対する包括的な参照を示しています。

チュートリアル

Sun Java Studio Enterprise 7 には、IDE の機能を理解する手助けとなるチュートリアルがいくつか用意されています。これらのチュートリアルにある技術、およびコード例は、そのまま、または編集を加えて、実際のアプリケーションの開発に利用することができます。すべてのチュートリアルで、Sun Java System Application Server への配備例が紹介されています。

チュートリアルは、すべて **Developers Source** ポータルのリンク「**Tutorials & Code Camps**」から利用可能です。IDE で「ヘルプ」>「コードサンプルとチュートリアル」>「概要」を選択すると、このサイトにアクセスできます。

- 「クイックスタートガイド」は、Sun Java Studio IDE の紹介をしています。チュートリアルは、Sun Java Studio を初めてご使用になる方や、特定の機能について早く知りたい場合は、このガイドから始めてください。これらのチュートリアルは、単純な Web アプリケーションや J2EE アプリケーションの開発方法、Web サービスの生成方法を説明しています。また、UML モデリング、リファクタリングの導入方法についても説明しています。ガイドを終えるための所要時間は数分です。

- 「チュートリアル」は、Sun Java Studio IDE の特定の 1 つの機能に焦点を当てています。ある機能の詳細に関心がある場合は、これらを実行してみてください。例で説明している機能によって、初めからアプリケーションを構築する場合と、提供されたソースファイルを使用して構築する場合があります。チュートリアルは 1 時間以内で完成できます。
- 「概要ビデオ」は、技術の説明がビデオで提供されています。IDE の視覚的な概要や、特定の機能の詳細説明を見ることができます。概要ビデオにかかる時間は数分です。概要ビデオは、任意の個所で開始、終了することもできます。

オンラインヘルプ

Sun Java Studio Enterprise 7 IDE には、オンラインヘルプが用意されています。ヘルプキー (Microsoft Windows 環境では F1 キー、Solaris オペレーティング環境では Help キー) を押すか、「ヘルプ」>「ヘルプ (すべて)」を選択して開くことができます。ヘルプの項目と検索機能が表示されます。

アクセシブルな製品マニュアル

マニュアルは、技術的な補足をすることで、ご不自由なユーザーの方々にとって読みやすい形式のマニュアルを提供しております。アクセシブルなマニュアルは以下の表に示す場所から参照することができます。

マニュアルの種類	アクセシブルな形式と格納場所
マニュアルとチュートリアル	形式: HTML 場所: http://docs.sun.com
チュートリアル	形式: HTML 場所: Developers Source ポータル (http://developers.sun.com/jsenterprise) のリンク 「Examples & Code Camps」
リリースノート	形式: HTML 場所: http://docs.sun.com

第1章

概要

ライブラリの概要

Web アプリケーションフレームワークタグライブラリは、Web アプリケーションフレームワークと組み合わせたときに動的な JSP 出力を生成する強力なツールです。このマニュアルはタグライブラリに関する簡潔な入門書となっているばかりでなく、ライブラリに用意されているタグに関する包括的なリファレンスにもなっています。Web アプリケーションフレームワークアプリケーションの作成に関するさらに詳しい情報は、増補版の Web アプリケーションフレームワークのマニュアルを参照してください。

タグライブラリと Web アプリケーションフレームワークの対話の仕組み

Web アプリケーションフレームワークは、任意の入れ子のビューオブジェクトという考え方に基づいて設計されています。こうしたビューには、他のビュー（「コンテンツビュー」自体も含む）を含む「コンテンツビュー」というビューもあれば、アクセス可能な値という考え方を持つ「表示フィールド」というビューもあります。

それぞれの JSP は、関連付けられている ViewBean を 1 つだけ持ちます。このオブジェクトは、その JSP で使用される他のすべてのビューを含む最上位のビューであるため、JSP の「ルートビュー」ともいいます。Web アプリケーションフレームワークタグライブラリ内の各タグは、ルートビューに含まれているビューの 1 つに対する参照です。タグが参照するオブジェクトは、「ピアビュー」あるいは単に「ピア」とい

います。開発者が自身の JSP で使用する各タグは、その JSP の ViewBean または ViewBean の子ビュー内の既存の 1 つのピアオブジェクトを参照する必要があります。

Web アプリケーションフレームワークタグライブラリに含まれるタグの大半は、その中で他のピアを簡単に参照することが可能なコンテキストを構築するか、実際にピアビューを参照して、文字列値やテキストフィールド、リストボックスなどの可視の HTML フラグメントを生成します。タグライブラリを使用した JSP が生成されると、そこには、可視の HTML フォームのコントロールや、各タグ内の中からの各ピアとの対話で派生したその他コンテンツが含まれます。つまり、タグは事前に定義された一群のピアと協力し、JSP から動的な HTML ページを生成します。

アプリケーションからのタグライブラリの利用

Web アプリケーションから Web アプリケーションフレームワークタグライブラリを使用するには、タグライブラリを使用するそれぞれの JSP で Web アプリケーションフレームワークタグライブラリ記述子 (TLD) を参照する必要があります。この参照は、JSP の先頭に以下の命令を含めることによって行います。

```
<%@taglib uri="/WEB-INF/jato.tld" prefix="jato"%>
```

(注 : Web アプリケーションフレームワーク IDE を使用している場合は、この行は自動的に追加されます。) URI 属性で TLD の場所およびタグ接頭辞名は任意ですが、上記の値を使用し、アプリケーションの WAR ファイルの WEB-INF ディレクトリのルートに Web アプリケーションフレームワーク TLD ファイルを置くことを推奨します。このマニュアルでは、この JATO タグ接頭辞を前提にします。

非 HTML JSP からのタグライブラリの利用

Web アプリケーションフレームワークタグライブラリには、可視および不可視両方のタグが含まれています。可視タグは HTML 4.01 準拠のマークアップを生成しますが、不可視タグには関連付けられているマークアップがなく、生成するマークアップを自分で指定することができます。不可視タグはいくつかのカテゴリに分類されますが、一般に、抽象的な方法で Web アプリケーションフレームワークビューの構造を表します。この階層構造は異なるコンテンツの種類にまたがって維持されるため、適切なコンテンツの種類に対して可視マークアップが指定されているという条件を満たさずれば、Web アプリケーションフレームワークを使用し、HTML 以外のマークアップを使った JSP を開発することができます。

既存の `getDisplayFieldValue` タグは、表示イベントをトリガーし、現在のコンテナビューコンテキストと連携しながら、関連付けられたマークアップなしで動的な表示フィールドデータを簡単にインライン化する手段を提供します。このタグによって、開発者は JSP で宣言した静的マークアップに動的な値を追加することができます。

す。このタグは、新しい JSP タグを作成することなく非 HTML ページ (HTML ページも可) を開発する簡単な手段です。ただし、一般に、非 HTML JSP を生成する方法としては、新しい JSP タグを作成する方が簡単です。

Sun ONE Application Framework 2.0 からは、タグが標準の Web アプリケーションフレームワークタグライブラリのタグが独自に作成したタグであるに関係なく、開発者はコンポーネントライブラリを使用し、複数のコンテンツタイプに対する独自のタグを自動的に管理するコンポーネントを簡単にパッケージ化して、提供することができます。また、Sun ONE Application Framework 2.1 からは、Web アプリケーションフレームワークの taglib パッケージが再実装され、任意の方法でビューコンポーネントを生成する新しいカスタムタグをずっと簡単に作成できるようになっています。開発者はこれらのツールを利用して、必要とするコンテンツタイプの可視マークアップを処理する既存のコンポーネントライブラリを見つけたり、HTML 以外のコンテンツタイプを使用するコンポーネントを生成するコンポーネントライブラリを作成したりすることが簡単に行えるようになります。独自の可視タグの作成についての詳細は、taglib パッケージのマニュアルを参照してください。

JSP コンテンツの取り込み：ページレット

JSP では、現在生成中のページに外部のコンテンツを取り込むことができます (詳細は、JSP 1.1 の仕様書を参照)。このため開発者は、JSP コンテンツをモジュール化し、そのモジュールを複合ドキュメントに組み込むことができます。この機能は、Web アプリケーションフレームワークアプリケーションと密接な関わりを持ちます。

JSP にコンテンツを取り込む方法は、変換時取り込みと要求時取り込みの 2 通りあります。変換時取り込みは、コンテナによってサーブレットに変換される前に取り囲む JSP に外部のファイルの内容を取り込みます。この方法の利点は、パフォーマンスに優れ、取り込まれた内容が、取り囲むページの一部であるかのように機能することにあります。欠点は必ずしも動的でないこと、すなわち、取り込まれた内容が静的に囲まれて、実行時に選択あるいは置換できないことにあります。

要求時取り込みは、対象となる URL/JSP に対して `RequestDispatcher.include()` 操作を行って、取り囲むページにそのコンテンツを動的にインライン化します。この取り込みは、取り囲むページが生成されるたびに行われます。この方法の利点は、取り込みファイルを動的に選択できることにあります。欠点は、取り込みリソースに要求を送信する際に多少パフォーマンスのオーバーヘッドがあることです。

Web アプリケーションフレームワークアプリケーションへの JSP フラグメントの取り込みで一般に推奨されているのは、「変換時取り込み」です。これは、パフォーマンスに優れ、取り込みの範囲に適したビュー / フィールドだけを取り込めるためです。この手法により、コンポーネント作成者は JSP ページまたは Web アプリケーションフレームワークコンテンツのフラグメントを作成し、いくつかのページで

そのフラグメントを再利用できます。ただし、上記で示したように、取り囲むページに取り込むフラグメントは変換時に決定されるため固定であり、生成するページのダイナミズムは大幅に制限されることとなります。

Web アプリケーションフレームワークアプリケーションでの要求時取り込みは、「変換時取り込み」の静的の問題を回避するよい手段になります。しかし、以前のバージョンの **Web** アプリケーションフレームワークでは、**ViewBean** 全体とそのコンテンツをまるごと取り込む以外に要求時取り込みを行う方法がありませんでした。つまり、基本的にルートビューの内側に別のルートビューを取り込む必要があり、混乱する上に、必ずしも正しく機能することが保証されていません。ルートビューの個数は論理ページ1ページに1つだけであるべきです。この方法のもう1つの欠点としては、取り囲む **JSP** のコンテナビューの範囲を取り込まれたコンテンツが見失う、つまり取り込まれたページ内でアクセスされるオブジェクトの範囲が、直接取り囲む **ViewBean** によって制限されることにあり、このことは、変換時取り込みのように、必ずしも他の **JSP** に自由にページを取り込めないことを意味します。

Sun ONE Application Framework 1.2 以降は、「ページレット」およびそれに付属する `<jato:pagelet>` タグの形でこうした制限に対する解決策が用意されています。**Web** アプリケーションフレームワークでは、ページレットは **Web** アプリケーションフレームワーク **JSP** ページにシームレスに要求時取り込みができるという点で、ページレット以外のあらゆる **JSP** フラグメントと区別されます。この要求時取り込みは、`<jato:pagelet>` タグを使用することによって実現され、このタグは、取り込まれた **JSP** 内のすべてのビュータグを取り囲む **JSP** のコンテナの範囲に「接続」し、取り込まれたページレットが、そのコンテナビュースコープとして取り囲む **JSP** の現在のコンテナビューを使用するようにします。

たとえば以下の **JSP** とページレットが存在すると仮定します。

EnclosingPage.jsp

```
<%@page info="E0130" language="java"%>
<%@taglib uri="/WEB-INF/jato.tld" prefix="jato"%>

<jato:useViewBean ...>

<jato:containerView name="foo">
    ...
    <jsp:include page="MyPagelet.jsp"/>
    ...
</jato:containerView>

</jato:useViewBean>
```

MyPagelet.jsp

```
<%@page info="MyPagelet" language="java"%>
<%@taglib uri="/WEB-INF/jato.tld" prefix="jato"%>

<jato:pagelet>
    <jato:combobox name="month"/>...
</jato:pagelet>
```

「month」フィールドが「foo」コンテナビューをそれを取り囲むコンテナと見なし、こうして「month」フィールドは「foo」の適切な子として宣言することができます。<jato:pagelet> タグは、取り囲むページのコンテナビュータグのプロキシとして機能し、実際にはそれぞれ独立した 2 つの生成対象であっても、同じ生成対象の一部であるかのように 2 つの JSP を接続します。

さらに、開発者は Web アプリケーションフレームワークで表示イベントを使用して、要求時に取り込むページレットを動的に選択することができます (イベントでは `PageContext.include()` を使用)。この方法の実例は、Web アプリケーションフレームワークサンプルアプリケーションにあります。

次の表は、Web アプリケーションフレームワークで利用可能な取り込み方法をまとめています。

JSP フラグメントの種類	変換時取り込み	実行時取り込み
ページレット	○*	○
フラグメント (ViewBean なし)	○	×
フラグメント (ViewBean あり)	×	○

* 技術的には問題ありませんが、取り込み JSP が単に取り囲む JSP の一部になるだけで、一貫性の維持という以外に、この状況で <jato:pagelet> タグを使用する理由はありません。ページレットタグは、この状況では何も行いません。

タグの概要

Web アプリケーションフレームワークタグライブラリに含まれるタグは、コンテキストタグ、値タグ、可視タグの 3 つの基本グループに分類されます。詳細は、この後の項を参照してください。

コンテキストタグ

コンテキストタグは、ピアビューのスキープの宣言を目的としていて、そのスキープ内で参照オブジェクトは、他の埋め込みタグに対する現在のコンテキストを定義します。具体的には、このことは、タグはそれぞれコンテナビュー (または特殊なコンテナビュー) の使用を宣言することを意味します。そうした宣言はそれぞれに、短い非限定名で子ビューを参照することが可能な範囲を示すネームスコープを確定します。コンテナビューコンテキストは入れ子にすることができ、直接には HTML を生成しません。

タグ名	代替名	説明
containerView		コンテナビューの使用を宣言
tiledView		タイルビューの使用を宣言
treeView		ツリービューの使用を宣言
useViewBean	viewBean	ViewBean の使用を宣言

値タグ

値タグでは、表示フィールドおよびモデルの値を直接操作することができます。これらのタグは、JSP 内のスクリプトレットや式、HTML に埋め込むことができます。この後に説明すると可視タグと異なり、値タグは HTML フォームのコントロールを生成しません。値タグは値を直接操作します。値タグを使用して、任意の表示フィールドのサブタイプ、あるいはアプリケーションで使用可能な任意のモデルインスタンスで値を取得 / 設定することができます。値タグは、コンテキストタグのスキープ内に現れる必要があります。

タグ名	説明
getDisplayFieldValue	表示フィールドから値を取り出す。
getModelFieldValue	モデルフィールドから値を取り出す。
setDisplayFieldValue	表示フィールドの値を設定する。
setModelFieldValue	モデルフィールドの値を設定する。

可視 (HTML) タグ

可視タグは、タグ属性と表示フィールドのピアの値を組み合わせたものを使用して、HTML フォームのコントロールを生成します。可視タグは、コンテキストタグのスキープ内に現れる必要があります。

タグ名	代替名	説明
button		ボタンコントロールを生成する。
checkbox	checkboxBox	チェックボックスコントロールを生成する。
comboBox	comboBox	コンボボックスコントロールを生成する。
fileUpload		ファイルアップロード要素を生成する。
form		HTML フォームを定義する。
frameSrc		フレームソース要素を生成する。
hidden		非表示要素を生成する。
href		ハイパーリンク要素を生成する。
image		イメージを生成する。
listbox	listBox	リストコントロールを生成する。
password		パスワードコントロールを生成する。
radioButtons		一群のラジオボタンコントロールを生成する。
text	staticText	任意のテキストを生成する。
textArea		複数行検査テキスト領域コントロールを生成する。
textField		単一行検査テキストコントロールを生成する。
validatingTextArea		複数行検査テキスト領域コントロールを生成する。
validatingTextField		単一行検査テキストコントロールを生成する。
datasetLocator		現在の場所の静的なテキスト表示と現在表示されているレコード数を生成する。
datasetNavigator		「先頭」、「前」、「次」、「最後」の4つのデータセットナビゲーションアクションを生成する。
dateTimeView		月、日、年の範囲、時、分、am/pm オプションを生成する。
dateView		月、日、年の範囲のオプションを生成する。
goToPage		可視で、クリックすることができる href 要素を生成する。
maskedTextField		テキストフィールド要素を生成する。
menu		メニュー要素を生成する。
staticBreadCrumb		bread crumb 要素を生成する。
timeView		3つのコンボボックス（時、分、am/pm オプション）を生成する。

ツリータグ

ツリータグを組み合わせて、ツリー生成ロジックを指定することができます。

タグ名	説明
treeLevel	一致するツリーレベルに対してコンテンツ部をレベル回数生成することを示す。
treeNode	一致するツリーノードに対してコンテンツ部を生成することを示す。
treeNodeHandle	ノード展開/縮小コントロールを生成する。

その他のタグ

その他のタグは、追加機能を提供します。

タグ名	説明
content	コンテンツセクションを示し、表示イベントに関連付ける。
pagelet	取り込まれた JSP フラグメント内の Web アプリケーションフレームワークタグが取り囲むページのコンテナビュースコープを継承して、要求時に取り込まれることを可能にする。

タグリファレンス

注の用語説明：

「RT 式」 = 属性値として実行時評価式の指定が可能

「必須」 = 必須属性

containerView

<jato:containerView>

containerView は、ビュー Bean またはその子の ContainerViews 1 つから使用可能な containerView ピアインスタンスの使用を宣言します。このタグで囲まれたビューまたは表示フィールドの参照はすべて、関連付けられている containerView の子に対する参照と見なされ、その name 属性は、親コンテナの名前空間内で解決されます。このタグは、useViewBean か tiledView、または別の containerView タグの入れ子になっている場合にのみ有効です。

containerView タグを使用すると、取り囲むタグ本体のスコープ内に新しいスクリプト変数が定義されます。このスクリプト変数の名前は name 属性の値か、id 属性が指定されている場合は、その ID 名です。containerView タグの処理では、暗黙のスクリプト変数 currentContainerView に、このタグが参照する containerView インスタンスが設定されます。

useViewBean および tiledView タグは特殊なケースの containerView タグで、このタグが定義する基本動作に別の動作を追加します。

例：

```
<jato:containerView name="header">
...
</jato:containerView>
```

属性名	説明	注
id	<p>ContainerView インスタンスを参照する暗黙のスクリプト変数の識別に使用する名前を指定する。指定した名前の英大文字と小文字は区別され、最新のスクリプト言語変数命名規則に従っていること。</p> <p>省略された場合は、name 属性に指定された値を使って名前が付けられる。</p>	
name	<p>親 (ContainerView、TiledView、ViewBean どれか) に宣言されている ContainerView ピアインスタンスの名前。</p> <p>この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照していると思なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。</p> <p>例：</p> <p>/header/orderList/customerName (ルートビューからの絶対位置)</p> <p>orderList/customerName (現在のコンテナからの相対位置)</p> <p>../footer/orderList/customerName (親からの相対位置)</p>	必須、RT 式
type	<p>指定された場合は、このタグの範囲内に定義されているスクリプト変数の型を定義する。この定義によって、関係はするが、指定された実装クラスの型とスクリプト変数の型を区別できる。型は、この特定の ContainerView クラスそのものか、このクラスのスーパークラス、指定されたクラスが実装するインターフェースのいずれかである必要がある。参照先のオブジェクトはこの型である必要がある。この型でない場合、要求時、スクリプト変数への参照されたオブジェクトの割り当てが試みられたときに、java.lang.ClassCastException が発生する。</p> <p>省略された場合は、スクリプト変数は <JATO package>.view.ContainerView 型と宣言される。</p>	

第3章

tilableView

<jato:tilableView>

tilableView タグは、JSP の表示中に繰り返し評価、生成する本体内容のブロックを宣言します。tilableView は、ビュー Bean またはその子の containerView の 1 つから使用可能な TiledView オブジェクトに関連付ける必要があります。TiledView オブジェクトへのコールバックでは、tilableView 本体内容の表示を制御します。

tilableView は、HTML 表で複数の行を表示する際に特に有用ですが、その用途は動的な HTML 行の生成に限りません。

tilableView タグを使用すると、取り囲むタグ本体の範囲内に新しいスクリプト変数が定義されます。このスクリプト変数の名前は name 属性の値か、id 属性が指定されている場合は、その ID 名です。tilableView タグの処理では、暗黙のスクリプト変数 currentContainerView と currentTiledView に、このタグが参照する TiledView インスタンスが設定されます。

tilableView タグは、特殊なケースの containerView タグと見なすことができます。このタグは、useViewBean か containerView、または別の tilableView タグの入れ子になっている場合にのみ有効です。

例：

```
<jato:tilableView name="orderItems" maxTiles="10">
...
</jato:tilableView>
```

属性名	説明	注
id	<p>TiledView インスタンスを参照する暗黙のスクリプト変数の識別に使用する名前を指定する。指定した名前の英大文字と小文字は区別され、最新のスクリプト言語変数名規則に従っていること。</p> <p>省略された場合は、name 属性に指定された値を使って名前が付けられる。</p>	
maxTiles	<p>有効な値：正数。</p> <p>この TiledView がその本体内容の生成を繰り返す最大回数を指定する。この値は、TiledView オブジェクトの対応する値に優先する。省略された場合、maxTiles は TiledView ピアによって決まる。</p>	RT 式
name	<p>親 (ContainerView、TiledView、ViewBean どれか) に宣言されている TiledView ピアインスタンスの名前。</p> <p>この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か containerView から派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照していると見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。</p> <p>例：</p> <p>/header/orderList/customerName (ルートビューからの絶対位置)</p> <p>orderList/customerName (現在のコンテナからの相対位置)</p> <p>../footer/orderList/customerName (親からの相対位置)</p>	必須、RT 式
type	<p>指定された場合は、このタグの範囲内に定義されているスクリプト変数の型を定義する。この定義によって、関係はするが、指定された実装クラスの型とスクリプト変数の型を区別できる。型は、この特定の TiledView クラスそのものか、このクラスのスーパークラス、指定されたクラスが実装するインタフェースのいずれかである必要がある。参照先のオブジェクトはこの型である必要がある。この型でない場合、要求時、スクリプト変数への参照されたオブジェクトの割り当てが試みられたときに、java.lang.ClassCastException が発生する。</p> <p>省略された場合は、スクリプト変数は <JATO package>.view.TiledView 型と宣言される。</p>	
reset	<p>有効な値：「true」または「false」</p> <p>値「true」は、beginDisplay() イベントの起動直後に TiledView の resetTileIndex() メソッドを呼び出す指定になる。これで、表示に際して、ビューの一次モデルが適切にリセットされる。</p> <p>省略された場合は、値「true」と見なされる。これは、以前のバージョンの Web アプリケーションフレームワークからの変更であることに注意。以前のバージョンでは、リセットは行われず、この動作は開発者に任されていた。この変更は、まれに存在する以前のデフォルトの動作に依存しているアプリケーションに対し、下位互換性を提供する一方で、使い易さを高めるものと考えられる。</p>	

第4章

treeView

<jato:treeView>

`treeView` タグは、JSP の表示中に繰り返し評価、生成する本体の内容のブロックを宣言します。`treeView` は、ビュー Bean またはその子の `ContainerView` の 1 つから使用可能な `TreeView` オブジェクトに関連付ける必要があります。`TreeView` オブジェクトへのコールバックでは、`treeView` 本体内容の表示を制御します。`treeView` は、階層データの表示に特に有用です。定義されている訳ではありませんが、一般に、`treeView` タグには、1 つ以上の `treeNode` タグが含まれます。

`treeView` タグを使用すると、取り囲むタグ本体の範囲内に新しいスクリプト変数が定義されます。このスクリプト変数の名前は `name` 属性の値か、`id` 属性が指定されている場合は、その ID 名です。`treeView` タグの処理では、暗黙のスクリプト変数 `currentContainerView` と `currentTreeView` に、このタグの参照する `TreeView` インスタンスが設定されます。

`treeView` タグは、特殊なケースの `containerView` タグと見なすことができます。このタグは、`useViewBean` や `containerView`、`tiledView`、別の `tiledView` タグの入れ子になっている場合にのみ有効です。

関連項目 : `treeNode`、`treeLevel`、`treeNodeHandle`

例 :

```
<jato:treeView name="treeMenu">
...
</jato:treeView>
```

属性名	説明	注
id	<p>ContainerView インスタンスを参照する暗黙のスクリプト変数の識別に使用する名前を指定する。指定した名前の英大文字と小文字は区別され、最新のスクリプト言語変数名規則に従っていること。</p> <p>省略された場合は、name 属性に指定された値を使って名前が付けられる。</p>	
name	<p>その親コンテナビュー内で宣言されている TreeView ビアインスタンスの名前。この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照していると見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。</p> <p>例：</p> <p>/header/orderList/customerName (ルートビューからの絶対位置)</p> <p>orderList/customerName (現在のコンテナからの相対位置)</p> <p>../footer/orderList/customerName (親からの相対位置)</p>	必須、RT 式
type	<p>指定された場合は、このタグの範囲内に定義されているスクリプト変数の型を定義する。この定義によって、関係はするが、指定された実装クラスの型とスクリプト変数の型を区別できる。型は、この特定の ContainerView クラスそのものか、このクラスのスーパークラス、指定されたクラスが実装するインタフェースのいずれかである必要がある。参照先のオブジェクトはこの型である必要がある。この型でない場合、要求時、スクリプト変数への参照されたオブジェクトの割り当てが試みられたときに、java.lang.ClassCastException が発生する。</p> <p>省略された場合、スクリプト変数は <JATO package>.view.TreeView 型と宣言される。</p>	
reset	<p>有効な値：「true」または「false」</p> <p>値「true」は、beginDisplay() イベントの起動直後に TreeView の resetNodeLocation() メソッドを呼び出す指定になる。これで、表示に際して、ビューの一次モデルが適切にリセットされる。</p> <p>省略された場合は、値「true」と見なされる。これは、以前のバージョンの Web アプリケーションフレームワークからの変更であることに注意。以前のバージョンでは、リセットは行われず、この動作は開発者に任されていた。この変更は、まれに存在する以前のデフォルトの動作に依存しているアプリケーションに対し、下位互換性を提供する一方で、使い易さを高めるものと考えられる。</p>	

第5章

useViewBean

<jato:useViewBean>

`useViewBean` タグは、現在の JSP の生成に使用するルートの `ViewBean` ピアを作成します。1 つの JSP 内で使用できる `useViewBean` タグは 1 つだけで、そのページ上の他のすべての Web アプリケーションフレームワークタグはこのタグで囲む必要があります。`useViewBean` タグは JSP のルート名前空間を作成し、ページの生成中、コントローラとして機能します。

`useViewBean` タグを使用すると、取り囲まれたタグ本体のスコープ内に、`viewBean` という新しいスクリプト変数が定義されます。`useViewBean` タグの処理では、暗黙のスクリプト変数 `currentContainerView` と `currentTiledView` が作成されます。`currentContainerView` 変数には、このタグが参照する `ViewBean` インスタンスが設定されます。

`useViewBean` タグは、特殊なケースの `containerView` タグと見なすことができます。

例：

```
<jato:useViewBean className="com.mycomp.myapp.MyViewBean">  
...  
</jato:useViewBean>
```

属性名	説明	注
className	ViewBean ピアクラスの完全限定名。クラス名は英大文字と小文字が区別される。	必須
type	指定された場合は、このタグの範囲内に定義されている viewBean スクリプト変数の型を定義する。この定義によって、関係はするが、指定された実装クラスの型とスクリプト変数の型を区別できる。型は、この特定の ViewBean クラスそのものか、このクラスのスーパークラス、指定されたクラスが実装するインタフェースのいずれかである必要がある。参照先のオブジェクトはこの型である必要がある。この型でない場合、要求時、スクリプト変数への参照されたオブジェクトの割り当てが試みられたときに、 <code>java.lang.ClassCastException</code> が発生する。 省略された場合、スクリプト変数は <code>className</code> 属性で指定された型と宣言される。	

第6章

getDisplayFieldValue

<jato:getDisplayFieldValue>

このタグは、指定された表示フィールドで `DisplayField.getValue()` メソッドを起動し、HTML 出力ストリームに結果値をインライン化します。このタグは、HTML フォームのコントロールを手動で作成したり、実行時値式をとることが可能な他のタグ属性に値を提供したりする目的で、任意の HTML 内に埋め込むことができます。

このタグは、`useViewBean` か `containerView`、または `tiledView` タグで囲まれている場合にのみ有効です。このタグに本体内容を含めることはできません。

例：

```
<jato:getDisplayFieldValue name="salutation" defaultValue="Mr." />
```

属性名	説明	注
defaultValue	DisplayField ピア値が null の場合に表示する値。	RT 式
escape	有効な値: 「true」または「false」 値「true」は、出力する前に HTML 内で特殊な意味を持つ文字 (< や > など) を同等のエンティティ表現に変更する指定になる。 省略された場合は、値「true」と見なされる。	
name	DisplayField ピアの名前。このピア、現在の親の ContainerView、TiledView、ViewBean いずれかの子である必要がある。DisplayField 名は、現在の親ビューを基準に解決される。 この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照していると見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。 例: /header/orderList/customerName (ルートビューからの絶対位置) orderList/customerName (現在のコンテナからの相対位置) ../footer/orderList/customerName (親からの相対位置)	必須、RT 式

getModelFieldValue

<jato:getModelFieldValue>

このタグは、指定されたモデルインスタンスで `Model.getValue()` メソッドを起動し、HTML 出力ストリームに結果値をインライン化します。このタグは、HTML フォームのコントロールを手動で作成したり、実行時値式をとることが可能な他のタグ属性に値を提供したりする目的で、任意の HTML 内に埋め込むことができます。

このタグは、`useViewBean` か `containerView`、または `tiledView` タグで囲まれている場合にのみ有効です。このタグに本体内容を含めることはできません。

例：

```
<jato:getModelFieldValue modelClass=
  "com.mycompany.myapp.MyModel" name="salutation" defaultValue=
  "Mr." />
```

属性名	説明	注
<code>defaultValue</code>	Model フィールド値が <code>null</code> の場合に表示する値。	RT 式
<code>escape</code>	有効な値：「 <code>true</code> 」または「 <code>false</code> 」 値「 <code>true</code> 」は、出力する前に HTML 内で特殊な意味を持つ文字 (< や > など) を同等のエンティティ表現に変更する指定になる。 省略された場合は、値「 <code>true</code> 」と見なされる。	
<code>lookInSession</code>	有効な値：「 <code>true</code> 」または「 <code>false</code> 」 値「 <code>true</code> 」は、 <code>modelName</code> 属性が指定するモデルをセッションで探すようにという <code>ModelManager</code> に対する指定になる。この属性を指定した場合は、 <code>modelName</code> 属性も指定する必要がある。 省略された場合は、「 <code>false</code> 」と見なされる。	RT 式

属性名	説明	注
modelClass	完全限定のモデルクラスインタフェースか実装名 (指定するクラスの詳細は、 <code><JATO package>.ModelManager</code> のマニュアルを参照)。このクラス名に対する応答として <code>ModelManager</code> から返されるインスタンスは、 <code><JATO package>.model.Model</code> インタフェースを実装する必要がある。	必須、 RT 式
modelName	この属性が指定された場合は、その属性値が示すモデルの取得を試みる。省略された場合は、デフォルトのモデルインスタンスが使用される。 <code>lookInSession</code> 属性を「true」に設定した場合、この属性は必須。 <code>lookInSession</code> 属性が「true」の場合、 <code>ModelManager</code> は、この属性で指定された名前を使って、セッションからモデルを取得しようとする。	RT 式
name	モデルフィールドの名前。	必須、 RT 式

setDisplayFieldValue

<jato:setDisplayFieldValue>

このタグは、指定された値で `DisplayField.setValue(Object)` メソッドを起動します。このタグが HTML を生成することはありません。ページの任意の HTML 内に埋め込むことができます。

このタグは、`useViewBean` か `containerView`、または `tiledView` タグで囲まれている場合にのみ有効です。このタグに本体内容を含めることはできません。

例：

```
<jato:setDisplayFieldValue name="numItems" value="8"
valueType="int" />
```

属性名	説明	注
name	<p>DisplayField ピアの名前。このピア、現在の親の ContainerView、TiledView、ViewBean いずれかの子である必要がある。DisplayField 名は、現在の親ビューを基準に解決される。</p> <p>ビューピアは、<JATO package>.view.DisplayField 型である必要がある。この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照していると見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。</p> <p>例：</p> <ul style="list-style-type: none"> /header/orderList/customerName (ルートビューからの絶対位置) orderList/customerName (現在のコンテナからの相対位置) ../footer/orderList/customerName (親からの相対位置) 	必須、RT 式
value	<p>指定された DisplayField ピアの DisplayField.setValue(Object) メソッドに渡す値。</p>	必須、RT 式
valueType	<p>有効な値：「string」、「int」、「long」、「float」、「double」、「short」、「bigdecimal」、「boolean」、「byte」、「char」、「sqldate」、「sqltime」、「sqltimestamp」</p> <p>DisplayField に設定する前の value 属性の文字列表現の変換後の型。文字列表現からの型変換は正当である必要があり、そうでない場合は、例外がスローされる。</p> <p>この属性が省略された場合、String 型と見なされる。</p>	

setModelFieldValue

<jato:setModelFieldValue>

このタグは、指定された値で `Model.setValue(String, Object)` メソッドを起動します。このタグが HTML を生成することはありません。ページの任意の HTML 内に埋め込むことができます。

このタグは、`useViewBean` か `containerView`、または `tiledView` タグで囲まれている場合にのみ有効です。このタグに本体内容を含めることはできません。

例：

```
<jato:setModelFieldValue modelClass=
  "com.mycompany.myapp.MyModel" name="numItems" value="8"
  valueType="int" />
```

属性名	説明	注
<code>lookInSession</code>	有効な値：「true」または「false」 値「true」は、 <code>modelName</code> 属性が指定するモデルをセッションで探すようにという <code>ModelManager</code> に対する指定になる。この属性を指定した場合は、 <code>modelName</code> 属性も指定する必要がある。 省略された場合は、「false」と見なされる。	RT 式
<code>name</code>	<code>DisplayField</code> ピアの名前。このピア、現在の親の <code>ContainerView</code> 、 <code>TiledView</code> 、 <code>ViewBean</code> いずれかの子である必要がある。 <code>DisplayField</code> 名は、現在の親ビューを基準に解決される。 ビューピアは、 <code><JATO package>.view.DisplayField</code> 型である必要がある。	必須、 RT 式
<code>value</code>	指定された <code>DisplayField</code> ピアの <code>DisplayField.setValue(Object)</code> メソッドに渡す値。	必須、 RT 式

属性名	説明	注
modelClass	完全限定のモデルクラスインタフェースか実装名 (指定するクラスの詳細は、 <JATO package>.ModelManager のマニュアルを参照)。このクラス名に対する応 答として ModelManager から返されるインスタンスは、 <JATO package>.model.Model インタフェースを実装する必要がある。	必須、 RT 式
modelName	この属性が指定された場合は、その属性値が示すモデルの取得を試みる。省略され た場合は、デフォルトのモデルインスタンスが使用される。lookInSession 属性 を「true」に設定した場合、この属性は必須。lookInSession 属性が「true」の場 合、ModelManager は、この属性で指定された名前を使って、セッションからモデル を取得しようとする。	RT 式
valueType	有効な値: 「string」、「int」、「long」、「float」、「double」、「short」、 「bigdecimal」、「boolean」、「byte」、「char」、「sqldate」、「sqltime」、 「sqltimestamp」 DisplayField に設定する前の value 属性の文字列表現の変換後の型。文字列表 現からの型変換は正当である必要があり、そうでない場合は、例外がスローされ る。 この属性が省略された場合、String 型と見なされる。	

第10章

button

<jato:button>

HTML の `<input type="submit">` 要素を生成します。

このタグは、HTML の `<form>` 要素や `useViewBean`、`containerView`、`tiledView` タグで囲まれている場合にのみ有効です。このタグに本体内容を含めることはできません。

例：

```
<jato:button name="processForm" />
```

これは、次の HTML になります。

```
<input type="submit" name="PageFoo.processForm" value="Process">
```

属性名	説明	注
<code>accessKey</code>	生成される HTML 要素にアクセスキーを割り当てる。一般には、このアクセスキーは、ラベルテキストまたはアクセスキーの適用先に取り込まれる。	RT 式
<code>defaultValue</code>	<code>DisplayField</code> ビューピアの値が <code>null</code> の場合に使用する値。 ボタンタグの場合、この値は HTML 要素の <code>value</code> 属性になる値を表す。	RT 式
<code>disabled</code>	有効な値：「true」または「false」 ユーザー入力用のこの HTML コントロールを使用不可にする。 <ul style="list-style-type: none">使用不可のコントローラにはフォーカスは移らない。タブによる移動で、使用不可のコントロールは飛ばされる。 省略された場合は、「false」と見なされる。	RT 式

属性名	説明	注
elementId	<p>要素に名前を割り当てる。この名前は HTML ドキュメント内で一意である必要がある。これは HTML の id 属性に相応し、Web アプリケーションフレームワークには具体的な意味を持たないことに注意。</p> <p>id 属性には HTML でいくつかの役割がある。</p> <ul style="list-style-type: none"> • スタイルシートセレクタ • ハイパーテキストリンク用のターゲットアンカー • スクリプトから特定の要素を参照する手段 	RT 式
extraHtml	この HTML 要素の終了 (>) 直前に生成される任意の HTML	RT 式
name	<p>DisplayField ピアの名前。このピア、現在の親の ContainerView、TiledView、ViewBean いずれかの子である必要がある。DisplayField 名は、現在の親ビューを基準に解決される。</p> <p>ビューピアは、<JATO package>.view.CommandField 型である必要がある。</p> <p>この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照していると見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。</p> <p>例：</p> <pre> /header/orderList/customerName (ルートビューからの絶対位置) orderList/customerName (現在のコンテナからの相対位置) ../footer/orderList/customerName (親からの相対位置) </pre>	必須
src	このボタンをイメージボタンとして生成する場合のイメージの URL	RT 式
style	この HTML 要素に適用する CSS スタイル	RT 式
styleClass	この HTML 要素に適用する CSS スタイルシートクラス	RT 式
tabIndex	<p>現在のドキュメントに対するタブ移動における現在の要素の順番を指定する。この値は 0 から 32767 の範囲の数字である必要がある。</p> <p>このタグはまた、次の JavaScript イベントのサポートもしている。</p> <p>onBlur、onChange、onClick、onDbClick、onFocus、onKeyDown、onKeyPress、onKeyUp、onMouseDown、onMouseMove、onMouseOut、onMouseOver、onMouseUp</p>	

第11章

checkbox

<jato:checkbox>

HTML の `<input type="checkbox">` 要素を 1 つ生成します。

このタグは、HTML の `<form>` 要素や `useViewBean`、`containerView`、`tiledView` タグで囲まれている場合にのみ有効です。このタグに本体内容を含めることはできません。

例：

```
<jato:checkbox name="sendAdditionalInfo"/>
```

これは、次の HTML になります。

```
<input type="checkbox" name="PageFoo.sendAdditionalInfo" value="true">
```

属性名	説明	注
accessKey	生成される HTML 要素にアクセスキーを割り当てる。一般には、このアクセスキーは、ラベルテキストまたはアクセスキーの適用先に取り込まれる。	RT 式
disabled	有効な値: 「true」か「false」 ユーザー入力用のこの HTML コントロールを使用不可にする。 <ul style="list-style-type: none"> 使用不可のコントローラにはフォーカスは移らない。 タブによる移動で、使用不可のコントロールは飛ばされる。 省略された場合は、「false」と見なされる。	RT 式
elementId	要素に名前を割り当てる。この名前は HTML ドキュメント内で一意である必要がある。これは HTML の id 属性に相応し、Web アプリケーションフレームワークには具体的な意味を持たないことに注意。 id 属性には HTML でいくつかの役割がある。 <ul style="list-style-type: none"> スタイルシートセレクタ ハイパーテキストリンク用のターゲットアンカー スクリプトから特定の要素を参照する手段 	RT 式
extraHtml	この HTML 要素の終了 (>) 直前に生成される任意の HTML	RT 式
label	コントロールの右側に生成する、チェックボックスのラベル	RT 式
name	DisplayField ピアの名前。このピア、現在の親の ContainerView、TiledView、ViewBean いずれかの子である必要がある。DisplayField 名は、現在の親ビューを基準に解決される。 ビューピアは、<JATO package>.view.BooleanDisplayField 型である必要がある。 この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照しているから見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。 例: <ul style="list-style-type: none"> /header/orderList/customerName (ルートビューからの絶対位置) orderList/customerName (現在のコンテナからの相対位置) ../footer/orderList/customerName (親からの相対位置) 	必須
style	この HTML 要素に適用する CSS スタイル	RT 式

属性名	説明	注
styleClass	この HTML 要素に適用する CSS スタイルシートクラス	RT 式
tabIndex	<p>現在のドキュメントに対するタブ移動における現在の要素の順番を指定する。この値は 0 から 32767 の範囲の数字である必要がある。</p> <p>このタグはまた、次の JavaScript イベントのサポートもしている。 onBlur、onChange、onClick、onDblClick、onFocus、onKeyDown、onKeyPress、onKeyUp、onMouseDown、onMouseMove、onMouseOut、onMouseOver、onMouseUp、onSelect</p>	

第12章

combobox

<jato:combobox>

任意の個数の HTML `<option>` 要素からなる HTML `<input type="select">` 要素 1 つから構成される HTML コンボボックスを生成します。このタグには、コンボボックスを 1 つの要素として扱うことの便利さと、動的にリストコンテンツを生成するという柔軟性があります。

このタグは、HTML の `<form>` 要素や `useViewBean`、`containerView`、`tiledView` タグで囲まれている場合にのみ有効です。このタグに本体内容を含むことはできません。

例:

```
<jato:combobox name="itemCondition"/>
```

これは、次の HTML になります。

```
<select name="PageFoo.itemCondition">
  <option value="" selected>None Selected</option>
  <option value="Excellent">Excellent</option>
  <option value="Good">Good</option>
  <option value="Average">Average</option>
  <option value="Poor">Poor</option>
</select>
```

属性名	説明	注
accessKey	生成される HTML 要素にアクセスキーを割り当てる。一般には、このアクセスキーは、ラベルテキストまたはアクセスキーの適用先に取り込まれる。	RT 式
disabled	有効な値: 「true」か「false」 ユーザー入力用のこの HTML コントロールを使用不可にする。 <ul style="list-style-type: none"> 使用不可のコントローラにはフォーカスは移らない。 タブによる移動で、使用不可のコントロールは飛ばされる。 省略された場合は、「false」と見なされる。	RT 式
elementId	要素に名前を割り当てる。この名前は HTML ドキュメント内で一意である必要がある。これは HTML の id 属性に相応し、Web アプリケーションフレームワークには具体的な意味を持たないことに注意。 id 属性には HTML でいくつかの役割がある。 <ul style="list-style-type: none"> スタイルシートセレクタ ハイパーテキストリンク用のターゲットアンカー スクリプトから特定の要素を参照する手段 	RT 式
extraHtml	この HTML 要素の終了 (>) 直前に生成される任意の HTML	RT 式
name	DisplayField ピアの名前。このピア、現在の親の ContainerView、TiledView、ViewBean いずれかの子である必要がある。DisplayField 名は、現在の親ビューを基準に解決される。 ビューピアは、<JATO package>.view.html.SelectableGroup 型である必要がある。 この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照していると見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。 例: <pre> /header/orderList/customerName (ルートビューからの絶対位置) orderList/customerName (現在のコンテナからの相対位置) ../footer/orderList/customerName (親からの相対位置) </pre>	必須
style	この HTML 要素に適用する CSS スタイル	RT 式
styleClass	この HTML 要素に適用する CSS スタイルシートクラス	RT 式
tabIndex	現在のドキュメントに対するタブ移動における現在の要素の順番を指定する。この値は 0 から 32767 の範囲の数字である必要がある。 このタグはまた、次の JavaScript イベントのサポートもしている。 onBlur、onChange、onClick、onDblClick、onFocus、onKeyDown、onKeyPress、onKeyUp、onMouseDown、onMouseMove、onMouseOut、onMouseOver、onMouseUp、onSelect	RT 式

第13章

fileUpload

<jato:fileUpload>

HTML の `<input type="file">` 要素を生成します。
`com.iplanet.jato.MultipartFormServletFilter` クラスと組み合わせることによって、アプリケーション内で簡単にファイルアップロード機能を提供することができます。

`fileUpload` タグは、コンテンツの種類が `multipart/form-data` で宣言したフォーム内で使用する必要があります。

```
<jato:form name="form1" method="post" enctype="multipart/form-data">
  <jato:fileUpload name="fileUpload1"/>
</jato:form>
```

このタグに本体内容を含めることはできません。

例:

```
<jato:fileUpload name="fileUpload1" target="_top"/>
```

これは、次の HTML になります。

```
<input type="file" name="Page1.fileUpload1">
```

属性名	説明	注
accept	この入力タグを処理するサーバーが正しく処理するコンテンツの種類のコンマ区切りのリストを指定する。ユーザーエージェントはこの情報を使用して、サーバーに送信するファイルを選択するようユーザーに促す際に適合していないファイルを除外する。	
accessKey	生成される HTML 要素にアクセスキーを割り当てる。一般には、このアクセスキーは、ラベルテキストまたはアクセスキーの適用先に取り込まれる。	RT 式
disabled	有効な値: 「true」か「false」 ユーザー入力用のこの HTML コントロールを使用不可にする。 <ul style="list-style-type: none"> • 使用不可のコントローラにはフォーカスは移らない。 • タブによる移動で、使用不可のコントロールは飛ばされる。 省略された場合は、「false」と見なされる。	RT 式
elementId	要素に名前を割り当てる。この名前は HTML ドキュメント内で一意である必要がある。これは HTML の id 属性に相応し、Web アプリケーションフレームワークには具体的な意味を持たないことに注意。 id 属性には HTML でいくつかの役割がある。 <ul style="list-style-type: none"> • スタイルシートセレクタ • ハイパーテキストリンク用のターゲットアンカー • スクリプトから特定の要素を参照する手段 	RT 式
extraHtml	この HTML 要素の終了 (>) 直前に生成される任意の HTML	RT 式
maxLength	有効な値: 正数 ユーザーがフィールドに入力できる最大文字数を指定する。 省略された場合は、フィールドに入力できる文字数の制限はない。	RT 式
name	FileUpload ビアの名前。このピア、現在の親の ContainerView、TiledView、ViewBean いずれかの子である必要がある。名前は、現在の親ビューを基準に解決される。 ビューピアは、<JATO package>.view.html2.FileUpload 型である必要がある。 この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照しているから見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。 例: <ul style="list-style-type: none"> /header/orderList/customerName (ルートビューからの絶対位置) orderList/customerName (現在のコンテナからの相対位置) ../footer/orderList/customerName (親からの相対位置) 	必須
size	ファイル入力テキストフィールドのサイズ	RT 式

属性名	説明	注
style	この HTML 要素に適用する CSS スタイル	RT 式
styleClass	この HTML 要素に適用する CSS スタイルシートクラス	RT 式
tabIndex	<p>現在のドキュメントに対するタブ移動における現在の要素の順番を指定する。この値は 0 から 32767 の範囲の数字である必要がある。</p> <p>このタグはまた、次の JavaScript イベントのサポートもしている。 onBlur、onChange、onClick、ondblclick、onfocus、onkeydown、onkeypress、onkeyup、onmousedown、onmousemove、onmouseout、onmouseover、onmouseup、onselect</p>	

第14章

form

<jato:form>

HTML の <form> 要素を生成します。フォームの内容は、このタグの本体内容で記述します。HTML フォーム属性のアクションは、タグハンドラによって Web アプリケーションフレームワーク実行時環境に応じたものが動的に作成されるため、このタグでは表しません。

`form` タグにはビューピアがなく、JATO あるいは HTML の別の <form> 要素内に入れ子にできないことに注意してください。ただし、最終的には、このタグは `useViewBean` タグで囲む必要があります。このタグに本体内容を含むことはできません。

例：

```
<jato:form name="form1" target="_top"/>
```

これは、次の HTML になります。

```
<form name="form1" method="post" action="...(JATO URL)...">  
...  
</form>
```

プロパティ名	説明	注
accept	このフォームを処理するサーバーが正しく処理するコンテンツの種類のコマ区切りのリストを指定する。ユーザーエージェントはこの情報を使用して、サーバーに送信するファイルを選択するようユーザーに促す際に適合していないファイルを除外する。	
acceptCharset	このフォームを処理するサーバーが受け付ける入力データに対する文字エンコーディングのリストを指定する。値は、文字セット値を空白文字かコンマのいずれかで区切ったリスト。クライアントは、このリストを排他的論理和のリストとして、すなわち、受け取るエンティティごとに任意の1文字のエンコーディングをサーバーが受け付けられると解釈する必要がある。 この属性のデフォルト値は予約文字列の「UNKNOWN」。ユーザーエージェントは、この値を、form 要素を含むドキュメントの送信に使用された文字エンコーディングと解釈できる。	
defaultCommandChild	デフォルトの要求中に起動する子の <code>CommandField</code> の名前。デフォルトの要求とは、起動する子コマンドの指定を含まない要求である。一般に、こうした状況は、ユーザーがフォームフィールド内にいるときに Enter キーを押し、そのために、特定のボタンを選択することなく取り囲んでいるフォームが送信された場合に発生する。 ビューピアは、 <code><JATO package>.view.CommandField</code> 型である必要がある。 この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、 <code>ContainerView</code> か <code>ContainerView</code> からの派生物 (<code>TiledView</code> など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (<code>ViewBean</code>) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照していると見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。 例： <pre>defaultCommandChild="/okButton" (ルートビューからの絶対位置) defaultCommandChild="okButton" (現在のコンテナからの相対位置) defaultCommandChild="../okButton" (親からの相対位置)</pre>	RT 式
elementId	要素に名前を割り当てる。この名前は HTML ドキュメント内で一意である必要がある。これは HTML の <code>id</code> 属性に相応し、Web アプリケーションフレームワークには具体的な意味を持たないことに注意。 <code>id</code> 属性には HTML でいくつかの役割がある。 <ul style="list-style-type: none"> • スタイルシートセレクタ • ハイパーテキストリンク用のターゲットアンカー • スクリプトから特定の要素を参照する手段 	RT 式

プロパティ名	説明	注
encType	サーバーにフォームを送信するのに使用するコンテンツの種類を指定する (method の値が「post」の場合)。この属性のデフォルト値は「application/x-www-form-urlencoded」。この値の「multipart/form-data」は INPUT 要素の type="file" と組み合わせて使用することを推奨する。	RT 式
method	有効な値: 「post」か「get」 この要求の送信に使用する HTTP メソッド。省略された場合は「post」と見なされる。	
name	取り囲む HTML ドキュメント内のフォーム要素の名前。この名前は任意で、ピアインスタンスと無関係。	
style	この HTML 要素に適用する CSS スタイル	RT 式
styleClass	この HTML 要素に適用する CSS スタイルシートクラス	RT 式
target	このフォームの送信先のウィンドウターゲット このタグはまた、次の JavaScript イベントのサポートもしている。 onReset、onSubmit	RT 式

第15章

frameSrc

<jato:frameSrc>

HTML の <frame> タグ用の src および name 属性を生成します。

このタグは、HTML の <form> 要素や `useViewBean`、`containerView`、`tiledView` タグで囲まれている場合にのみ有効です。このタグに本体内容を含めることはできません。

注：HTML の <frame> タグが存在しない場合、HTML エディタで WYSIWYG 形式でフレームセットを編集することはできません。このため、このタグは HTML の <frame> タグの完全な置き換えを図るものではありません。このタグは、動的な src 値を持つ HTML の <frame> タグを提供することだけを目的にしています。

例：

```
<frameset rows="30%,70%">
  <frame <jato:frameSrc name="Frame1" location="internal" />
  >
  <frame <jato:frameSrc name="Frame2" location="external" />
  >
</frameset>
```

これは、次の HTML になります。

```
<frameset rows="30%,70%">
  <frame src="...(JATO page)..." name="Frame1" >
  <frame src="/blank.html" name="Frame2" >
</frameset>
```

属性名	説明	注
defaultValue	DisplayField ビューピアの値が null の場合に使用する値。 frameSrc タグで、location 属性が「internal」の場合、この値は JATO ページ名を表す。location 属性が「external」の場合は、JATO 以外の任意の URL を表す。	RT 式
name	DisplayField ピアの名前。このピア、現在の親の ContainerView、TiledView、ViewBean いずれかの子である必要がある。DisplayField 名は、現在の親ビューを基準に解決される。 ビューピアは、<JATO package>.view.DisplayField 型である必要がある。 この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照していると思なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。 例： /header/orderList/customerName (ルートビューからの絶対位置) orderList/customerName (現在のコンテナからの相対位置) ../footer/orderList/customerName (親からの相対位置)	必須
location	有効な値：「internal」か「external」 値「internal」は、このフレーム用のソースドキュメントが現在のアプリケーション内の JATO ページであることを示す。 値「external」は、このフレーム用のソースドキュメントが現在のアプリケーションの外部にある任意の URL であることを示す。 省略された場合は、「internal」と見なされる。 このタグは、JavaScript イベントをサポートしない。 JavaScript は、このタグを含む HTML の <frame> 要素内に直接追加できる。	

第 16 章

hidden

<jato:hidden>

HTML の `<input type="hidden">` 要素を生成します。

このタグは、HTML の `<form>` 要素や `useViewBean`、`containerView`、`tiledView` タグで囲まれている場合にのみ有効です。このタグに本体内容を含めることはできません。

例：

```
<jato:hidden name="nameFilter" defaultValue="*" />
```

これは、次の HTML になります。

```
<input type="hidden" name="PageFoo.nameFilter" value="Jo*">
```

属性名	説明	注
defaultValue	<p>DisplayField ビューピアの値が null の場合に使用する値。</p> <p>hidden タグの場合、この値は HTML の Hidden 要素の value 属性になる値を表す。</p>	RT 式
name	<p>DisplayField ピアの名前。このピア、現在の親の ContainerView、TiledView、ViewBean いずれかの子である必要がある。DisplayField 名は、現在の親ビューを基準に解決される。</p> <p>ビューピアは、<JATO package>.view.DisplayField の任意のサブタイプである必要がある。</p> <p>この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照していると見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。</p> <p>例：</p> <ul style="list-style-type: none"> /header/orderList/customerName (ルートビューからの絶対位置) orderList/customerName (現在のコンテナからの相対位置) ../footer/orderList/customerName (親からの相対位置) <p>このタグは、JavaScript イベントをサポートしない。</p>	必須

第17章

href

<jato:href>

HTML の `...` 要素を生成します。

このタグは、`useViewBean` か `containerView`、または `tiledView` タグで囲まれている場合にのみ有効です。このタグに本体内容を含めることはできません。この内容は、生成される要素の `<a>` と `` タグの間に含まれ、目に見えるクリック可能な HREF 要素として表示されます。

例：

```
<jato:href name="orderDrillDown">See orders</jato:href>
```

これは、次の HTML になります。

```
<a href="...(JATO URL)...?PageFoo.orderDrillDown=10345">See orders</a>
```

属性名	説明	注
<code>accessKey</code>	生成される HTML 要素にアクセスキーを割り当てる。一般には、このアクセスキーは、ラベルテキストまたはアクセスキーの適用先に取り込まれる。	RT 式
<code>anchor</code>	生成される HREF に付加するアンカー値。	RT 式
<code>disabled</code>	有効な値：「true」か「false」 ユーザー入力用のこの HTML コントロールを使用不可にする。 <ul style="list-style-type: none">使用不可のコントロールにはフォーカスは移らない。タブによる移動で、使用不可のコントロールは飛ばされる。 省略された場合は、「false」と見なされる。	RT 式

属性名	説明	注
defaultValue	<p>DisplayField ビューピアの値が null の場合に使用する値。</p> <p>href タグの場合、この値は、HREF の照会文字列の一部として自動的に含まれる特殊な名前と値のペアを表す。暗黙の名前と値ペアの書式は以下のとおり。</p> <p>...?<限定表示フィールド名>=<値>&...</p> <p>この仕組みによって href は表示文字列ばかりでなく、アプリケーションに重要な値を持つことができる。</p>	RT 式
elementId	<p>要素に名前を割り当てる。この名前は HTML ドキュメント内で一意である必要がある。これは HTML の id 属性に相応し、Web アプリケーションフレームワークには具体的な意味を持たないことに注意。</p> <p>id 属性には HTML でいくつかの役割がある。</p> <ul style="list-style-type: none"> • スタイルシートセレクタ • ハイパーテキストリンク用のターゲットアンカー • スクリプトから特定の要素を参照する手段 	RT 式
extraHtml	この HTML 要素の終了 (>) 直前に生成される任意の HTML	RT 式
name	<p>DisplayField ピアの名前。このピア、現在の親の ContainerView、TiledView、ViewBean いずれかの子である必要がある。DisplayField 名は、現在の親ビューを基準に解決される。</p> <p>ビューピアは、<JATO package>.view.CommandField 型である必要がある。</p> <p>この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照しているから見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。</p> <p>例：</p> <p>/header/orderList/customerName (ルートビューからの絶対位置)</p> <p>orderList/customerName (現在のコンテナからの相対位置)</p> <p>../footer/orderList/customerName (親からの相対位置)</p>	必須
queryParams	<p>HREF の照会文字列に付与する任意の個数の名前と値のペアを指定する。</p> <p>名前と値のペアは「名前 = 値」の書式に従い、値は必要に応じて JSP 作成者がエンコードした URL。追加の値と名前のペアはそれぞれ、「&」文字で区切る必要がある。</p> <p>例：</p> <p>queryParams="fname=Mike&lname=Jones"</p>	RT 式
style	この HTML 要素に適用する CSS スタイル	RT 式
styleClass	この HTML 要素に適用する CSS スタイルシートクラス	RT 式
tabIndex	現在のドキュメントに対するタブ移動における現在の要素の順番を指定する。この値は 0 から 32767 の範囲の数字である必要がある。	
target	フレームターゲット	RT 式

属性名	説明	注
title	この要素に関するヒント的な情報を提供する。	RT 式
trim	<p>有効な値：「true」か「false」</p> <p>値「true」は、タグの内容から先行または後続するすべての空白を削除すべきことを示す。値「false」はそれらの空白を残す。</p> <p>省略された場合は、「true」と見なされる。</p> <p>このタグはまた、次の JavaScript イベントのサポートもしている。</p> <p>onBlur、onChange、onClick、ondblclick、onfocus、onkeydown、onkeypress、onkeyup、onmousedown、onmousemove、onmouseout、onmouseover、onmouseup、onselect</p>	

第18章

image

<jato:image>

HTML の 要素を生成します。

このタグは、HTML の <form> 要素や `useViewBean`、`containerView`、`tiledView` タグで囲まれている場合にのみ有効です。このタグに本体内容を含めることはできません。

例：

```
<jato:image name="employeePhoto" defaultValue=
"/images/nophoto.jpg"/>
```

これは、次の HTML になります。

```

```

属性名	説明	注
align	<p>コンテキストを基準にした IMG の位置を指定する。</p> <p>取り囲むテキストを基準にしたオブジェクトの位置は、次の値によって決定される。</p> <ul style="list-style-type: none"> • bottom: 現在の基線とオブジェクトの最下部を合わせる (デフォルト値)。 • middle: 現在の基線とオブジェクトの中心を合わせる。 • top: 現在のテキスト行の最上部とオブジェクトの最上部を合わせる。 • left: イメージを現在の左マージンに寄せる。 • right: イメージを現在の右マージンに寄せる。 	RT 式
alt	イメージを表示できないユーザーエージェントの場合の代替テキストを指定する。	RT 式
border	ピクセル数で IMG 枠の幅を指定する。デフォルト値は、ユーザーエージェントに依存する。	RT 式
defaultValue	DisplayField ビューピアの値が null の場合に使用する値。 image タグの場合、この値は HTML の img 要素の src 属性になる値を表す。	RT 式
elementId	<p>要素に名前を割り当てる。この名前は HTML ドキュメント内で一意である必要がある。これは HTML の id 属性に相応し、Web アプリケーションフレームワークには具体的な意味を持たないことに注意。</p> <p>id 属性には HTML でいくつかの役割がある。</p> <ul style="list-style-type: none"> • スタイルシートセレクタ • ハイパーテキストリンク用のターゲットアンカー • スクリプトから特定の要素を参照する手段 	RT 式
extraHtml	この HTML 要素の終了 (>) 直前に生成される任意の HTML	RT 式
height	イメージの高さの優先指定。指定された場合は、イメージまたはオブジェクトのそのままのサイズではなく、width および height 属性の値を優先するようというユーザーエージェントへの指示になる。	RT 式
hspace	IMG の左右に挿入する空白量を指定する。デフォルト値はないが、一般には小さなゼロ以外の長さ。	RT 式
isMap	<p>有効な値: 「true」か「false」</p> <p>IMG がサーバー側のイメージマップであることを示す。</p> <p>ユーザーがイメージをクリックすることによってリンクを起動すると、ドキュメントが存在するサーバーに直接、画面座標が送信される。画面座標はイメージを基準にした画面ピクセルで表される。</p> <p>省略された場合は、「false」と見なされる。</p>	RT 式
longDesc	イメージの長い説明へのリンクを指定する。この説明は、alt 属性を使って指定した短い説明を補完するものである。	RT 式

属性名	説明	注
name	<p>DisplayField ビアの名前。このピア、現在の親の ContainerView、TiledView、ViewBean いずれかの子である必要がある。DisplayField 名は、現在の親ビューを基準に解決される。</p> <p>ビューピアは、<JATO package>.view.DisplayField の任意のサブタイプである必要がある。</p> <p>この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照していると見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。</p> <p>例：</p> <pre> /header/orderList/customerName (ルートビューからの絶対位置) orderList/customerName (現在のコンテナからの相対位置) ../footer/orderList/customerName (親からの相対位置) </pre>	必須
normalize	<p>有効な値: 「true」か「false」</p> <p>src の URL を自動的に正規化するように指示する。このことは、絶対 URL 値の前に現在のサブレットコンテキストが付加されることを意味する。相対 URL 値の前に現在のサブレットコンテキストが付加されることはない。相対および絶対ともに src の URL 値は URL エンコーディングされる。</p> <p>省略された場合は、「false」と見なされる。</p>	RT 式
style	この HTML 要素に適用する CSS スタイル	RT 式
styleClass	この HTML 要素に適用する CSS スタイルシートクラス	RT 式
title	この要素に関するヒント的な情報を提供する。	RT 式
useMap	イメージマップを要素に関連付ける。イメージマップは MAP 要素によって定義される。usemap 値は、関連付けられる MAP 要素の name 属性値と一致している必要がある。	RT 式
vspace	IMG の上下に挿入する空白量を指定する。デフォルト値はないが、一般には小さなゼロ以外の長さ。	RT 式
width	<p>イメージの幅の優先指定。指定された場合は、イメージまたはオブジェクトのそのままのサイズではなく、width および height 属性の値を優先するようというユーザーエージェントへの指示になる。</p> <p>このタグはまた、次の JavaScript イベントのサポートもしている。</p> <p>onBlur、onChange、onClick、onDbClick、onFocus、onKeyDown、onKeyPress、onKeyUp、onMouseDown、onMouseMove、onMouseOut、onMouseOver、onMouseUp、onSelect</p>	RT 式

第19章

listbox

<jato:listbox>

任意の個数の HTML <option> 要素からなる HTML <input type="select"> 要素 1 つから構成される HTML リストボックスを生成します。このタグには、リストボックスを 1 つの要素として扱うことの便利さと、動的にリストコンテンツを生成するという柔軟性があります。

このタグは、HTML の <form> 要素や `useViewBean`、`containerView`、`tiledView` タグで囲まれている場合にのみ有効です。このタグに本体内容を含めることはできません。

例：

```
<jato:listbox name="itemSize" size="4" multiple="false"/>
```

これは、次の HTML になります。

```
<select name="PageFoo.itemSize" size="4">  
  <option value="S">Small</option>  
  <option value="M">Medium</option>  
  <option value="L">Large</option>  
  <option value="XL">Extra Large</option>  
</select>
```

属性名	説明	注
accessKey	生成される HTML 要素にアクセスキーを割り当てる。一般には、このアクセスキーは、ラベルテキストまたはアクセスキーの適用先に取り込まれる。	RT 式
disabled	有効な値: 「true」か「false」 ユーザー入力用のこの HTML コントロールを使用不可にする。 <ul style="list-style-type: none"> 使用不可のコントローラにはフォーカスは移らない。 タブによる移動で、使用不可のコントロールは飛ばされる。 省略された場合は、「false」と見なされる。	RT 式
elementId	要素に名前を割り当てる。この名前は HTML ドキュメント内で一意である必要がある。これは HTML の id 属性に相応し、Web アプリケーションフレームワークには具体的な意味を持たないことに注意。 id 属性には HTML でいくつかの役割がある。 <ul style="list-style-type: none"> スタイルシートセレクタ ハイパーテキストリンク用のターゲットアンカー スクリプトから特定の要素を参照する手段 	RT 式
extraHtml	この HTML 要素の終了 (>) 直前に生成される任意の HTML	RT 式
multiple	有効な値: 「true」か「false」 値「true」は、ユーザーが複数の項目を選択できることを示す。 値「false」は、ユーザーが1つの項目だけ選択できることを示す。 省略された場合は、「false」と見なされる。	
name	DisplayField ピアの名前。このピア、現在の親の ContainerView、TiledView、ViewBean いずれかの子である必要がある。DisplayField 名は、現在の親ビューを基準に解決される。 ビューピアは、<JATO package>.view.html.SelectableGroup 型である必要がある。 この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照していると見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。 例: <ul style="list-style-type: none"> /header/orderList/customerName (ルートビューからの絶対位置) orderList/customerName (現在のコンテナからの相対位置) ../footer/orderList/customerName (親からの相対位置) 	必須
size	有効な値: 正数 リスト内に一度に表示する行数を指定する。 省略された場合は、生成される HTML から size 属性が省略される。	RT 式

属性名	説明	注
style	この HTML 要素に適用する CSS スタイル	RT 式
styleClass	この HTML 要素に適用する CSS スタイルシートクラス	RT 式
tabIndex	<p>現在のドキュメントに対するタブ移動における現在の要素の順番を指定する。この値は 0 から 32767 の範囲の数字である必要がある。</p> <p>このタグはまた、次の JavaScript イベントのサポートもしている。 onBlur、onChange、onClick、onDblClick、onFocus、onKeyDown、 onKeyPress、onKeyUp、onMouseDown、onMouseMove、onMouseOut、 onMouseOver、onMouseUp、onSelect</p>	

第20章

password

<jato:password>

HTML の `<input type="password">` 要素を生成します。

このタグは、HTML の `<form>` 要素や `useViewBean`、`containerView`、`tiledView` タグで囲まれている場合にのみ有効です。このタグに本体内容を含めることはできません。

例：

```
<jato:password name="newPassword" size="8" maxLength="16"/>
```

これは、次の HTML になります。

```
<input type="password" name="PageFoo.newPassword" value=""
maxLength="16" size="8">
```

属性名	説明	注
<code>accessKey</code>	生成される HTML 要素にアクセスキーを割り当てる。一般には、このアクセスキーは、ラベルテキストまたはアクセスキーの適用先に取り込まれる。	RT 式
<code>defaultValue</code>	<code>DisplayField</code> ビューピアの値が <code>null</code> の場合に使用する値。 <code>hidden</code> タグの場合、この値は HTML の <code>password</code> 要素の <code>value</code> 属性になる値を表す。	RT 式
<code>disabled</code>	有効な値：「true」か「false」 ユーザー入力用のこの HTML コントロールを使用不可にする。 <ul style="list-style-type: none">使用不可のコントローラにはフォーカスは移らない。タブによる移動で、使用不可のコントロールは飛ばされる。 省略された場合は、「false」と見なされる。	RT 式

属性名	説明	注
elementId	<p>要素に名前を割り当てる。この名前は HTML ドキュメント内で一意である必要がある。これは HTML の id 属性に相応し、Web アプリケーションフレームワークには具体的な意味を持たないことに注意。</p> <p>id 属性には HTML でいくつかの役割がある。</p> <ul style="list-style-type: none"> • スタイルシートセレクタ • ハイパーテキストリンク用のターゲットアンカー • スクリプトから特定の要素を参照する手段 	RT 式
extraHtml	この HTML 要素の終了 (>) 直前に生成される任意の HTML	RT 式
maxLength	<p>有効な値：正数</p> <p>ユーザーがフィールドに入力できる最大文字数を指定する。</p> <p>省略された場合は、フィールドに入力できる文字数の制限はない。</p>	RT 式
name	<p>DisplayField ビアの名前。このピア、現在の親の ContainerView、TiledView、ViewBean いずれかの子である必要がある。DisplayField 名は、現在の親ビューを基準に解決される。</p> <p>ビューピアは、<JATO package>.view.DisplayField 型である必要がある。この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照していると見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。</p> <p>例：</p> <pre> /header/orderList/customerName (ルートビューからの絶対位置) orderList/customerName (現在のコンテナからの相対位置) ../footer/orderList/customerName (親からの相対位置) </pre>	必須
readOnly	<p>有効な値：「true」か「false」</p> <p>値「true」は、この要素に含まれるテキストをユーザーが変更できないことを示す。</p> <p>省略された場合は、「false」と見なされる。</p>	RT 式
size	<p>有効な値：正数</p> <p>文字数でこのコントロールの初期幅を指定する。</p> <p>省略された場合は、生成される HTML から size 属性が省略される。</p>	RT 式
style	この HTML 要素に適用する CSS スタイル	RT 式

属性名	説明	注
styleClass	この HTML 要素に適用する CSS スタイルシートクラス	RT 式
tabIndex	<p>現在のドキュメントに対するタブ移動における現在の要素の順番を指定する。この値は 0 から 32767 の範囲の数字である必要がある。</p> <p>このタグはまた、次の JavaScript イベントのサポートもしている。 onBlur、onChange、onClick、onDblClick、onFocus、onKeyDown、 onKeyPress、onKeyUp、onMouseDown、onMouseMove、onMouseOut、 onMouseOver、onMouseUp、onSelect</p>	

第21章

radioButtons

<jato:radioButtons>

同じ name 属性値を持つ、任意の個数の HTML `<input type="radio">` 要素で構成される HTML リソースバンドルを生成します。このタグには、ラジオボタンを1つの要素として扱うことの便利さと、動的に選択肢を生成するという柔軟性があります。

このタグは、HTML の `<form>` 要素や `useViewBean`、`containerView`、`tiledView` タグで囲まれている場合にのみ有効です。このタグに本体内容を含めることはできません。

例:

```
<jato:radioButtons name="incomeRange" layout="vertical"/>
```

これは、次の HTML になります。

```
<input type="radio" name="PageFoo.incomeRange" value=""
checked>Nondisclosed
<br>
<input type="radio" name="PageFoo.incomeRange" value="19">$0
to $19,999
<Br>
<input type="radio" name="PageFoo.incomeRange" value=
"49">$20,000-$49,999
<Br>
<input type="radio" name="PageFoo.incomeRange" value=
"50">$50,000 or more
<Br>
```

属性名	説明	注
accessKey	生成される HTML 要素にアクセスキーを割り当てる。一般には、このアクセスキーは、ラベルテキストまたはアクセスキーの適用先に取り込まれる。	RT 式
disabled	有効な値: 「true」か「false」 ユーザー入力用のこの HTML コントロールを使用不可にする。 <ul style="list-style-type: none"> 使用不可のコントローラにはフォーカスは移らない。 タブによる移動で、使用不可のコントロールは飛ばされる。 省略された場合は、「false」と見なされる。	RT 式
elementId	要素に名前を割り当てる。この名前は HTML ドキュメント内で一意である必要がある。これは HTML の id 属性に相応し、Web アプリケーションフレームワークには具体的な意味を持たないことに注意。 id 属性には HTML でいくつかの役割がある。 <ul style="list-style-type: none"> スタイルシートセレクタ ハイパーテキストリンク用のターゲットアンカー スクリプトから特定の要素を参照する手段 	RT 式
extraHtml	この HTML 要素の終了 (>) 直前に生成される任意の HTML	RT 式
layout	有効な値: 「h」か「horizontal」または「h」から始まる任意の文字列 「h」から始まる値は、ラジオボタンをこのグループ内に横に並べることを指示する。 値がないか、「h」以外の文字で始まる値は、縦に並べることを指示する。この縦の並びは、生成する各 HTML ラジオボタンコントロールの後に 要素を付加することによって実現する。	RT 式
name	DisplayField ピアの名前。このピア、現在の親の ContainerView、TiledView、ViewBean いずれかの子である必要がある。DisplayField 名は、現在の親ビューを基準に解決される。 ビューピアは、<JATO package>.view.html.SelectableGroup 型である必要がある。 この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照していると見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。 例: <pre> /header/orderList/customerName (ルートビューからの絶対位置) orderList/customerName (現在のコンテナからの相対位置) ../footer/orderList/customerName (親からの相対位置) </pre>	必須
style	この HTML 要素に適用する CSS スタイル	RT 式

属性名	説明	注
styleClass	この HTML 要素に適用する CSS スタイルシートクラス	RT 式
tabIndex	<p>現在のドキュメントに対するタブ移動における現在の要素の順番を指定する。この値は 0 から 32767 の範囲の数字である必要がある。</p> <p>このタグはまた、次の JavaScript イベントのサポートもしている。</p> <p>onBlur、onChange、onClick、onDblClick、onFocus、onKeyDown、onKeyPress、onKeyUp、onMouseDown、onMouseMove、onMouseOut、onMouseOver、onMouseUp、onSelect</p>	

第22章

text

<jato:text>

表示フィールドの値をプレーンの HTML 本体テキストとして生成します。このタグは、`useViewBean` か `containerView`、または `tiledView` タグの本体内に入れ子になっている場合にのみ有効です。

このタグは、取り囲む `<form>` 要素を必要としませんが、`useViewBean` か `containerView`、または `tiledView` タグで囲む必要があります。このタグに本体内容を含めることはできません。

例：

```
<jato:text name="ssn" formatType="string" formatMask="??-??-????" />
```

これは、次の HTML になります。

```
123-45-6789
```

属性名	説明	注
defaultValue	DisplayField ビューピアの値が null の場合に使用する値。	RT 式
escape	有効な値: 「true」か「false」 値「true」は、値が HMML でエスケープすることを指示する。 省略された場合は、値「true」と見なされる。	
formatMask	formatMask 属性の有効な値は、formatType 属性の値によって異なる。 formatType の値が「string」または「alpha」の場合: フォーマットマスクは、静的テキストと予約メタキャラクタの「?」の任意の組み合わせから構成される。1 文字のメタキャラクタ「?」が、値を左から順に書式化する際に単一文字のプレースホルダとしての働きを持つ。リテラルの「?」は、「?」を「\\」でエスケープすることによって指定できる。 formatType の値が「dec」または「curr」の場合: フォーマットマスクは、セミコロンで区切られた 2 つのパターン、主パターン (精度と桁区切り文字) と負数パターンで構成される (パターンの詳細は、 <code>java.text.DecimalFormat</code> を参照)。 (* この表のあとの表を参照)	

属性名	説明	注
formatType	<p>有効な値: 「string」(「alpha」でも可)「dec」「curr」「date」のどれか 各値は特殊なマスク規則を持つ (formatMask 属性を参照)。 値「string」は、単純なテキストマスクを適用することによる任意のテキストの書式化をサポートする。これで、自由に動的コンテンツと静的テキストを組み合わせることができる。値「alpha」は「string」と同じ意味。 値「dec」は、数値マスクを適用することによる数値の書式化をサポートする。 値「curr」は、通貨マスクを適用することによる通貨値の書式化をサポートする。通貨の書式は小数の指定を受け取り、ロケールに固有の通貨書式を適用する。現在、未解決のバグのため、通貨の書式化は機能していない。 値「date」は有効ではあるが、現状、日付の書式化をサポートする実装はない。これは、java.text.DateFormat が、taglib 実装の必要とする一般的な書式化サポートにするのに適していないためである。この制約の回避策を調査中である。 このため、日付を書式化する場合は、 begin<fieldName>Display(ChildDisplayEvent event) か end<fieldName>Display(ChildContentDisplayEvent event)、またはその両方を適用し、そこで java.text.DateFormat クラスに用意されているデータ書式化サポートを明示的に利用することを推奨。 Web アプリケーションフレームワーク実装のテキストの書式化は、java.text パッケージとそこに含まれている書式関係のクラスに大きく依拠している。詳細は、そのパッケージを参照。また、<JATO package>.util.HtmlUtil クラスも参照のこと。</p>	
name	<p>DisplayField ピアの名前。このピア、現在の親の ContainerView、TiledView、ViewBean いずれかの子である必要がある。DisplayField 名は、現在の親ビューを基準に解決される。 ビューピアは、<JATO package>.view.DisplayField 型である必要がある。 この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照していると思なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。 例： /header/orderList/customerName (ルートビューからの絶対位置) orderList/customerName (現在のコンテナからの相対位置) ../footer/orderList/customerName (親からの相対位置)</p>	必須
このタグは、JavaScript イベントをサポートしない。		

(* 前の表の formatMask 属性を参照)

書式の種類	書式のマスク	値の例	生成される値
string	??-??-????	123456789	123-45-6789
string	??HELLO??	abcd	abHELLOcd
string	??HELLO\ \???	abcd	abHELLO?cd
dec	###0; (-#)	1000000.99	1000001
dec	#,##0.00; (-#)	1000000.99	1,000,000.99
dec	###0.00; (-#)	1000000.99	1000000.99
dec	###0; (-#)	-1000000.99	(-1000001)
dec	###0; (#)	-1000000.99	(1000001)
dec	#,##0.00; -#	-1000000.99	-1,000,000.99

第23章

textArea

<jato:textArea>

HTML の <textarea> 要素を生成します。

このタグは、HTML の <form> 要素や `useViewBean`、`containerView`、`tiledView` タグで囲まれている場合にのみ有効です。このタグに本体内容を含めることはできません。

例：

```
<jato:textArea name="comments" cols="80" rows="5" wrap="virtual"/>
```

これは、次の HTML になります。

```
<textarea name="PageFoo.comments" cols="80" rows="5" wrap="virtual">
...
</textarea>
```

属性名	説明	注
accessKey	生成される HTML 要素にアクセスキーを割り当てる。一般には、このアクセスキーは、ラベルテキストまたはアクセスキーの適用先に取り込まれる。	RT 式
cols	有効な値：正数 平均的な文字幅でフィールドの表示幅を指定する。	必須、 RT 式
defaultValue	DisplayField ビューピアの値が null の場合に使用する値。 textArea タグの場合、この値は、textarea 要素の開始タグと終了タグの間に生成されるテキストを表す。	RT 式
disabled	有効な値：「true」か「false」 ユーザー入力用のこの HTML コントロールを使用不可にする。 <ul style="list-style-type: none"> 使用不可のコントローラにはフォーカスは移らない。 タブによる移動で、使用不可のコントロールは飛ばされる。 省略された場合は、「false」と見なされる。	RT 式
elementId	要素に名前を割り当てる。この名前は HTML ドキュメント内で一意である必要がある。これは HTML の id 属性に相応し、Web アプリケーションフレームワークには具体的な意味を持たないことに注意。 id 属性には HTML でいくつかの役割がある。 <ul style="list-style-type: none"> スタイルシートセレクタ ハイパーテキストリンク用のターゲットアンカー スクリプトから特定の要素を参照する手段 	RT 式
extraHtml	<textarea> 要素の終了 (>) 直前に生成される任意の HTML	RT 式
formatMask	formatMask 属性の有効な値は、formatType 属性の値によって異なる。 formatType の値が「string」または「alpha」の場合： フォーマットマスクは、静的テキストと予約メタキャラクタの「?」の任意の組み合わせから構成される。1 文字のメタキャラクタ「?」が、値を左から順に書式化する際に単一文字のプレースホルダとしての働きを持つ。リテラルの「?」は、「?」を「\」でエスケープすることによって指定できる。 formatType の値が「dec」または「curr」の場合： フォーマットマスクは、セミコロンで区切られた 2 つのパターン、主パターン (精度と桁区切り文字) と負数パターンで構成される (パターンの詳細は、java.text.DecimalFormat を参照)。 (* この表のあとの表を参照)	

属性名	説明	注
formatType	<p>有効な値: 「string」 (「alpha」でも可) 「dec」 「curr」 「date」 のどれか 各値は特殊なマスク規則を持つ (formatMask 属性を参照)。 値「string」は、単純なテキストマスクを適用することによる任意のテキストの書式化をサポートする。これで、自由に動的コンテンツと静的テキストを組み合わせることができる。値「alpha」は「string」と同じ意味。 値「dec」は、数値マスクを適用することによる数値の書式化をサポートする。 値「curr」は、通貨マスクを適用することによる通貨値の書式化をサポートする。通貨の書式は小数の指定を受け取り、ロケールに固有の通貨書式を適用する。現在、未解決のバグのため、通貨の書式化は機能していない。 値「date」は有効ではあるが、現状、日付の書式化をサポートする実装はない。これは、java.text.DateFormat が、taglib 実装の必要とする一般的な書式化サポートに適用するのに適していないためである。この制約の回避策を調査中である。 このため、日付を書式化する場合は、 begin<fieldName>Display (ChildDisplayEvent event) か end<fieldName>Display (ChildContentDisplayEvent event)、またはその両方を適用し、そこで java.text.DateFormat クラスに用意されているデータ書式化サポートを明示的に利用することを推奨。 Web アプリケーションフレームワーク実装のテキストの書式化は、java.text パッケージとそこに含まれている書式関係のクラスに大きく依拠している。詳細は、そのパッケージを参照。また、<JATO package>.util.HtmlUtil クラスも参照のこと。</p>	
name	<p>DisplayField ピアの名前。このピア、現在の親の ContainerView、TiledView、ViewBean いずれかの子である必要がある。DisplayField 名は、現在の親ビューを基準に解決される。 ビューピアは、<JATO package>.view.DisplayField 型である必要がある。この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照していると見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。 例： /header/orderList/customerName (ルートビューからの絶対位置) orderList/customerName (現在のコンテナからの相対位置) ../footer/orderList/customerName (親からの相対位置)</p>	必須
readOnly	<p>有効な値: 「true」か「false」 値「true」は、この要素に含まれるテキストをユーザーが変更できないことを示す。 省略された場合は、「false」と見なされる。</p>	RT 式
rows	<p>有効な値: 正数 生成するコントロールに含める表示テキスト行数を指定する。</p>	必須、 RT 式

属性名	説明	注
style	この HTML 要素に適用する CSS スタイル	RT 式
styleClass	この HTML 要素に適用する CSS スタイルシートクラス	RT 式
tabIndex	現在のドキュメントに対するタブ移動における現在の要素の順番を指定する。この値は 0 から 32767 の範囲の数字である必要がある。	
wrap	有効な値: 「virtual」「physical」などのブラウザがサポートする有効なラップ値。 HTML の <code>textarea</code> 要素の <code>wrap</code> 属性に生成する値を指定する。 省略された場合は、生成される HTML から <code>wrap</code> 属性が省略される。 このタグはまた、次の JavaScript イベントのサポートもしている。 <code>onBlur</code> 、 <code>onChange</code> 、 <code>onClick</code> 、 <code>onDblClick</code> 、 <code>onFocus</code> 、 <code>onKeyDown</code> 、 <code>onKeyPress</code> 、 <code>onKeyUp</code> 、 <code>onMouseDown</code> 、 <code>onMouseMove</code> 、 <code>onMouseOut</code> 、 <code>onMouseOver</code> 、 <code>onMouseUp</code> 、 <code>onSelect</code>	RT 式

(* 前の表の `formatMask` 属性を参照)

書式の種類	書式のマスク	値の例	生成される値
string	???-??-????	123456789	123-45-6789
string	??HELLO??	abcd	abHELLOcd
string	??HELLO\ \???	abcd	abHELLO?cd
dec	###0; (-#)	1000000.99	1000001
dec	#,##0.00; (-#)	1000000.99	1,000,000.99
dec	###0.00; (-#)	1000000.99	1000000.99
dec	###0; (-#)	-1000000.99	(-1000001)
dec	###0; (#)	-1000000.99	(1000001)
dec	#,##0.00; -#	-1000000.99	-1,000,000.99

第24章

textField

<jato:textField>

HTML の `<input type="text">` 要素を生成します。

このタグは、HTML の `<form>` 要素や `useViewBean`、`containerView`、`tiledView` タグで囲まれている場合にのみ有効です。このタグに本体内容を含めることはできません。

例：

```
<jato:textField name="firstName" size="12" maxLength="32"/>
```

これは、次の HTML になります。

```
<input type="text" name="PageFoo.firstName" value="John"
maxLength="32" size="12">
```

属性名	説明	注
<code>accessKey</code>	生成される HTML 要素にアクセスキーを割り当てる。一般には、このアクセスキーは、ラベルテキストまたはアクセスキーの適用先に取り込まれる。	RT 式
<code>defaultValue</code>	<code>DisplayField</code> ビューピアの値が <code>null</code> の場合に使用する値。 <code>textField</code> タグの場合、この値は HTML の <code>text</code> 要素の <code>value</code> 属性になる値を表す。	RT 式
<code>disabled</code>	有効な値：「true」か「false」 ユーザー入力用のこの HTML コントロールを使用不可にする。 <ul style="list-style-type: none">使用不可のコントローラにはフォーカスは移らない。タブによる移動で、使用不可のコントロールは飛ばされる。 省略された場合は、「false」と見なされる。	RT 式

属性名	説明	注
elementId	<p>要素に名前を割り当てる。この名前は HTML ドキュメント内で一意である必要がある。これは HTML の id 属性に相応し、Web アプリケーションフレームワークには具体的な意味を持たないことに注意。</p> <p>id 属性には HTML でいくつかの役割がある。</p> <ul style="list-style-type: none"> • スタイルシートセレクタ • ハイパーテキストリンク用のターゲットアンカー • スクリプトから特定の要素を参照する手段 	RT 式
extraHtml	この HTML 要素の終了 (>) 直前に生成される任意の HTML	RT 式
formatMask	<p>formatMask 属性の有効な値は、formatType 属性の値によって異なる。</p> <p>formatType の値が「string」または「alpha」の場合：</p> <p>フォーマットマスクは、静的テキストと予約メタキャラクタの「?」の任意の組み合わせから構成される。1 文字のメタキャラクタ「?」が、値を左から順に書式化する際に単一文字のプレースホルダとしての働きを持つ。リテラルの「?」は、「?」を「\?」でエスケープすることによって指定できる。</p> <p>formatType の値が「dec」または「curr」の場合：</p> <p>フォーマットマスクは、セミコロンで区切られた 2 つのパターン、主パターン (精度と桁区切り文字) と負数パターンで構成される (パターンの詳細は、<code>java.text.DecimalFormat</code> を参照)。</p> <p>(* この表のあとの表を参照)</p>	
formatType	<p>有効な値：「string」(「alpha」でも可)「dec」「curr」「date」のどれか</p> <p>各値は特殊なマスク規則を持つ (formatMask 属性を参照)。</p> <p>値「string」は、単純なテキストマスクを適用することによる任意のテキストの書式化をサポートする。これで、自由に動的コンテンツと静的テキストを組み合わせることができる。値「alpha」は「string」と同じ意味。</p> <p>値「dec」は、数値マスクを適用することによる数値の書式化をサポートする。</p> <p>値「curr」は、通貨マスクを適用することによる通貨値の書式化をサポートする。通貨の書式は小数の指定を受け取り、ロケールに固有の通貨書式を適用する。現在、未解決のバグのため、通貨の書式化は機能していない。</p> <p>値「date」は有効ではあるが、現状、日付の書式化をサポートする実装はない。これは、<code>java.text.DateFormat</code> が、<code>taglib</code> 実装の必要とする一般的な書式化サポートにするのに適していないためである。この制約の回避策を調査中である。</p> <p>このため、日付を書式化する場合は、<code>begin<fieldName>Display(ChildDisplayEvent event)</code> か <code>end<fieldName>Display(ChildContentDisplayEvent event)</code>、またはその両方を適用し、そこで <code>java.text.DateFormat</code> クラスに用意されているデータ書式化サポートを明示的に利用することを推奨。</p> <p>Web アプリケーションフレームワーク実装のテキストの書式化は、<code>java.text</code> パッケージとそこに含まれている書式関係のクラスに大きく依拠している。詳細は、そのパッケージを参照。また、<code><JATO package>.util.HtmlUtil</code> クラスも参照のこと。</p>	
maxLength	<p>有効な値：正数</p> <p>ユーザーがフィールドに入力できる最大文字数を指定する。</p> <p>省略された場合は、フィールドに入力できる文字数の制限はない。</p>	RT 式

属性名	説明	注
name	<p>DisplayField ビアの名前。このピア、現在の親の ContainerView、TiledView、ViewBean いずれかの子である必要がある。DisplayField 名は、現在の親ビューを基準に解決される。</p> <p>ビューピアは、<JATO package>.view.DisplayField 型である必要がある。この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照していると見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。</p> <p>例：</p> <pre>/header/orderList/customerName (ルートビューからの絶対位置) orderList/customerName (現在のコンテナからの相対位置) ../footer/orderList/customerName (親からの相対位置)</pre>	必須
readOnly	<p>有効な値：「true」か「false」</p> <p>値「true」は、この要素に含まれるテキストをユーザーが変更できないことを示す。</p> <p>省略された場合は、「false」と見なされる。</p>	RT 式
size	<p>有効な値：正数</p> <p>文字数でこのコントロールの初期幅を指定する。</p> <p>省略された場合は、生成される HTML から size 属性が省略される。</p>	RT 式
style	この HTML 要素に適用する CSS スタイル	RT 式
styleClass	この HTML 要素に適用する CSS スタイルシートクラス	RT 式
tabIndex	<p>現在のドキュメントに対するタブ移動における現在の要素の順番を指定する。この値は 0 から 32767 の範囲の数字である必要がある。</p> <p>このタグはまた、次の JavaScript イベントのサポートもしている。</p> <p>onBlur、onChange、onClick、onDbClick、onFocus、onKeyDown、onKeyPress、onKeyUp、onMouseDown、onMouseMove、onMouseOut、onMouseOver、onMouseUp、onSelect</p>	

(* 前の表の formatMask 属性を参照)

書式の種類	書式のマスク	値の例	生成される値
string	??-??-????	123456789	123-45-6789
string	??HELLO??	abcd	abHELLOcd
string	??HELLO\???	abcd	abHELLO?cd
dec	###0; (-#)	1000000.99	1000001

書式の種類	書式のマスク	値の例	生成される値
dec	#,##0.00; (-#)	1000000.99	1,000,000.99
dec	###0.00; (-#)	1000000.99	1000000.99
dec	###0; (-#)	-1000000.99	(-1000001)
dec	###0; (#)	-1000000.99	(1000001)
dec	#,##0.00; -#	-1000000.99	-1,000,000.99

第25章

validatingTextArea

<jato:validatingTextArea>

HTML の <textarea> 要素を生成します。

このタグは、HTML の <form> 要素や `useViewBean`、`containerView` タグで囲まれている場合にのみ有効です。このタグに本体内容を含めることはできません。また、`tiledView` タグ内で使用することもできません。

このタグのピアコンポーネントの種類は、`ValidatingDisplayField` である必要があります。

例：

```
<jato:validatingTextArea name="comments" cols="80" rows="5"
wrap="virtual" />
```

これは、次の HTML になります。

```
<textarea name="PageFoo.comments" cols="80" rows="5" wrap=
"virtual">
...
</textarea>
```

属性名	説明	注
accessKey	生成される HTML 要素にアクセスキーを割り当てる。一般には、このアクセスキーは、ラベルテキストまたはアクセスキーの適用先に取り込まれる。	RT 式
cols	有効な値：正数 平均的な文字幅でフィールドの表示幅を指定する。	必須、RT 式
defaultValue	DisplayField ビューピアの値が null の場合に使用する値。 validatingTextArea タグの場合、この値は、textarea 要素の開始タグと終了タグの間に生成されるテキストを表す。	RT 式
disabled	有効な値：「true」か「false」 ユーザー入力用のこの HTML コントロールを使用不可にする。 <ul style="list-style-type: none"> • 使用不可のコントローラにはフォーカスは移らない。 • タブによる移動で、使用不可のコントロールは飛ばされる。 省略された場合は、「false」と見なされる。	RT 式
elementId	要素に名前を割り当てる。この名前は HTML ドキュメント内で一意である必要がある。これは HTML の id 属性に相応し、Web アプリケーションフレームワークには具体的な意味を持たないことに注意。 id 属性には HTML でいくつかの役割がある。 <ul style="list-style-type: none"> • スタイルシートセレクタ • ハイパーテキストリンク用のターゲットアンカー • スクリプトから特定の要素を参照する手段 	RT 式
extraHtml	<textarea> 要素の終了 (>) 直前に生成される任意の HTML	RT 式
formatMask	formatMask 属性の有効な値は、formatType 属性の値によって異なる。 formatType の値が「string」または「alpha」の場合： フォーマットマスクは、静的テキストと予約メタキャラクタの「?」の任意の組み合わせから構成される。1 文字のメタキャラクタ「?」が、値を左から順に書式化する際に単一文字のプレースホルダとしての働きを持つ。リテラルの「?」は、「?」を「\」でエスケープすることによって指定できる。 formatType の値が「dec」または「curr」の場合： フォーマットマスクは、セミコロンで区切られた 2 つのパターン、主パターン (精度と桁区切り文字) と負数パターンで構成される (パターンの詳細は、java.text.DecimalFormat を参照)。 (* この表のあとの表を参照)	

属性名	説明	注
formatType	<p>有効な値: 「string」(「alpha」でも可)「dec」「curr」「date」のどれか 各値は特殊なマスク規則を持つ (formatMask 属性を参照)。 値「string」は、単純なテキストマスクを適用することによる任意のテキストの書式化をサポートする。これで、自由に動的コンテンツと静的テキストを組み合わせることができる。値「alpha」は「string」と同じ意味。 値「dec」は、数値マスクを適用することによる数値の書式化をサポートする。 値「curr」は、通貨マスクを適用することによる通貨値の書式化をサポートする。通貨の書式は小数の指定を受け取り、ロケールに固有の通貨書式を適用する。現在、未解決のバグのため、通貨の書式化は機能していない。 値「date」は有効ではあるが、現状、日付の書式化をサポートする実装はない。これは、java.text.DateFormat が、taglib 実装の必要とする一般的な書式化サポートにするのに適していないためである。この制約の回避策を調査中である。 このため、日付を書式化する場合は、 begin<fieldName>Display(ChildDisplayEvent event) か end<fieldName>Display(ChildContentDisplayEvent event)、またはその両方を適用し、そこで java.text.DateFormat クラスに用意されているデータ書式化サポートを明示的に利用することを推奨。 Web アプリケーションフレームワーク実装のテキストの書式化は、java.text パッケージとそこに含まれている書式関係のクラスに大きく依拠している。詳細は、そのパッケージを参照。また、<JATO package>.util.HtmlUtil クラスも参照のこと。</p>	
name	<p>ValidatingDisplayField ビアの名前。このビア、現在の親の ContainerView か ViewBean いずれかの子である必要がある。名前は、現在の親ビューを基準に解決される。 ビューピアは、<JATO package>.view.DisplayField 型である必要がある。 この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照していると見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。 例： /header/orderList/customerName (ルートビューからの絶対位置) orderList/customerName (現在のコンテナからの相対位置) ../footer/orderList/customerName (親からの相対位置)</p>	必須
readOnly	<p>有効な値: 「true」か「false」 値「true」は、この要素に含まれるテキストをユーザーが変更できないことを示す。 省略された場合は、「false」と見なされる。</p>	RT 式
rows	<p>有効な値: 正数 生成するコントロールに含める表示テキスト行数を指定する。</p>	必須、RT 式
style	この HTML 要素に適用する CSS スタイル	RT 式

属性名	説明	注
styleClass	この HTML 要素に適用する CSS スタイル	RT 式
tabIndex	現在のドキュメントに対するタブ移動における現在の要素の順番を指定する。この値は 0 から 32767 の範囲の数字である必要がある。	
wrap	有効な値：「virtual」「physical」などのブラウザがサポートする有効なラップ値。 HTML の textarea 要素の wrap 属性に生成する値を指定する。 省略された場合は、生成される HTML から wrap 属性が省略される。 このタグはまた、次の JavaScript イベントのサポートもしている。 onBlur、onChange、onClick、onDbClick、onFocus、onKeyDown、onKeyPress、onKeyUp、onMouseDown、onMouseMove、onMouseOut、onMouseOver、onMouseUp、onSelect	RT 式

(* 前の表の formatMask 属性を参照)

書式の種類	書式のマスク	値の例	生成される値
string	???-??-????	123456789	123-45-6789
string	??HELLO??	abcd	abHELLOcd
string	??HELLO\ \ ???	abcd	abHELLO?cd
dec	###0; (-#)	1000000.99	1000001
dec	#,##0.00; (-#)	1000000.99	1,000,000.99
dec	###0.00; (-#)	1000000.99	1000000.99
dec	###0; (-#)	-1000000.99	(-1000001)
dec	###0; (#)	-1000000.99	(1000001)
dec	#,##0.00; -#	-1000000.99	-1,000,000.99

第26章

validatingTextField

<jato:validatingTextField>

HTML の `<input type="text">` 要素を生成します。

このタグは、HTML の `<form>` 要素や `useViewBean`、`containerView` タグで囲まれている場合にのみ有効です。このタグに本体内容を含めることはできません。また、`tiledView` タグ内で使用することもできません。

このタグのピアコンポーネントの種類は、`ValidatingDisplayField` である必要があります。

例：

```
<jato:validatingTextField name="firstName" size="12"
  maxLength="32" />
```

これは、次の HTML になります。

```
<input type="text" name="PageFoo.firstName" value="John"
  maxLength="32" size="12">
```

属性名	説明	注
accessKey	生成される HTML 要素にアクセスキーを割り当てる。一般には、このアクセスキーは、ラベルテキストまたはアクセスキーの適用先に取り込まれる。	RT 式
defaultValue	DisplayField ビューピアの値が null の場合に使用する値。 validatingTextField タグの場合、この値は HTML の text 要素の value 属性になる値を表す。	RT 式
disabled	有効な値: 「true」か「false」 ユーザー入力用のこの HTML コントロールを使用不可にする。 <ul style="list-style-type: none"> • 使用不可のコントロールにはフォーカスは移らない。 • タブによる移動で、使用不可のコントロールは飛ばされる。 省略された場合は、「false」と見なされる。	RT 式
elementId	要素に名前を割り当てる。この名前は HTML ドキュメント内で一意である必要がある。これは HTML の id 属性に相応し、Web アプリケーションフレームワークには具体的な意味を持たないことに注意。 id 属性には HTML でいくつかの役割がある。 <ul style="list-style-type: none"> • スタイルシートセレクタ • ハイパーテキストリンク用のターゲットアンカー • スクリプトから特定の要素を参照する手段 	RT 式
extraHtml	この HTML 要素の終了 (>) 直前に生成される任意の HTML	RT 式
formatMask	formatMask 属性の有効な値は、formatType 属性の値によって異なる。 formatType の値が「string」または「alpha」の場合: フォーマットマスクは、静的テキストと予約メタキャラクタの「?」の任意の組み合わせから構成される。1 文字のメタキャラクタ「?」が、値を左から順に書式化する際に単一文字のプレースホルダとしての働きを持つ。リテラルの「?」は、「?」を「\?」でエスケープすることによって指定できる。 formatType の値が「dec」または「curr」の場合: フォーマットマスクは、セミコロンで区切られた 2 つのパターン、主パターン (精度と桁区切り文字) と負数パターンで構成される (パターンの詳細は、java.text.DecimalFormat を参照)。 (* この表のあとの表を参照)	

属性名	説明	注
formatType	<p>有効な値：「string」（「alpha」でも可）「dec」「curr」「date」のどれか 各値は特殊なマスク規則を持つ（formatMask 属性を参照）。</p> <p>値「string」は、単純なテキストマスクを適用することによる任意のテキストの書式化をサポートする。これで、自由に動的コンテンツと静的テキストを組み合わせることができる。値「alpha」は「string」と同じ意味。</p> <p>値「dec」は、数値マスクを適用することによる数値の書式化をサポートする。</p> <p>値「curr」は、通貨マスクを適用することによる通貨値の書式化をサポートする。通貨の書式は小数の指定を受け取り、ロケールに固有の通貨書式を適用する。現在、未解決のバグのため、通貨の書式化は機能していない。</p> <p>値「date」は有効ではあるが、現状、日付の書式化をサポートする実装はない。これは、java.text.DateFormat が、taglib 実装の必要とする一般的な書式化サポートにするのに適していないためである。この制約の回避策を調査中である。</p> <p>このため、日付を書式化する場合は、 begin<fieldName>Display(ChildDisplayEvent event) か end<fieldName>Display(ChildContentDisplayEvent event)、またはその両方を適用し、そこで java.text.DateFormat クラスに用意されているデータ書式化サポートを明示的に利用することを推奨。</p> <p>Web アプリケーションフレームワーク実装のテキストの書式化は、java.text パッケージとそこに含まれている書式関係のクラスに大きく依拠している。詳細は、そのパッケージを参照。また、<JATO package>.util.HtmlUtil クラスも参照のこと。</p>	
maxLength	<p>有効な値：正数</p> <p>ユーザーがフィールドに入力できる最大文字数を指定する。</p> <p>省略された場合は、フィールドに入力できる文字数の制限はない。</p>	RT 式
name	<p>ValidatingDisplayField ピアの名前。このピア、現在の親の ContainerView か ViewBean いずれかの子である必要がある。名前は、現在の親ビューを基準に解決される。</p> <p>ビューピアは、<JATO package>.view.DisplayField 型である必要がある。</p> <p>この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照していると見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。</p> <p>例：</p> <pre> /header/orderList/customerName (ルートビューからの絶対位置) orderList/customerName (現在のコンテナからの相対位置) ../footer/orderList/customerName (親からの相対位置) </pre>	必須
readOnly	<p>有効な値：「true」か「false」</p> <p>値「true」は、この要素に含まれるテキストをユーザーが変更できないことを示す。</p> <p>省略された場合は、「false」と見なされる。</p>	RT 式

属性名	説明	注
size	有効な値：正数 文字数でこのコントロールの初期幅を指定する。 省略された場合は、生成される HTML から size 属性が省略される。	RT 式
style	この HTML 要素に適用する CSS スタイル	RT 式
styleClass	この HTML 要素に適用する CSS スタイルシートクラス	RT 式
tabIndex	現在のドキュメントに対するタブ移動における現在の要素の順番を指定する。この値は 0 から 32767 の範囲の数字である必要がある。 このタグはまた、次の JavaScript イベントのサポートもしている。 onBlur、onChange、onClick、onDblClick、onFocus、onKeyDown、onKeyPress、onKeyUp、onMouseDown、onMouseMove、onMouseOut、onMouseOver、onMouseUp、onSelect	

(* 前の表の formatMask 属性を参照)

書式の種類	書式のマスク	値の例	生成される値
string	???-??-????	123456789	123-45-6789
string	??HELLO??	abcd	abHELLOcd
string	??HELLO\\???	abcd	abHELLO?cd
dec	###0; (-#)	1000000.99	1000001
dec	#,##0.00; (-#)	1000000.99	1,000,000.99
dec	###0.00; (-#)	1000000.99	1000000.99
dec	###0; (-#)	-1000000.99	(-1000001)
dec	###0; (#)	-1000000.99	(1000001)
dec	#,##0.00; -#	-1000000.99	-1,000,000.99

第27章

content

<jato:content>

`begin<name>Display()` イベントとそれに関連付けられている
`end<name>Display()` イベントを持つ部分が JSP コンテンツであることを示します。これらイベントの関連付けは、親コンテナビューで指定された名前のピアを必要としないことを除いて可視タグのときと同様に機能します。

このタグの本体の全内容は、対応する `begin<name>Display()` イベントから `false` 値が返されると条件付きで無効になります。無効でない場合は、処理された本体の内容は `end<name>Display()` イベントに提供されます。本体の内容は、このイベントに渡される前に、埋め込まれた JSP または Web アプリケーションフレームワーク /JATO タグを含めて完全に処理されます。

タグそのものまたはその親タグの組み合わせによって無効にできる、可視タグの場合の表示イベントのトリガーと異なり、このタグに対する表示イベントは必ずトリガーされます。このため、効率性のため、開発者は、`content` タグ用の表示イベントを少なくとも 1 つ実装することを推奨します。実装されていない場合、このタグを使用する意味はありません。

このタグは、`useViewBean` か `containerView`、または `tiledView` タグで囲まれている場合にのみ有効です。

例：

```
...
Username: <jato:textField name="username"/>
<jato:content name="superuser">
<br>
Password: <jato:textField name="password"/>
```

```
</jato:content>
<hr>
...
```

これは、現在の親ビュー内の `beginSuperuserDisplay()` イベントから `true` が返された場合にのみ次の HTML になります。

```
...
Username: <input type="text" name="PageFoo.username" value=
"scott">
<br>
Password: <input type="text" name="PageFoo.password" value=
"tiger">
<hr>
...
```

`endSuperuserDisplay()` イベントは、メソッドのパラメータとして次の HTML を受け取ります。

```
<br>
Password: <input type="text" name="PageFoo.password" value=
"tiger">
```

`beginSuperuserDisplay()` イベントから `false` が返された場合は、以下が生成されます。

```
...
Username: <input type="text" name="PageFoo.username" value=
"scott">
<hr>
...
```

`endSuperuserDisplay()` イベントはトリガーされません。

属性名	説明	注
name	コンテンツ部の名前で、現在の親ビュー内での表示イベントのトリガーに使用される。	必須
roles	コマンドで区切られたロール名のリスト。ロール名は、配備記述子ファイル (Web アプリケーションの WEB-INF ディレクトリにある web.xml) に宣言されているものに限られる。この属性がある場合は、表示イベントはトリガーされない。リストにあるロールのいずれかが現在の要求の認証された主体 (ユーザー) のロールで見つかった場合は、このタグの本体の内容が表示される。つまり、ログインしたユーザーがリストにあるロールを最低 1 つは持っていない限り、このタグによってラップされた JSP の内容はスキップされる。	
negate	有効な値: 「true」か「false」 「true」に設定すると、表示イベントまたはロール認証に基づくタグによって行われたテストの結果を、無効化または逆転させる。たとえば、表示イベントが「false」を返した場合またはロール属性のリストにあるロールが不十分な場合でも content タグの本文は表示される。 省略された場合は、「false」と見なされる。	

第28章

pagelet

<jato:pagelet>

取り込まれた JSP フラグメント内の Web アプリケーションフレームワーク/JATO タグが取り囲むページのコンテナビュースコープを継承して、要求時に取り込まれることを可能にします。

このタグは、取り囲まれた JSP フラグメント(「ページレット」ともいう)内で使用されている場合にのみ有効です。pagelet タグは、ページレット内の他のすべての Web アプリケーションフレームワーク/JATO タグを取り囲みます。

例:

EnclosingPage.jsp

```
<%@page info="E0130" language="java"%>
<%@taglib uri="/WEB-INF/jato.tld" prefix="jato"%>

<jato:useViewBean ...>
<jato:containerView name="foo">
    ...
    <jsp:include page="MyPagelet.jsp" />
    ...
</jato:containerView>
</jato:useViewBean>
```

MyPagelet.jsp

```
<%@page info="MyPagelet" language="java"%>
<%@taglib uri="/WEB-INF/jato.tld" prefix="jato"%>

<jato:pagelet>
  <jato:combobox name="month"/>...
</jato:pagelet>
```

第29章

treeNode

<jato:treeNode>

一致するツリーノードがあるたびに生成する任意の JSP コンテンツ部を示します。**treeView** タグには、複数の **treeNode** タグが含まれることがあります。**TreeView** の横断のたびに、そこに含まれている **treeNode** タグに、現在のノードに一致しているものがないか評価されます。**treeNode** タグが現在のノードに一致している場合は、その **treeNode** タグの内容が生成されます。

treeNode タグは、次の条件のすべてが **true** の場合に、現在の **TreeView** ノードと「一致している」とみなされます (タグ属性が指定されていない場合は、あらゆる比較が **true** とみなされる)。

- 現在のノード名がタグ属性の **nodeName** に一致している。
- 現在のノードタイプがタグ属性の **nodeType** に一致している。
- 現在のノードの子がタグ属性の **hasChildren** に一致している。
- 現在のノードの展開状態が、タグ属性の **isExpanded** に一致している。

nodeName と **nodeType** 属性は、一致条件をコンマ区切りのリストで指定することができます。現在のノードの展開状態は、そのノードに関連付けられている **treeNodeHandle** タグによるユーザーとの対話で決まります。

このタグは、最終的に **treeView** タグで囲まれている場合に有効です。このタグが、取り囲んでいる **treeView** タグの直接の子タグである必要はなく、実際には、**treeView** 親タグの中に自由に入れ子にすることができます。他の **treeNode** タグの中に含まれていてもかまいません。

例：

```
...  
<jato:treeView name="treeMenu">
```

```

<jato:treeNode nodeName="menuCategory" hasChildren="true" >
  <jato:href name="category"/>
</jato:treeNode>

<jato:treeNode nodeName="menuCategory" hasChildren="false"
>
  <jato:text name="category"/>
</jato:treeNode>

<jato:treeNode nodeName="menuChoice,menuItem">
  <jato:href name="item"/>
</jato:treeNode>

</jato:treeView>

...

```

属性名	説明	注
nodeName	有効な値：コンマで区切られたノード名からなる任意の文字列。 現在のノード名に一致するものを指定する。この属性には、現在のノードに一致する、指定した名前の中のどれかを意図して、コンマ区切りのノード名のリストを指定できる。 省略された場合、ノードの一致評価でこの属性は考慮されない。	RT 式
nodeType	有効な値：コンマで区切られたノードタイプからなる任意の文字列。 現在のノードタイプに一致するものを指定する。この属性には、現在のノードに一致する、指定したタイプのうちのどれかを意図して、コンマ区切りのノード名のリストを指定できる。 省略された場合、ノードの一致評価でこの属性は考慮されない。	RT 式
hasChildren	有効な値：「true」か「false」 値「true」は、現在のノードに子がある場合にのみ現在のタグが現在のノードに一致することを示す。省略された場合、ノードの一致評価でこの属性は考慮されない。	RT 式
isExpanded	有効な値：「true」か「false」 値「true」は、現在のノードがすでに展開されている場合にのみ現在のタグが現在のノードに一致することを示す。省略された場合、ノードの一致評価でこの属性は考慮されない。	RT 式

第30章

treeLevel

<jato:treeLevel>

一致するツリーノードレベルがあるたびに生成する任意の JSP コンテンツ部を示します。treeView タグには、複数の treeLevel タグが含まれることがあります。TreeLevel の横断のたびに、そこに含まれる treeLevel タグに、現在のノードの深さレベルに一致しているものがないか評価されます。treeLevel タグが現在のノードの深さレベルに一致している場合は、そのタグの内容が、現在のレベルに等しい回数生成されます。

treeLevel タグは、次の条件のすべてが true の場合に、現在のレベルと「一致している」と見なされます (タグ属性が指定されていない場合は、あらゆる比較が true と見なされる)。

- 現在のレベル \geq タグ属性の minLevel
- 現在のレベル \leq タグ属性の maxLevel

treeLevel タグが現在のツリーの深さレベルに一致している場合は、埋め込まれているすべての JSP または Web アプリケーションフレームワーク/JATO タグを含めて、本体の内容が、現在のレベルに等しい回数処理されます。たとえば現在のノードレベルが 4 の場合は、現在のノードについて、一致しているすべての treeLevel タグの本体の内容が 4 回生成されます。一般に、treeLevel タグは、レベルに固有のインデントやその他の構造上のマークアップの目的に使用します。

このタグは、最終的に treeView タグで囲まれている場合に有効です。このタグが、取り囲んでいる treeView タグの直接の子タグである必要はなく、実際には、treeView 親タグの中に自由に入れ子にすることができます。

例：

```
<jato:treeView name="treeMenu">
```

```

<!-- Add a space for each level of the current node -->
<jato:treeLevel>
  &nbsp;
</jato:treeLevel>

...

</jato:treeView>

```

属性名	説明	注
minLevel	有効な値：正数 一致するノードの深さの最低レベルを指定する。	RT 式
maxLevel	有効な値：正数 一致するノードの深さの最高レベルを指定する。	RT 式
offset	有効な値：正または負の整数 このタグを生成するときに適用する正または負のノードの深さレベル修飾子を指定する。オフセットが -n の場合は、(レベル -n) 回タグが生成されるのに対し、オフセット n では、(レベル + n) 回タグが生成される。	RT 式

第31章

treeNodeHandle

<jato:treeNodeHandle>

ユーザーが特定のツリーノードの展開 / 縮小状態を切り替えることができる可視コントロールを示す。このタグは、タグによって生成され、かつ起動時に親ツリービューによって自動的に処理される暗黙の URL を含む、HTML の `...` 要素として生成されます。

このタグに本体内容を含むことはできません。この内容は、生成される要素の `<a>` と `` タグの間に含まれ、目に見えるクリック可能な HREF 要素として表示されません。

このタグは、最終的に `treeView` タグで囲まれている場合にのみ有効です。 `treeNode` タグを、このタグの直接の親にすることはできません。

例:

...

```
<jato:treeNodeHandle>
  
</jato:treeNodeHandle>
```

...

属性名	説明	注
accessKey	生成される HTML 要素にアクセスキーを割り当てる。一般には、このアクセスキーは、ラベルテキストまたはアクセスキーの適用先に取り込まれる。	RT 式
disabled	有効な値：「true」か「false」 ユーザー入力用のこの HTML コントロールを使用不可にする。 <ul style="list-style-type: none"> 使用不可のコントローラにはフォーカスは移らない。 タブによる移動で、使用不可のコントロールは飛ばされる。 省略された場合は、「false」と見なされる。	RT 式
elementId	要素に名前を割り当てる。この名前は HTML ドキュメント内で一意である必要がある。これは HTML の id 属性に相応し、Web アプリケーションフレームワークには具体的な意味を持たないことに注意。 id 属性には HTML でいくつかの役割がある。 <ul style="list-style-type: none"> スタイルシートセレクタ ハイパーテキストリンク用のターゲットアンカー スクリプトから特定の要素を参照する手段 	RT 式
extraHtml	この HTML 要素の終了 (>) 直前に生成される任意の HTML	RT 式
queryParams	HREF の照会文字列に付与する任意個の名前と値のペアを指定する。 名前と値のペアは「名前 = 値」の書式に従い、値は必要に応じて JSP 作成者がエンコードした URL。追加の値と名前とのペアはそれぞれ、「&」文字で区切る必要がある。 例： queryParams="fname=Mike&lname=Jones"	RT 式
style	この HTML 要素に適用する CSS スタイル	RT 式
styleClass	この HTML 要素に適用する CSS スタイルシートクラス	RT 式
tabIndex	現在のドキュメントに対するタブ移動における現在の要素の順番を指定する。この値は 0 から 32767 の範囲の数字である必要がある。	
title	この要素に関するヒント的な情報を提供する。	RT 式
trim	有効な値：「true」か「false」 値「true」は、タグの内容から先行または後続するすべての空白を削除すべきことを示す。値「false」はそれらの空白を残す。 省略された場合は、「true」と見なされる。 このタグはまた、次の JavaScript イベントのサポートもしている。 onBlur、onChange、onClick、onDbClick、onFocus、onKeyDown、onKeyPress、onKeyUp、onMouseDown、onMouseMove、onMouseOut、onMouseOver、onMouseUp、onSelect	

datasetLocator

<jatox:datasetLocator>

現在の場所の静的なテキスト表示とターゲットのコンテナビュー (一部の `DatasetModel` にバウンドされています) によって現在表示されているレコード数を生成します。このタグは、`useViewBean`、`containerView`、`tiledView` タグの本体の内部に入れ子にされている場合のみ有効です。ただし、`tiledView` タグの内部では通常使用されません。

このタグは、HTML の `<form>` 要素や `useViewBean`、`containerView`、`tiledView` タグで囲まれている場合にのみ有効です。このタグに本体内容を含めることはできません。

例：

```
<jatox:datasetLocator name="datasetLocator1">
  <jato:containerView name="datasetLocator1">
    <%@ include file="/com/sun/jatox/view/DatasetLocator.jsp" %>
  </jato:containerView>
</jatox:datasetLocator>
```

これは、次の HTML フォームになります。

1. レコード X ~ Y / Z (例: レコード 11 ~ 20 / 53)
2. レコード X ~ Y (例: レコード 6 ~ 10)
3. レコード X / Z (例: レコード 23 / 43)
4. レコード X (例: レコード 14)

含まれているファイルは、いくつかの制限付きでカスタマイズできる事前作成されたページレットです。誤ったタグに手を加えると意図とは異なる、または予期しない動作を引き起こす可能性があります。これは、静的な内容に変更を加える、またはこのファイルをカスタマイズする際に追加の HTML マークアップに加えるためだけに使用することを推奨します。

属性名	説明	注
name	<p>DisplayField ピアの名前。このピア、現在の親の ContainerView、TiledView、ViewBean いずれかの子である必要がある。DisplayField 名は、現在の親ビューを基準に解決される。</p> <p>ビューピアは、<JATOX package>.view.DatasetLocator 型である必要がある。</p> <p>この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照しているから見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。</p> <p>例：</p> <ul style="list-style-type: none">/header/orderList/customerName (ルートビューからの絶対位置)orderList/customerName (現在のコンテナからの相対位置)../footer/orderList/customerName (親からの相対位置) <p>このタグは、JavaScript イベントをサポートしない。</p>	必須

datasetNavigator

<jatox:datasetNavigator>

4つのデータセットナビゲーションアクション(「先頭」、「前」、「次」、「最後」)を表すグラフィカルなボタンを生成します。このアクションは、コンポーネント内に構成されるターゲットコンテナビューに対して機能します。このタグは、useViewBean、containerView、tiledView タグの本体の中に入れ子にされた場合のみ有効です。ただし通常は、tiledView タグ内では使用しません。

「先頭」と「前」ボタンは、表示されているレコードがデータセット内の最初のレコードである場合は使用不可または非表示(表示しない)にすることができます。「次」、「最後」ボタンは、表示されているレコードが最後のレコード(これ以上レコードがない)である場合、同様に使用不可または非表示にすることができます。何らかの理由でデータが返されない場合には、ボタンの代わりにテキスト(no data found)が表示されます。

このタグは、HTML の <form> 要素や useViewBean、containerView、tiledView タグで囲まれている場合にのみ有効です。このタグに本体内容を含めることはできません。

例:

```
<jatox:datasetNavigator name="datasetNavigator1">
<jato:containerView name="datasetNavigator1">
<% include file="/com/sun/jatox/view/DatasetNavigator.jsp" %>
</jato:containerView>
</jatox:datasetNavigator>
```

これは、次の HTML フォーム (太字は使用可能、太字でないものは使用不可) になります。

1. [**<<**] [**<**] [**>**] [**>>**]
2. [**<<**] [**<**] [**>**] [**>>**]
3. [**<<**] [**<**] [**>**] [**>>**]
4. [**<<**] [**<**]
5. [**>**] [**>>**]

含まれているファイルは、いくつかの制限付きでカスタマイズできる事前作成されたページレットです。誤ったタグに手を加えると意図とは異なる、または予期しない動作を引き起こす可能性があります。これは、静的な内容に変更を加える、またはこのファイルをカスタマイズする際に追加の HTML マークアップに加えるためだけに使用することを推奨します。

属性名	説明	注
name	<p>DisplayField ピアの名前。このピア、現在の親の ContainerView、TiledView、ViewBean いずれかの子である必要がある。DisplayField 名は、現在の親ビューを基準に解決される。</p> <p>ビューピアは、<JATOX package>.view.DatasetLocator 型である必要がある。</p> <p>この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照しているから見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。</p> <p>例：</p> <ul style="list-style-type: none">/header/orderList/customerName (ルートビューからの絶対位置)orderList/customerName (現在のコンテナからの相対位置)../footer/orderList/customerName (親からの相対位置) <p>このタグは、JavaScript イベントをサポートしない。</p>	必須

第34章

dateTimeView

<jatox:dateTimeView>

6つのコンボボックス(月、日、年の範囲、時、分、am/pm オプション)を生成します。オプションとして、小さなカレンダーをポップアップして選択した日付をコンボボックスに反映させることができる、グラフィカルなボタンも生成できます。このタグは、useViewBean か containerView、または tiledView タグの本体内に入れ子になっている場合にのみ有効です。

このコンポーネントでポップアップカレンダープロパティが有効になっている場合、タグには JavaScript が必要です。この JavaScript は、自動的に次のように JSP のヘッダーセクションに含まれます。

```
<script language="JavaScript" type="text/javascript" src='<%=
request.getContextPath()+
"/com/sun/jatox/view/popupminicalendar.js"%>'></script>
```

このタグは、HTML の <form> 要素や useViewBean、containerView、tiledView タグで囲まれている場合にのみ有効です。このタグに本体内容を含めることはできません。

例:

```
<jatox:dateTimeView name="dateTimeView1"/>
```

これは、次の HTML になります (数多くあるため、すべての HTML ではありません)。

```
[month combo box] [day combo box] [year range combo box]
[hour combo box]: [minute combo box] [am/pm combo box]
[popup calendar graphic button]
```

属性名	説明	注
name	<p>DisplayField ピアの名前。このピア、現在の親の ContainerView、TiledView、ViewBean いずれかの子である必要がある。DisplayField 名は、現在の親ビューを基準に解決される。</p> <p>ビューピアは、<JATOX package>.view.DateView 型である必要がある。</p> <p>この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照しているから見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。</p> <p>例：</p> <ul style="list-style-type: none"> /header/orderList/customerName (ルートビューからの絶対位置) orderList/customerName (現在のコンテナからの相対位置) ../footer/orderList/customerName (親からの相対位置) 	必須
<p>このタグは、JavaScript イベントをサポートしない。</p>		

dateView

<jatox:dateView>

3つのコンボボックス (月、日、年の範囲オプション) を生成します。オプションとして、小さなカレンダーをポップアップして選択した日付をコンボボックスに反映させることができる、グラフィカルなボタンも生成できます。このタグは、useViewBean か containerView、または tiledView タグの本体内に入れ子になっている場合にのみ有効です。

このコンポーネントでポップアップカレンダープロパティが有効になっている場合、タグには JavaScript が必要です。この JavaScript は、自動的に次のように JSP のヘッダーセクションに含まれます。

```
<script language="JavaScript" type="text/javascript" src='<%=
request.getContextPath()+
"/com/sun/jatox/view/popupminicalendar.js"%>'></script>
```

このタグは、HTML の <form> 要素や useViewBean、containerView、tiledView タグで囲まれている場合にのみ有効です。このタグに本体内容を含めることはできません。

例:

```
<jatox:dateView name="dateView1"/>
```

これは、次の HTML になります (数多くあるため、すべての HTML ではありません)。

```
[month combo box] [day combo box] [year range combo box]
[popup calendar graphic button]
```

属性名	説明	注
name	<p>DisplayField ピアの名前。このピア、現在の親の ContainerView、TiledView、ViewBean いずれかの子である必要がある。DisplayField 名は、現在の親ビューを基準に解決される。</p> <p>ビューピアは、<JATOX package>.view.DateView 型である必要がある。</p> <p>この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照しているから見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。</p> <p>例：</p> <ul style="list-style-type: none"> /header/orderList/customerName (ルートビューからの絶対位置) orderList/customerName (現在のコンテナからの相対位置) ../footer/orderList/customerName (親からの相対位置) 	必須
<p>このタグは、JavaScript イベントをサポートしない。</p>		

goToPage

<jatox:goToPage>

次のように、HTML を生成します。

```
<a href=" ../<ParentContainerView>;jsessionid=
82BESadfaeae?<FullQualifiedName>=&jato.pageSession=">Click Here</a>
```

このタグは、useViewBean か containerView、または tiledView タグで囲まれている場合にのみ有効です。このタグに本体内容を含むことはできません。この内容は、生成される要素の <a> と タグの間に含まれ、目に見えるクリック可能な HREF 要素として表示されます。

例：

```
<jatox:goToPage name="goToHref1">Click Here</jato:href>
```

このタグの属性は、追加属性を持たない jato:href タグの属性と同等です。

属性名	説明	注
accessKey	生成される HTML 要素にアクセスキーを割り当てる。一般には、このアクセスキーは、ラベルテキストまたはアクセスキーの適用先に取り込まれる。	RT 式
anchor	生成される HREF に付加するアンカー値。	RT 式
disabled	有効な値：「true」または「false」 ユーザー入力用のこの HTML コントロールを使用不可にする。 <ul style="list-style-type: none">• 使用不可のコントローラにはフォーカスは移らない。• タブによる移動で、使用不可のコントロールは飛ばされる。 省略された場合は、「false」と見なされる。	RT 式

属性名	説明	注
defaultValue	<p>DisplayField ビューピアの値が null の場合に使用する値。</p> <p>href タグの場合、この値は、HREF の照会文字列の一部として自動的に含まれる特殊な名前と値のペアを表す。暗黙の名前と値ペアの書式は以下のとおり。</p> <p>...?<限定表示フィールド名>=<値>&...</p> <p>この仕組みによって href は表示文字列ばかりでなく、アプリケーションに重要な値を持つことができる。</p>	RT 式
elementId	<p>要素に名前を割り当てる。この名前は HTML ドキュメント内で一意である必要がある。これは HTML の id 属性に相応し、Web アプリケーションフレームワークには具体的な意味を持たないことに注意。</p> <p>id 属性には HTML でいくつかの役割がある。</p> <ul style="list-style-type: none"> • スタイルシートセレクタ • ハイパーテキストリンク用のターゲットアンカー • スクリプトから特定の要素を参照する手段 	RT 式
extraHtml	この HTML 要素の終了 (>) 直前に生成される任意の HTML	RT 式
name	<p>DisplayField ピアの名前。このピア、現在の親の ContainerView、TiledView、ViewBean いずれかの子である必要がある。DisplayField 名は、現在の親ビューを基準に解決される。</p> <p>ビューピアは、<JATO package>.view.CommandField 型である必要がある。</p> <p>この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照していると見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。</p> <p>例：</p> <p>/header/orderList/customerName (ルートビューからの絶対位置)</p> <p>orderList/customerName (現在のコンテナからの相対位置)</p> <p>../footer/orderList/customerName (親からの相対位置)</p>	必須

第37章

maskedTextField

<jatox:maskedTextField>

次のように、HTML を生成します。

```
<input type="text" name="<fieldName>" onkeypress =' return  
anyMask(event, "null");'>
```

必要な JavaScript は、自動的に同じ JSP ファイルのヘッダーに含められます。

```
<script language="javascript" src='/<ServletContext>  
/com/sun/jatox/scripts/anyMask.js'></script>
```

このタグは、HTML の <form> 要素や useViewBean、containerView、tiledView タグで囲まれている場合にのみ有効です。このタグに本体内容を含めることはできません。

例：

```
<jatox:maskedTextField name="maskedTextField1"/>
```

このタグの属性は、追加属性を持たない jato:textField タグの属性と同等です。

属性名	説明	注
accessKey	生成される HTML 要素にアクセスキーを割り当てる。一般には、このアクセスキーは、ラベルテキストまたはアクセスキーの適用先に取り込まれる。	RT 式
defaultValue	DisplayField ビューピアの値が null の場合に使用する値。 textField タグの場合、この値は HTML の text 要素の value 属性になる値を表す。	RT 式
disabled	有効な値: 「true」または「false」 ユーザー入力用のこの HTML コントロールを使用不可にする。 <ul style="list-style-type: none"> • 使用不可のコントロールにはフォーカスは移らない。 • タブによる移動で、使用不可のコントロールは飛ばされる。 省略された場合は、「false」と見なされる。	RT 式
elementId	要素に名前を割り当てる。この名前は HTML ドキュメント内で一意である必要がある。これは HTML の id 属性に相応し、Web アプリケーションフレームワークには具体的な意味を持たないことに注意。 id 属性には HTML でいくつかの役割がある。 <ul style="list-style-type: none"> • スタイルシートセレクタ • ハイパーテキストリンク用のターゲットアンカー • スクリプトから特定の要素を参照する手段 	RT 式
extraHtml	この HTML 要素の終了 (>) 直前に生成される任意の HTML	RT 式
formatMask	formatMask 属性の有効な値は、formatType 属性の値によって異なる。 formatType の値が「string」または「alpha」の場合: フォーマットマスクは、静的テキストと予約メタキャラクタの「?」の任意の組み合わせから構成される。1 文字のメタキャラクタ「?」が、値を左から順に書式化する際に単一文字のプレースホルダとしての働きを持つ。リテラルの「?」は、「?」を「\」でエスケープすることによって指定できる。 formatType の値が「dec」または「curr」の場合: フォーマットマスクは、セミコロンで区切られた 2 つのパターン、主パターン (精度と桁区切り文字) と負数パターンで構成される (パターンの詳細は、java.text.DecimalFormat を参照)。 (* この表のあとの表を参照)	

属性名	説明	注
formatType	<p>有効な値: 「string」(「alpha」)、 「dec」、 「curr」、 「date」</p> <p>各値は特殊なマスク規則を持つ (formatMask 属性を参照)。</p> <p>値「string」は、単純なテキストマスクを適用することによる任意のテキストの書式化をサポートする。これで、自由に動的コンテンツと静的テキストを組み合わせることができる。値「alpha」は「string」と同じ意味。</p> <p>値「dec」は、数値マスクを適用することによる数値の書式化をサポートする。</p> <p>値「curr」は、通貨マスクを適用することによる通貨値の書式化をサポートする。通貨の書式は小数の指定を受け取り、ロケールに固有の通貨書式を適用する。現在、未解決のバグのため、通貨の書式化は機能していない。</p> <p>値「date」は有効ではあるが、現状、日付の書式化をサポートする実装はない。これは、java.text.DateFormat が、taglib 実装の必要とする一般的な書式化サポートにするのに適していないためである。この制約の回避策を調査中である。</p> <p>このため、日付を書式化する場合は、begin<fieldName>Display (ChildDisplayEvent event) か end<fieldName>Display (ChildContentDisplayEvent event)、またはその両方を適用し、そこで java.text.DateFormat クラスに用意されているデータ書式化サポートを明示的に利用することを推奨。</p> <p>Web アプリケーションフレームワーク実装のテキストの書式化は、java.text パッケージとそこに含まれている書式関係のクラスに大きく依拠している。詳細は、そのパッケージを参照。また、<JATO package>.util.HtmlUtil クラスも参照のこと。</p>	
maxLength	<p>有効な値: 正数</p> <p>ユーザーがフィールドに入力できる最大文字数を指定する。</p> <p>省略された場合は、フィールドに入力できる文字数の制限はない。</p>	RT 式
name	<p>DisplayField ピアの名前。このピア、現在の親の ContainerView、TiledView、ViewBean いずれかの子である必要がある。DisplayField 名は、現在の親ビューを基準に解決される。</p> <p>ビューピアは、<JATO package>.view.DisplayField 型である必要がある。この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照していると見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。</p> <p>例:</p> <pre> /header/orderList/customerName (ルートビューからの絶対位置) orderList/customerName (現在のコンテナからの相対位置) ../footer/orderList/customerName (親からの相対位置) </pre>	必須
readOnly	<p>有効な値: 「true」または「false」</p> <p>値「true」は、この要素に含まれるテキストをユーザーが変更できないことを示す。</p> <p>省略された場合は、「false」と見なされる。</p>	RT 式

属性名	説明	注
size	有効な値: 正数 文字数でこのコントロールの初期幅を指定する。 省略された場合は、生成される HTML から size 属性が省略される。	RT 式
style	この HTML 要素に適用する CSS スタイル	RT 式
styleClass	この HTML 要素に適用する CSS スタイルシートクラス	RT 式
tabIndex	現在のドキュメントに対するタブ移動における現在の要素の順番を指定する。この値は 0 から 32767 の範囲の数字である必要がある。 このタグはまた、次の JavaScript イベントのサポートもしている。 onBlur、onChange、onClick、onDblClick、onFocus、onKeyDown、onKeyPress、onKeyUp、onMouseDown、onMouseMove、onMouseOut、onMouseOver、onMouseUp、onSelect	

(* 前の表の formatMask 属性を参照)

書式の種類	書式のマスク	値の例	生成される値
string	??-??-????	123456789	123-45-6789
string	??HELLO??	abcd	abHELLOcd
string	??HELLO\ \???	abcd	abHELLO?cd
dec	###0; (-#)	1000000.99	1000001
dec	#,##0.00; (-#)	1000000.99	1,000,000.99
dec	###0.00; (-#)	1000000.99	1000000.99
dec	###0; (-#)	-1000000.99	(-1000001)
dec	###0; (#)	-1000000.99	(1000001)
dec	#,##0.00; -#	-1000000.99	-1,000,000.99

第38章

menu

<jatox:menu>

このタグに本体内容を含めることはできません。

例:

```
<jatox:menu name="menu1">
```

属性名	説明	注
name	<p>DisplayField ピアの名前。このピア、現在の親の ContainerView、TiledView、ViewBean いずれかの子である必要がある。DisplayField 名は、現在の親ビューを基準に解決される。</p> <p>ビューピアは、<JATO package>.view.Menu 型である必要がある。</p> <p>この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照しているから見なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。</p> <p>例:</p> <pre>/header/orderList/customerName (ルートビューからの絶対位置) orderList/customerName (現在のコンテナからの相対位置) ../footer/orderList/customerName (親からの相対位置)</pre>	必須

このタグは、JavaScript イベントをサポートしない。

第39章

staticBreadCrumb

<jatox:staticBreadCrumb>

このタグに本体内容を含めることはできません。

例:

```
<jatox:staticBreadCrumb name="staticBreadCrumb1">
```

属性名	説明	注
name	<p>DisplayField ピアの名前。このピア、現在の親の ContainerView、TiledView、ViewBean いずれかの子である必要がある。DisplayField 名は、現在の親ビューを基準に解決される。</p> <p>ビューピアは、<JATO package>.view.Menu 型である必要がある。</p> <p>この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照していると思なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。</p> <p>例:</p> <pre>/header/orderList/customerName (ルートビューからの絶対位置) orderList/customerName (現在のコンテナからの相対位置) ../footer/orderList/customerName (親からの相対位置)</pre>	必須

このタグは、JavaScript イベントをサポートしない。

第40章

timeView

<jatox:timeView>

3つのコンボボックス (時、分、am/pm オプション) を生成します。このタグは、useViewBean か containerView、または tiledView タグの本体内に入れ子になっている場合にのみ有効です。

このタグは、HTML の <form> 要素や useViewBean、containerView、tiledView タグで囲まれている場合にのみ有効です。このタグに本体内容を含めることはできません。

例:

```
<jatox:timeView name="timeView1"/>
```

これは、次の HTML になります (数多くあるため、すべての HTML ではありません)。

```
[hour combo box]: [minute combo box] [am/pm combo box]
```

属性名	説明	注
name	<p>DisplayField ピアの名前。このピア、現在の親の ContainerView、TiledView、ViewBean いずれかの子である必要がある。DisplayField 名は、現在の親ビューを基準に解決される。</p> <p>ビューピアは、<JATO package>.view.TimeView 型である必要がある。</p> <p>この名前は、区切り文字としてスラッシュ (/) を使った限定ビューパスでもよい。パス内の、最後のコンポーネント以外のあらゆるコンポーネントは、ContainerView か ContainerView からの派生物 (TiledView など) を参照する。相対または絶対パスのどちらも可能。名前パスがスラッシュから始まる場合は、ルートビュー (ViewBean) を基準にすると見なされる。パスがスラッシュで始まっていない場合は、現在のコンテナを基準にした子を参照していると思なされる。ドット 2 つ (..) を使って、現在のコンテナの親であるコンテナを表すことができる。</p> <p>例：</p> <ul style="list-style-type: none"> /header/orderList/customerName (ルートビューからの絶対位置) orderList/customerName (現在のコンテナからの相対位置) ../footer/orderList/customerName (親からの相対位置) 	必須
<p>このタグは、JavaScript イベントをサポートしない。</p>		

索引

B

button, 25

C

checkbox, 27
combobox, 31
containerView, 9
content, 85

D

datasetLocator, 97
datasetNavigator, 99
dateTimeView, 101
dateView, 103

F

fileUpload, 33
form, 37
frameSrc, 41

G

getDisplayFieldValue, 17
getModelFieldValue, 19

goToPage, 105

H

hidden, 43
href, 45

I

image, 49

J

JSP、非 HTML、タグライブラリの利用, 2

L

listbox, 53

M

maskedTextField, 107
menu, 111

P

pagelet, 89

R

radioButtons, 61

RT 式, 8

S

setDisplayFieldValue, 21

setModelFieldValue, 23

staticBreadCrumb, 113

T

text, 65

textArea, 69

textField, 73

tiledView, 11

timeView, 115

treeLevel, 93

treeNode, 91

treeNodeHandle, 95

treeView, 13

U

useViewBean, 15

V

validatingTextArea, 77

validatingTextField, 81

W

Web アプリケーションフレームワーク、タグライブラリとの対話の仕組み, 1

あ

値タグ, 6

アプリケーション、タグライブラリの利用, 2

か

可視 (HTML) タグ, 6

こ

コンテキストタグ, 6

そ

その他のタグ, 8

た

タグの概要, 5

タグライブラリ、アプリケーションからの利用, 2

タグリファレンス, 8

つ

ツリータグ, 8

は

パスワード, 57

ひ

非 HTML JSP、タグライブラリの利用, 2

必須, 8

へ

ページレット : JSP コンテンツの取り込み, 3

変換時取り込み, 3

よ

用語説明, 8

ら
ライブラリの概要, 1

