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

Sun Java™ Studio Enterprise 7 2004Q4

Sun Microsystems, Inc.
www.sun.com

Part No. 819-1286-10
2004 年 12 月, Revision A

原典:　*Web Application Framework Component Reference Guide*
Part No: 819-0725-10
Revision A

目次

はじめに

このガイドでは、Web アプリケーションフレームワークライブラリ内のコンポーネントを紹介します。コンポーネントは、可視コンポーネント、モデルコンポーネント、コマンドコンポーネント、不可視コンポーネントという、基本的な 4 つのコンポーネントグループに分類されます。

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。また、この章の後半にある Web アプリケーションフレームワークの関連マニュアルを読んで、Web アプリケーションフレームワークアーキテクチャや Sun Java Studio Enterprise 7 開発環境 (以降、IDE と呼ぶ) に関する知識を習得している必要があります。

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 サイトのご利用に関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、あるいは資料に関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:
AaBbCc123 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 *filename* 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 '^#define ** **XV_VERSION_STRING'**

* 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.com`SM 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 にアセンブ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

コンポーネントの概要

Web アプリケーションフレームワークコンポーネントライブラリに含まれるコンポーネントは、可視コンポーネント、モデルコンポーネント、コマンドコンポーネント、不可視コンポーネントの 4 つの基本グループに分類されます。詳細は、この後の節を参照してください。

IDE では、「拡張可能」および「拡張不可」両方の可視コンポーネントを利用することができます。

拡張可能コンポーネントとは、サブクラス化することが可能なコンポーネントです。拡張可能コンポーネントのサブクラス化は、IDE によってトランスペアレントに簡単に行われます。Web アプリケーションフレームワーク IDE のウィザードは、コンポーネントのベースクラスを拡張するアプリケーションに固有のクラスを自動的に作成します。

拡張不可コンポーネントは、可視および不可視ともに、Web アプリケーションフレームワーク IDE の使用中に通常サブクラス化されることのないコンポーネントです。コンポーネントパレットから新しい拡張不可の可視コンポーネントが選択されると、新しいサブタイプではなく、指定された名前のインスタンスが作成されます。

可視コンポーネント

可視コンポーネントは、アプリケーションのユーザーインタフェースを組むために使用するコンポーネントです。

拡張可能な可視コンポーネント

コンポーネント名	説明
基本コンテナビュー (ページレット)	他のビューを含むことが可能なビュー
基本タイルビュー	複数のタイルまたは領域の繰り返しで子のビューコンポーネントを提供できる特殊なタイプのコンテナビュー
基本ツリービュー	ツリー形式で情報を提供するのに役立つ特殊なコンテナビュー
基本 ViewBean (ページ)	ビューコンポーネント階層の最上位 (ルート) として機能できる特殊なコンテナビュー

拡張不可の可視コンポーネント

コンポーネント名	説明
ボタン	ボタンとして表現されるコマンドフィールド
チェックボックス	相互排他的な 2 つの状態フィールド値 (true/false、yes/no、on/off、A/B など) を提供するコンポーネント
コンボボックス	ドロップダウンリスト形式のインタフェースで選択肢一覧を提供する選択コンポーネント
データ駆動型コンボボックス	モデルコンポーネントから選択肢が生成される選択コンポーネント。ドロップダウンリスト形式のインタフェースで選択肢一覧を提供する。
データ駆動型リストボックス	モデルコンポーネントから選択肢が生成される選択コンポーネント。リストボックス形式で選択肢の一覧を提供する。複数の選択肢をユーザーが選択できるようにすることもできる。
データ駆動型ラジオボタン	モデルコンポーネントから選択肢が生成される選択コンポーネント。一群のラジオボタン形式で選択肢の一覧を提供する。
データセットロケータ	表示されたデータのレコードを表示する (たとえば、レコード 1 ～ 10 / 53)
データセットナビゲータ	基本 ViewBean、基本タイルビュー、基本コンテナビューと同様に、コンテナビューから一連のデータのページネーション (ナビゲーション) コントロールを可能にする 4 つのコマンドフィールドのセット。
日付表示	コンボボックスで表わされたフィールド (月、日、年) の日付を表示する可視表示フィールドコンポーネント。JavaScript のミニカレンダポップアップも含まれる。
日時表示	コンボボックスで表わされたフィールド (月、日、年、時、分など) の日付を表示する可視表示フィールドコンポーネント。JavaScript のミニカレンダポップアップも含まれる。
ファイルアップロード	ユーザーがサーバーにファイルを送信する手段を提供するコンポーネント

コンポーネント名	説明
ページ移動 (リンク)	Goto View Bean コマンド記述子を使用するために事前構成された、HREF 表示フィールドコンポーネント。
非表示フィールド	ページに不可視データを埋め込んで、そのデータを囲っているフォームが送信されたときにサーバーに送り返されるようにする手段を提供するコンポーネント。
ハイパーリンク (HREF)	ハイパーリンクとして表現されるコマンドフィールド
イメージ	ページ上にイメージを表示することを可能にするコンポーネント
リストボックス	リストボックス形式で選択肢の一覧を提供する選択コンポーネント。複数の選択肢をユーザーが選択できるようにすることもできる。
マスク付きテキストフィールド	自由形式の単一行テキスト入力フィールド
メニュー	ユーザーが Web アプリケーションにドロップダウンメニュー機能を設計するためのビューコンポーネント
パスワードフィールド	ユーザーがテキストを入力する際に、文字を表示させない手段を提供するコンポーネント。
ラジオボタン	一群のラジオボタン形式で選択肢の一覧を提供する選択コンポーネント
静的パンくずリスト	サイト構造内での現在のページのコンテキストを表示する
静的テキストフィールド	読み取り専用のテキストまたはマークアップを表示するコンポーネント
テキストフィールド	自由形式の単一行テキスト入力フィールド
テキスト領域	自由形式の複数行テキスト入力フィールド
時刻表示	コンボボックスで表わされたフィールド (時、分など) の時刻を表示する可視表示フィールドコンポーネント。
検査テキストフィールド	自由形式の単一行テキスト入力フィールド。関連付けられている妥当性検査コンポーネントでユーザー提供のテキストの妥当性を検査できます。
検査テキスト領域	自由形式の複数行テキスト入力フィールド。関連付けられている妥当性検査コンポーネントでユーザー提供のテキストの妥当性を検査できる。

モデルコンポーネント

モデルコンポーネントとは、任意のデータストア (Java クラス、CORBA オブジェクト、EJB、データベース、メインフレーム、ERP システム、トランザクションプロセッサなど) への業務委託者あるいはデータプロキシとして機能するコンポーネントです。

コンポーネント名	説明
Bean アダプタモデル	バッキングデータストアとして 1 つ以上の JavaBean を使用するモデル
クライアントセッションモデル	バッキングデータストアとして JATO クライアントセッションを使用する
カスタムモデル	`com.iplanet.model.Model` インタフェースの任意の実装版
カスタムツリーモデル	XML ドキュメントや LDAP リポジトリ、ファイルシステムなどの、ツリーやディレクトリなどの階層データ構造を利用するデータストアへのアクセス手段を提供するモデル。
ディレクトリ検索モデル	モデルを LDAP 照会の結果セットのバッキングストアとして使用する
HTTP セッションモデル	バッキングデータストアとして HTTP セッションを使用するモデル
JDBC 結果セットアダプタモデル	渡される結果セットに適応する。ユーザーは、設計時にフィールド名と型を設定できる。
JDBC SQL 照会モデル	バッキングデータストアとして 1 つ以上の RDBMS テーブルを使用するモデル
JDBC ストアドプロシージャモデル	ストアドプロシージャを実行するモデル
オブジェクトアダプタモデル	オブジェクトメンバーへの深いアクセスを指定したパス式を使用する任意のオブジェクトのフィールド、Bean プロパティ、メソッドへのアクセス手段を提供するモデル。
リソースバンドルモデル	バッキングデータストアとしてリソースバンドルを使用するモデル
単純カスタムモデル	高度なデータセット管理およびページングサポートを必要とする任意の新しいモデルに対する土台を提供するモデル
Web サービスモデル	Web サービス処理を簡単に実行することを可能にするモデル

コマンドコンポーネント

コマンドコンポーネントは任意の動作をカプセル化します。一般に、カプセル化するのは、要求処理ロジックあるいはコントローラ機能です。コマンドコンポーネントで、もっとも多く使用されるのは、コマンドフィールド (ボタン、HREF) です。

コンポーネント名	説明
基本コマンド	任意のコントローラまたは要求ハンドラコンポーネント
コマンド連鎖	順に起動する複数のコマンドコンポーネントを 1 つにつなぐコマンド

不可視コンポーネント

コンポーネント名	説明
アプリケーション属性ファクトリ	アプリケーションスコープからオブジェクトを取得するファクトリ
実行モデルとページ移動コマンド	`com.iplanet.jato.view.command.ExecuteAndForwardCommand` のインスタンスの構成
モデル実行コマンド	`com.iplanet.jato.view.command.ExecuteModelCommand` のインスタンスの構成
転送コマンド	`com.iplanet.jato.view.command.ForwardCommand` のインスタンスの構成
ページ移動コマンド	`com.iplanet.jato.view.command.GotoViewBeanCommand` のインスタンスの構成
取り込みコマンド	`com.iplanet.jato.view.command.RedirectCommand` のインスタンスの構成
リダイレクトコマンド	`com.iplanet.jato.view.command.IncludeCommand` のインスタンスの構成
正規表現妥当性検査	指定された型への変換の成否に基づく妥当性の検査
要求属性ファクトリ	要求スコープからオブジェクトを取得するファクトリ
セッション属性ファクトリ	セッションスコープからオブジェクトを取得するファクトリ
単純選択	単純な選択実装
モデル参照	`com.iplanet.jato.model.SimpleModelReference` のインスタンスの構成
型妥当性検査	JDK 1.4 正規表現を使って値の妥当性を検査する単純な検査
ユーザー定義コマンド	`com.iplanet.jato.command.Command` の任意の実装のインスタンスの構成
WebAction コマンド	`com.iplanet.jato.view.command.ExecuteAndForwardCommand` のインスタンスの構成

コンポーネントリファレンス

注の用語説明：

「必須」= 必須プロパティ

「依存」= 依存プロパティ (たとえば別のプロパティの値に依存するプロパティ)

第2章

基本コンテナビュー (ページレット)

基本コンテナビューコンポーネントは、ページレットコンポーネントともいいます。他のビジュアル開発環境のパネルコンポーネントと似たコンポーネントであり、含まれている一群のコンポーネントをグループ化して、1 つの単位として操作できるようにする手段を提供します。コンテナビューはまた大部分の複雑なコンポーネント (配信可能と配信不可の両方) の基礎ともなり、それらコンポーネントは他のページおよびページレットコンポーネントから再利用できます。すなわち、他のページやページレットコンポーネントに含めたり、それらコンポーネントの子にしたりできます。

プロパティ名	説明	注
自動削除モデル	このコンテナビューに対して「削除 WebAction」が呼び出されたときに `delete(...)` メソッドが起動される「削除型」モデルの一覧。これらのモデルは、`com.iplanet.jato.model.DeletingModel` インタフェースを実装する必要がある。	
自動実行モデル	このコンテナビューに対して「実行 WebAction」が呼び出されたときに `execute(...)` メソッドが起動される「実行型」モデルの一覧。これらのモデルは、`com.iplanet.jato.model.ExecutingModel` インタフェースを実装する必要がある。	
自動挿入モデル	このコンテナビューに対して「挿入 WebAction」が呼び出されたときに `insert(...)` メソッドが起動される「挿入型」モデルの一覧。これらのモデルは、`com.iplanet.jato.model.InsertingModel` インタフェースを実装する必要がある。	
自動検出モデル	このコンテナビューに対して「検出 WebAction」が呼び出されたときに `retrieve(...)` メソッドが起動される「検出型」モデルの一覧。これらのモデルは、`com.iplanet.jato.model.RetrievingModel` インタフェースを実装する必要がある。	
自動更新モデル	このコンテナビューに対して「更新 WebAction」が呼び出されたときに `update(...)` メソッドが起動される「更新型」モデルの一覧。これらのモデルは、`com.iplanet.jato.model.UpdatingModel` インタフェースを実装する必要がある。	

プロパティ名	説明	注
名前	コンポーネントのクラス名	必須
検査例外ハンドラ	検査例外の処理方法を示す (handleValidationException(CommandEvent) イベントまたは独自の検査コマンドコンポーネントを使用)	
表示	コンポーネントを表示するかどうかを制御する。コンポーネントの setVisible(boolean) メソッドを使い、プログラムから設定することもできる。	

第3章

基本タイルビュー

基本 TiledView (タイルビュー) は、ページレットコンポーネントの一種です。基本タイルビューは、複数のタイルまたは領域の繰り返しでその子 (ページレットや、表示フィールドなどのその他可視コンポーネント) を提供できる特殊な種類のコンテナビューです。タイル例としては、表の行や列、タブ付きコンポーネントのタブが挙げられます。表のレイアウトの前提になるものはありません。単純に、タイルの繰り返しという考え方です。

プロパティ名	説明	注
自動検出モデル	このコンテナビューに対して「検出 WebAction」が呼び出されたときに `retrieve(...)` メソッドが起動される「検出型」モデルの一覧。これらのモデルは、`com.iplanet.jato.model.RetrievingModel` インタフェースを実装する必要がある。	
タイル最大表示数	表示するタイル数を制御する。デフォルトは -1 で無制限のタイル (使用可能なタイルすべてを表示) を意味する。	
名前	コンポーネントのクラス名	必須
一次モデルデータセット名	一部のモデルは、Web サービスモデルのように複数の並列データセットを持つことができる。一次データセットとは、プログラムで特に何も指定されていない場合にデフォルトで表示されるデータセットのこと。	
一次モデル参照	タイルビューは複数のモデルに関連付けることができる。「一次」モデルは、タイルビューの反復タイル動作を制御する。	
検査例外ハンドラ	検査例外の処理方法を示す (`handleValidationException(CommandEvent)` イベントまたは独自の検査コマンドコンポーネントを使用)	
表示	コンポーネントを表示するかどうかを制御する。コンポーネントの `setVisible(boolean)` メソッドを使い、プログラムから設定することもできる。	

第4章

基本ツリービュー

基本 TreeView (ツリービュー) は、ページレットコンポーネントの一種です。基本ツリービューは、XML や LDAP データ構造のようにツリー形式の構造化された情報を提供するのに役立ちます。

プロパティ名	説明	注
名前	コンポーネントのクラス名	必須
ノードハンドル要求ハンドラ	クライアント側ではなく、サーバー側で「ツリーノード展開」要求を処理する方法を指示する。デフォルトは、ノードを展開して、ページを再度読み込む。ツリービューコンポーネントの `handleTreeNodeHandleRequest(CommandEvent)` イベントまたは独自のコマンドコンポーネントで独自の動作を定義することもできる。	
一次モデル参照	ツリービューは複数のモデルに関連付けることができる。「一次」モデルは、ツリービューの反復ノード動作を制御する。	
状態データセッション属性	ツリーの状態の格納に使用される HTTP セッション属性の名前	
検査例外ハンドラ	検査例外の処理方法を示す (`handleValidationException(CommandEvent)` イベントまたは独自の検査コマンドコンポーネントを使用)	
表示	コンポーネントを表示するかどうかを制御する。コンポーネントの `setVisible(boolean)` メソッドを使い、プログラムから設定することもできる。	

第5章

基本 ViewBean (ページ)

基本 ViewBean はページコンポーネントともいいます。基本 ViewBean は、任意の複雑なビュー階層でルートビューとして機能する特殊なビューコンポーネントです。言い替えれば ViewBean は、他のビューコンポーネントを含むことができ、それ自体は親を持たない最上位のビューコンポーネントです。ViewBean は、主としてアプリケーション内のページと見なすことができます。基本 ViewBean (または `com.iplanet.jato.view.ViewBean` インタフェースを実装したサン以外のコンポーネント) は、IDE 内から実行 (テスト実行) 可能な唯一のビューコンポーネントです。

プロパティ名	説明	注
自動削除モデル	このコンテナビューに対して「削除 WebAction」が呼び出されたときに `delete(...)` メソッドが起動される「削除型」モデルの一覧。これらのモデルは、`com.iplanet.jato.model.DeletingModel` インタフェースを実装する必要がある。	
自動実行モデル	このコンテナビューに対して「実行 WebAction」が呼び出されたときに `execute(...)` メソッドが起動される「実行型」モデルの一覧。これらのモデルは、`com.iplanet.jato.model.ExecutingModel` インタフェースを実装する必要がある。	
自動挿入モデル	このコンテナビューに対して「挿入 WebAction」が呼び出されたときに `insert(...)` メソッドが起動される「挿入型」モデルの一覧。これらのモデルは、`com.iplanet.jato.model.InsertingModel` インタフェースを実装する必要がある。	
自動検出モデル	このコンテナビューに対して「検出 WebAction」が呼び出されたときに `retrieve(...)` メソッドが起動される「検出型」モデルの一覧。これらのモデルは、`com.iplanet.jato.model.RetrievingModel` インタフェースを実装する必要がある。	

プロパティ名	説明	注
自動更新モデル	このコンテナビューに対して「更新 WebAction」が呼び出されたときに update(...) メソッドが起動される「更新型」モデルの一覧。これらのモデルは、com.iplanet.jato.model.UpdatingModel インタフェースを実装する必要がある。	
名前	コンポーネントのクラス名	必須
検査例外ハンドラ	検査例外の処理方法を示す (handleValidationException(CommandEvent) イベントまたは独自の検査コマンドコンポーネントを使用)	

第6章

ボタン

ボタンは、フォームのデータを送信するコマンドフィールドの一種です。ボタンの要求処理動作は、インスタンス別の要求ハンドラメソッド (handle<*ComponentName*>Request) に実装するか、コマンドコンポーネントに委託します。

プロパティ名	説明	注
初期値	インスタンス化したときに可視コンポーネントに初期設定する値。バインドモデルフィールドに値が存在する場合、この初期値によって上書きされることに注意。モデルフィールドの値を上書きすることなくコンポーネントに値を設定する場合は、このプロパティを使わずに、overwrite パラメータに false を設定した setValue(Object value,boolean overwrite) メソッドを使用する。このメソッドは必要に応じてコード (イベントハンドラなど) から呼び出したり、同じコンポーネントの初期化後コードプロパティから呼び出したりできる。	
モデルフィールドバインド	可視コンポーネントの値の格納/検出場所となるバインド先のモデルフィールド。このプロパティを設定するには、事前に「モデル参照」プロパティを設定しておく必要がある。	モデル参照に依存
モデル参照	可視コンポーネントのバインドモデルフィールドが属するモデルへの参照。「モデルフィールドバインド」プロパティを設定するには、事前にこのプロパティを設定しておく必要がある。	必須
名前	コンポーネントインスタンスの名前	
要求ハンドラ	コマンドフィールドに対する要求処理機構を定義する。要求イベントハンドラメソッド (handle<*ComponentName*>Request) や独自のコマンドコンポーネント、WebAction コマンドのような組み込みコマンドコンポーネントを指定できる。デフォルトの設定は、要求イベントハンドラメソッド。	
表示	コンポーネントを表示するかどうかを制御する。コンポーネントの setVisible(boolean) メソッドを使い、プログラムから設定することもできる。	

第7章

チェックボックス

チェックボックスコンポーネントは、相互排他的な 2 つの状態フィールド値 (true/false、yes/no、on/off、A/B など) を提供します。True および False 状態の実際のフィールドの型と値はカスタマイズできます。

プロパティ名	説明	注
False 値	可視コンポーネントが false 状態 (選択されていない状態) のときのフィールドの型と値	
モデルフィールドバインド	可視コンポーネントの値の格納/検出場所となるバインド先のモデルフィールド。このプロパティを設定するには、事前に「モデル参照」プロパティを設定しておく必要がある。	モデル参照に依存
モデル参照	可視コンポーネントのバインドモデルフィールドが属するモデルへの参照。「モデルフィールドバインド」プロパティを設定するには、事前にこのプロパティを設定しておく必要がある。	
名前	コンポーネントインスタンスの名前	必須
状態	チェックボックスの初期状態 (true または false 値)	
True 値	可視コンポーネントが true 状態 (選択されている状態) のときのフィールドの型と値	
表示	コンポーネントを表示するかどうかを制御する。コンポーネントの `setVisible(boolean)` メソッドを使い、プログラムから設定することもできる。	

第8章

コンボボックス

コンボボックスコンポーネントは、ドロップダウンリスト形式のインタフェースで選択肢一覧を提供する選択コンポーネントの一種です。コンポーネントの選択肢の実際のフィールドの型と値はカスタマイズできます。

プロパティ名	説明	注
選択	ユーザーが選択可能な一群の選択肢。各選択肢は値とラベルを提供する。	
初期値	インスタンス化したときに可視コンポーネントに初期設定する値。バインドモデルフィールドに値が存在する場合、この初期値によって上書きされることに注意。モデルフィールドの値を上書きすることなくコンポーネントに値を設定する場合は、このプロパティを使わずに、`overwrite` パラメータに `false` を設定した `setValue(Object value,boolean overwrite)` メソッドを使用する。このメソッドは必要に応じてコード (イベントハンドラなど) から呼び出したり、同じコンポーネントの初期化後コードプロパティから呼び出したりできる。	
モデルフィールドバインド	可視コンポーネントの値の格納/検出場所となるバインド先のモデルフィールド。このプロパティを設定するには、事前に「モデル参照」プロパティを設定しておく必要がある。	モデル参照に依存
モデル参照	可視コンポーネントのバインドモデルフィールドが属するモデルへの参照。「モデルフィールドバインド」プロパティを設定するには、事前にこのプロパティを設定しておく必要がある。	
名前	コンポーネントインスタンスの名前	必須
Null 選択ラベル	フィールドの値が null の場合にコンポーネントの選択肢として表示するテキスト。このプロパティを空 (null) のままにすると、ユーザーに null 選択肢が提供されず、ユーザーは必ず一覧から選択肢のうちの 1 つを選択する必要がある。	
表示	コンポーネントを表示するかどうかを制御する。コンポーネントの `setVisible(boolean)` メソッドを使い、プログラムから設定することもできる。	

第9章

データ駆動型コンボボックス

データ駆動型コンボボックスは、モデルコンポーネントからその選択肢を生成できる選択コンポーネントの一種です。この選択コンポーネントは、ドロップダウンリスト形式のインタフェースで選択肢一覧を提供します。

プロパティ名	説明	注
キャッシュされた選択属性名	再利用の目的で選択肢をキャッシュする際に使用する属性の名前。この値は、「選択検出ポリシー」プロパティの値に従って使用される。	
選択ラベルバインド	コンポーネントの選択肢一覧用のラベルを生成するために使用するモデルフィールド。このプロパティを設定するには、事前に「選択モデル参照」プロパティを設定しておく必要がある。このプロパティでは、複数のバインドが可能であり、それらバインドは、「選択ラベルメッセージフォーマット」プロパティを使って合体されて、単一の選択ラベルにされる。	必須 選択モデル参照に依存
選択ラベルメッセージフォーマット	未加工の選択データをフォーマットした選択ラベルに変換する際に適用するメッセージ書式文字列。この書式文字列には、プレーンテキストばかりでなく、標準的な Java メッセージフォーマットトークン ("{0}", "{1}" など) を含めることができる。各メッセージフォーマットトークンは、指定されたインデックスに対応するラベルモデルバインドの値に置き換えられる。	
選択モデル参照	値と選択ラベルモデルフィールドバインドを使って取得するコンポーネントの選択リストが存在するモデルへの参照。「選択ラベルバインド」プロパティと「選択値バインド」プロパティを設定するには、事前にこのプロパティを設定しておく必要がある。	必須

プロパティ名	説明	注
選択検出例外ハンドラ	選択値または選択ラベルがモデルから読み出されているときに例外がスローされた場合の例外の処理方法を定義する。このプロパティに指定できる設定は、デフォルト (例外をスロー)、データ駆動型選択コンポーネント検出例外ハンドラメソッド (handle<*ComponentName*>ChoicesRetrievalException)、コマンドコンポーネントへの委託のいずれか。	
選択検出ポリシー	コンポーネントの選択肢を検出し、生成する方法とタイミングを定義する。このプロパティに指定できる設定は、手動検出 (選択肢検出の開始を開発者が完全に制御)、要求ごとに 1 回、セッションごとに 1 回 (新しい HTTP セッションが作成されるたびに 1 回)、アプリケーションごとに 1 回 (仮想マシンの最初の要求時に 1 回)、あらゆるタイミング (要求内でコンポーネントが複数回使用された場合は、コンポーネントの生成のたびに 1 回選択肢が生成される)。デフォルトの設定は要求ごとに 1 回。	
選択値バインド	コンポーネントの選択肢一覧用の値を生成するために使用するモデルフィールド。このプロパティを設定するには、事前に「選択モデル参照」プロパティを設定しておく必要がある。	必須 選択モデル参照に依存
初期値	インスタンス化したときに可視コンポーネントに初期設定する値。バインドモデルフィールドに値が存在する場合、この初期値によって上書きされることに注意。モデルフィールドの値を上書きすることなくコンポーネントに値を設定する場合は、このプロパティを使わずに、overwrite パラメータに false を設定した setValue(Object value,boolean overwrite) メソッドを使用する。このメソッドは必要に応じてコード (イベントハンドラなど) から呼び出したり、同じコンポーネントの初期化後コードプロパティから呼び出したりできる。	
モデルフィールドバインド	可視コンポーネントの値の格納/検出場所となるバインド先のモデルフィールド。このプロパティを設定するには、事前に「モデル参照」プロパティを設定しておく必要がある。	モデル参照に依存
モデル参照	可視コンポーネントのバインドモデルフィールドが属するモデルへの参照。「モデルフィールドバインド」プロパティを設定するには、事前にこのプロパティを設定しておく必要がある。	必須

プロパティ名	説明	注
名前	コンポーネントインスタンスの名前	
Null 選択ラベル	フィールドの値が null の場合にコンポーネントの選択肢として表示するテキスト。このプロパティを空 (null) のままにすると、ユーザーに null 選択肢が提供されず、ユーザーは必ず一覧から選択肢のうちの 1 つを選択する必要がある。	
表示	コンポーネントを表示するかどうかを制御する。コンポーネントの `setVisible(boolean)` メソッドを使い、プログラムから設定することもできる。	

第10章

データ駆動型リストボックス

データ駆動型リストボックスは、モデルコンポーネントからその選択肢を生成できる選択コンポーネントの一種です。この選択コンポーネントは、リストボックス形式で選択肢の一覧を提供します。複数の選択肢をユーザーが選択できるようにすることもできます。

プロパティ名	説明	注
複数選択可	複数の選択肢をユーザーが選択できるようにするかどうかを定義する。デフォルトの設定は `false`。	
キャッシュされた選択属性名	再利用の目的で選択肢をキャッシュする際に使用する属性の名前。この値は、「選択検出ポリシー」プロパティの値に従って使用される。	
選択ラベルバインド	コンポーネントの選択肢一覧用のラベルを生成するために使用するモデルフィールド。このプロパティを設定するには、事前に「選択モデル参照」プロパティを設定しておく必要がある。このプロパティでは、複数のバインドが可能であり、それらバインドは、「選択ラベルメッセージフォーマット」プロパティを使って合体されて、単一の選択ラベルにされる。	必須 選択モデル参照に依存
選択ラベルメッセージフォーマット	未加工の選択データをフォーマットした選択ラベルに変換する際に適用するメッセージ書式文字列。この書式文字列には、プレーンテキストばかりでなく、標準的な Java メッセージフォーマットトークン ("{0}", "{1}" など) を含めることができる。各メッセージフォーマットトークンは、指定されたインデックスに対応するラベルモデルバインドの値に置き換えられる。	

プロパティ名	説明	注
選択モデル参照	値と選択ラベルモデルフィールドバインドを使って取得するコンポーネントの選択リストが存在するモデルへの参照。「選択ラベルバインド」プロパティと「選択値バインド」プロパティを設定するには、事前にこのプロパティを設定しておく必要がある。	必須
選択検出例外ハンドラ	選択値または選択ラベルがモデルから読み出されているときに例外がスローされた場合の例外の処理方法を定義する。このプロパティに指定できる設定は、デフォルト (例外をスロー)、データ駆動型選択コンポーネント検出例外ハンドラメソッド (`handle<`*ComponentName*`>ChoicesRetrievalException`)、コマンドコンポーネントへの委託のいずれか。	
選択検出ポリシー	コンポーネントの選択肢を検出し、生成する方法とタイミングを定義する。このプロパティに指定できる設定は、手動検出 (選択肢検出の開始を開発者が完全に制御)、要求ごとに 1 回、セッションごとに 1 回 (新しい HTTP セッションが作成されるたびに 1 回)、アプリケーションごとに 1 回 (仮想マシンの最初の要求時に 1 回)、あらゆるタイミング (要求内でコンポーネントが複数回使用された場合は、コンポーネントの生成のたびに 1 回選択肢が生成される)。デフォルトの設定は要求ごとに 1 回。	
選択値バインド	コンポーネントの選択肢一覧用の値を生成するために使用するモデルフィールド。このプロパティを設定するには、事前に「選択モデル参照」プロパティを設定しておく必要がある。	必須 選択モデル参照に依存
初期値	インスタンス化したときに可視コンポーネントに初期設定する値。バインドモデルフィールドに値が存在する場合、この初期値によって上書きされることに注意。モデルフィールドの値を上書きすることなくコンポーネントに値を設定する場合は、このプロパティを使わずに、`overwrite` パラメータに `false` を設定した `setValue(Object value,boolean overwrite)` メソッドを使用する。このメソッドは必要に応じてコード (イベントハンドラなど) から呼び出したり、同じコンポーネントの初期化後コードプロパティから呼び出したりできる。	
モデルフィールドバインド	可視コンポーネントの値の格納/検出場所となるバインド先のモデルフィールド。このプロパティを設定するには、事前に「モデル参照」プロパティを設定しておく必要がある。	モデル参照に依存

プロパティ名	説明	注
モデル参照	可視コンポーネントのバインドモデルフィールドが属するモデルへの参照。「モデルフィールドバインド」プロパティを設定するには、事前にこのプロパティを設定しておく必要がある。	
名前	コンポーネントインスタンスの名前	必須
Null 選択ラベル	フィールドの値が null の場合にコンポーネントの選択肢として表示するテキスト。このプロパティを空 (null) のままにすると、ユーザーに null 選択肢が提供されず、ユーザーは必ず一覧から選択肢のうちの 1 つを選択する必要がある。	
表示	コンポーネントを表示するかどうかを制御する。コンポーネントの `setVisible(boolean)` メソッドを使い、プログラムから設定することもできる。	

第11章

データ駆動型ラジオボタン

データ駆動型ラジオボタンは、モデルコンポーネントからその選択肢を生成できる選択コンポーネントの一種です。この選択コンポーネントは、一群のラジオボタン形式で選択肢の一覧を提供します。

プロパティ名	説明	注
キャッシュされた選択属性名	再利用の目的で選択肢をキャッシュする際に使用する属性の名前。この値は、「選択検出ポリシー」プロパティの値に従って使用される。	
選択ラベルバインド	コンポーネントの選択肢一覧用のラベルを生成するために使用するモデルフィールド。このプロパティを設定するには、事前に「選択モデル参照」プロパティを設定しておく必要がある。このプロパティでは、複数のバインドが可能であり、それらバインドは、「選択ラベルメッセージフォーマット」プロパティを使って合体されて、単一の選択ラベルにされる。	必須 選択モデル参照に依存
選択ラベルメッセージフォーマット	未加工の選択データをフォーマットした選択ラベルに変換する際に適用するメッセージ書式文字列。この書式文字列には、プレーンテキストばかりでなく、標準的な Java メッセージフォーマットトークン ("{0}", "{1}" など) を含めることができる。各メッセージフォーマットトークンは、指定されたインデックスに対応するラベルモデルバインドの値に置き換えられる。	
選択モデル参照	値と選択ラベルモデルフィールドバインドを使って取得するコンポーネントの選択リストが存在するモデルへの参照。「選択ラベルバインド」プロパティと「選択値バインド」プロパティを設定するには、事前にこのプロパティを設定しておく必要がある。	必須

プロパティ名	説明	注
選択検出例外ハンドラ	選択値または選択ラベルがモデルから読み出されているときに例外がスローされた場合の例外の処理方法を定義する。このプロパティに指定できる設定は、デフォルト (例外をスロー)、データ駆動型選択コンポーネント検出例外ハンドラメソッド (handle<*ComponentName*>ChoicesRetrievalException)、コマンドコンポーネントへの委託のいずれか。	
選択検出ポリシー	コンポーネントの選択肢を検出し、生成する方法とタイミングを定義する。このプロパティに指定できる設定は、手動検出 (選択肢検出の開始を開発者が完全に制御)、要求ごとに 1 回、セッションごとに 1 回 (新しい HTTP セッションが作成されるたびに 1 回)、アプリケーションごとに 1 回 (仮想マシンの最初の要求時に 1 回)、あらゆるタイミング (要求内でコンポーネントが複数回使用された場合は、コンポーネントの生成のたびに 1 回選択肢が生成される)。デフォルトの設定は要求ごとに 1 回。	
選択値バインド	コンポーネントの選択肢一覧用の値を生成するために使用するモデルフィールド。このプロパティを設定するには、事前に「選択モデル参照」プロパティを設定しておく必要がある。	必須 選択モデル参照に依存
初期値	インスタンス化したときに可視コンポーネントに初期設定する値。バインドモデルフィールドに値が存在する場合、この初期値によって上書きされることに注意。モデルフィールドの値を上書きすることなくコンポーネントに値を設定する場合は、このプロパティを使わずに、overwrite パラメータに false を設定した setValue(Object value,boolean overwrite) メソッドを使用する。このメソッドは必要に応じてコード (イベントハンドラなど) から呼び出したり、同じコンポーネントの初期化後コードプロパティから呼び出したりできる。	
モデルフィールドバインド	可視コンポーネントの値の格納/検出場所となるバインド先のモデルフィールド。このプロパティを設定するには、事前に「モデル参照」プロパティを設定しておく必要がある。	モデル参照に依存
モデル参照	可視コンポーネントのバインドモデルフィールドが属するモデルへの参照。「モデルフィールドバインド」プロパティを設定するには、事前にこのプロパティを設定しておく必要がある。	

プロパティ名	説明	注
名前	コンポーネントインスタンスの名前	必須
Null 選択ラベル	フィールドの値が null の場合にコンポーネントの選択肢として表示するテキスト。このプロパティを空 (null) のままにすると、ユーザーに null 選択肢が提供されず、ユーザーは必ず一覧から選択肢のうちの 1 つを選択する必要がある。	
表示	コンポーネントを表示するかどうかを制御する。コンポーネントの `setVisible(boolean)` メソッドを使い、プログラムから設定することもできる。	

第 12 章

ファイルアップロード

ファイルアップロードコンポーネントはユーザーがサーバーにファイルを送信する手段、およびアップロードされたファイルの内容に開発者がアクセスする簡単な手段を提供します。ファイルアップロードを制御するグローバルアプリケーションプロパティは、アプリケーションの「設定と構成」ノードで設定することができます。

プロパティ名	説明	注
名前	コンポーネントインスタンスの名前	必須
表示	コンポーネントを表示するかどうかを制御する。コンポーネントの `setVisible(boolean)` メソッドを使い、プログラムから設定することもできる。	

第13章

非表示フィールド

非表示フィールドコンポーネントは、ページに不可視データを埋め込んで、そのデータを囲っているフォームが送信されたときにサーバーに送り返されるようにする手段を提供します。

プロパティ名	説明	注
初期値	インスタンス化したときに可視コンポーネントに初期設定する値。バインドモデルフィールドに値が存在する場合、この初期値によって上書きされることに注意。モデルフィールドの値を上書きすることなくコンポーネントに値を設定する場合は、このプロパティを使わずに、overwrite パラメータに false を設定した setValue(Object value,boolean overwrite) メソッドを使用する。このメソッドは必要に応じてコード (イベントハンドラなど) から呼び出したり、同じコンポーネントの初期化後コードプロパティから呼び出したりできる。	
モデルフィールドバインド	可視コンポーネントの値の格納/検出場所となるバインド先のモデルフィールド。このプロパティを設定するには、事前に「モデル参照」プロパティを設定しておく必要がある。	モデル参照に依存
モデル参照	可視コンポーネントのバインドモデルフィールドが属するモデルへの参照。「モデルフィールドバインド」プロパティを設定するには、事前にこのプロパティを設定しておく必要がある。	
名前	コンポーネントインスタンスの名前	必須
表示	コンポーネントを表示するかどうかを制御する。コンポーネントの setVisible(boolean) メソッドを使い、プログラムから設定することもできる。	

第 14 章

ハイパーリンク (HREF)

ハイパーリンク (HREF) コンポーネントは、コマンドフィールドの一種です。ボタンと異なり、ハイパーリンクの起動によってフォームデータがサーバーに送信されることはありません。ハイパーリンクには、それぞれ専用の一文の照会パラメータがあります。ハイパーリンクの要求処理動作は、インスタンス別の要求ハンドラメソッド (`handle<`*ComponentName*`>Request`) に実装するか、コマンドコンポーネントに委託します。

プロパティ名	説明	注
初期値	インスタンス化したときに可視コンポーネントに初期設定する値。バインドモデルフィールドに値が存在する場合、この初期値によって上書きされることに注意。モデルフィールドの値を上書きすることなくコンポーネントに値を設定する場合は、このプロパティを使わずに、`overwrite` パラメータに `false` を設定した `setValue(Object value,boolean overwrite)` メソッドを使用する。このメソッドは必要に応じてコード (イベントハンドラなど) から呼び出したり、同じコンポーネントの初期化後コードプロパティから呼び出したりできる。	
モデルフィールドバインド	可視コンポーネントの値の格納/検出場所となるバインド先のモデルフィールド。このプロパティを設定するには、事前に「モデル参照」プロパティを設定しておく必要がある。	モデル参照に依存
モデル参照	可視コンポーネントのバインドモデルフィールドが属するモデルへの参照。「モデルフィールドバインド」プロパティを設定するには、事前にこのプロパティを設定しておく必要がある。	

プロパティ名	説明	注
名前	コンポーネントインスタンスの名前	必須
要求ハンドラ	コマンドフィールドに対する要求処理機構を定義する。要求イベントハンドラメソッド (`handle`<*ComponentName*>`Request`) や独自のコマンドコンポーネント、WebAction コマンドのような組み込みコマンドコンポーネントを指定できる。デフォルトの設定は、要求イベントハンドラメソッド。	
表示	コンポーネントを表示するかどうかを制御する。コンポーネントの `setVisible(boolean)` メソッドを使い、プログラムから設定することもできる。	

第 15 章

イメージ

イメージコンポーネントは、ページ上にイメージを表示することを可能にします。このコンポーネントは、アプリケーションによってイメージの URL が動的に決定される場合に使用することを推奨します (静的イメージは、単にページのマークアップでエンコーディング)。

プロパティ名	説明	注
初期値	インスタンス化したときに可視コンポーネントに初期設定する値。バインドモデルフィールドに値が存在する場合、この初期値によって上書きされることに注意。モデルフィールドの値を上書きすることなくコンポーネントに値を設定する場合は、このプロパティを使わずに、`overwrite` パラメータに `false` を設定した `setValue(Object value,boolean overwrite)` メソッドを使用する。このメソッドは必要に応じてコード (イベントハンドラなど) から呼び出したり、同じコンポーネントの初期化後コードプロパティから呼び出したりできる。	
モデルフィールドバインド	可視コンポーネントの値の格納/検出場所となるバインド先のモデルフィールド。このプロパティを設定するには、事前に「モデル参照」プロパティを設定しておく必要がある。	モデル参照に依存
モデル参照	可視コンポーネントのバインドモデルフィールドが属するモデルへの参照。「モデルフィールドバインド」プロパティを設定するには、事前にこのプロパティを設定しておく必要がある。	
名前	コンポーネントインスタンスの名前	必須
表示	コンポーネントを表示するかどうかを制御する。コンポーネントの `setVisible(boolean)` メソッドを使い、プログラムから設定することもできる。	

第16章

リストボックス

リストボックスコンポーネントは、リストボックスに選択肢の一覧を提供する選択コンポーネントの一種です。複数の選択肢をユーザーが選択できるようにすることもできます。

プロパティ名	説明	注
複数選択可	複数の選択肢をユーザーが選択できるようにするかどうかを定義する。デフォルトの設定は `false`。	
選択	ユーザーが選択可能な一群の選択肢。各選択肢は値とラベルを提供する。	
初期値	インスタンス化したときに可視コンポーネントに初期設定する値。バインドモデルフィールドに値が存在する場合、この初期値によって上書きされることに注意。モデルフィールドの値を上書きすることなくコンポーネントに値を設定する場合は、このプロパティを使わずに、`overwrite` パラメータに `false` を設定した `setValue(Object value,boolean overwrite)` メソッドを使用する。このメソッドは必要に応じてコード (イベントハンドラなど) から呼び出したり、同じコンポーネントの初期化後コードプロパティから呼び出したりできる。	
モデルフィールドバインド	可視コンポーネントの値の格納/検出場所となるバインド先のモデルフィールド。このプロパティを設定するには、事前に「モデル参照」プロパティを設定しておく必要がある。	モデル参照に依存
モデル参照	可視コンポーネントのバインドモデルフィールドが属するモデルへの参照。「モデルフィールドバインド」プロパティを設定するには、事前にこのプロパティを設定しておく必要がある。	

プロパティ名	説明	注
名前	コンポーネントインスタンスの名前	必須
Null 選択ラベル	フィールドの値が null の場合にコンポーネントの選択肢として表示するテキスト。このプロパティを空 (null) のままにすると、ユーザーに null 選択肢が提供されず、ユーザーは必ず一覧から選択肢のうちの 1 つを選択する必要がある。	
表示	コンポーネントを表示するかどうかを制御する。コンポーネントの `setVisible(boolean)` メソッドを使い、プログラムから設定することもできる。	

第17章

パスワードフィールド

パスワードフィールドは、ユーザーがテキストを入力する際に、文字を表示させない手段を提供します。入力された各文字の代わりにアスタリスクが表示されます。

プロパティ名	説明	注
初期値	インスタンス化したときに可視コンポーネントに初期設定する値。バインドモデルフィールドに値が存在する場合、この初期値によって上書きされることに注意。モデルフィールドの値を上書きすることなくコンポーネントに値を設定する場合は、このプロパティを使わずに、overwrite パラメータに false を設定した setValue(Object value,boolean overwrite) メソッドを使用する。このメソッドは必要に応じてコード (イベントハンドラなど) から呼び出したり、同じコンポーネントの初期化後コードプロパティから呼び出したりできる。	
モデルフィールドバインド	可視コンポーネントの値の格納/検出場所となるバインド先のモデルフィールド。このプロパティを設定するには、事前に「モデル参照」プロパティを設定しておく必要がある。	モデル参照に依存
モデル参照	可視コンポーネントのバインドモデルフィールドが属するモデルへの参照。「モデルフィールドバインド」プロパティを設定するには、事前にこのプロパティを設定しておく必要がある。	
名前	コンポーネントインスタンスの名前	必須
表示	コンポーネントを表示するかどうかを制御する。コンポーネントの setVisible(boolean) メソッドを使い、プログラムから設定することもできる。	

第18章

ラジオボタン

ラジオボタンコンポーネントは、一群のラジオボタンで選択肢の一覧を提供する選択コンポーネントの一種です。コンポーネントが提供する選択肢は、「選択」プロパティで開発者が定義します。

プロパティ名	説明	注
選択	ユーザーが選択可能な一群の選択肢。各選択肢は値とラベルを提供する。	
初期値	インスタンス化したときに可視コンポーネントに初期設定する値。バインドモデルフィールドに値が存在する場合、この初期値によって上書きされることに注意。モデルフィールドの値を上書きすることなくコンポーネントに値を設定する場合は、このプロパティを使わずに、`overwrite` パラメータに `false` を設定した `setValue(Object value,boolean overwrite)` メソッドを使用する。このメソッドは必要に応じてコード (イベントハンドラなど) から呼び出したり、同じコンポーネントの初期化後コードプロパティから呼び出したりできる。	
モデルフィールドバインド	可視コンポーネントの値の格納/検出場所となるバインド先のモデルフィールド。このプロパティを設定するには、事前に「モデル参照」プロパティを設定しておく必要がある。	モデル参照に依存
モデル参照	可視コンポーネントのバインドモデルフィールドが属するモデルへの参照。「モデルフィールドバインド」プロパティを設定するには、事前にこのプロパティを設定しておく必要がある。	
名前	コンポーネントインスタンスの名前	必須
Null 選択ラベル	フィールドの値が null の場合にコンポーネントの選択肢として表示するテキスト。このプロパティを空 (null) のままにすると、ユーザーに null 選択肢が提供されず、ユーザーは必ず一覧から選択肢のうちの 1 つを選択する必要がある。	
表示	コンポーネントを表示するかどうかを制御する。コンポーネントの `setVisible(boolean)` メソッドを使い、プログラムから設定することもできる。	

第19章

静的テキストフィールド

静的テキストフィールドコンポーネントは、読み取り専用のテキストまたはマークアップを表示します。このコンポーネントを使用して、ユーザーが見えるテキスト(ページ上のラベルなど) を表示したり、マークアップやその他不可視のテキスト内容を生成したりできます。

プロパティ名	説明	注
初期値	インスタンス化したときに可視コンポーネントに初期設定する値。バインドモデルフィールドに値が存在する場合、この初期値によって上書きされることに注意。モデルフィールドの値を上書きすることなくコンポーネントに値を設定する場合は、このプロパティを使わずに、`overwrite` パラメータに `false` を設定した `setValue(Object value,boolean overwrite)` メソッドを使用する。このメソッドは必要に応じてコード (イベントハンドラなど) から呼び出したり、同じコンポーネントの初期化後コードプロパティから呼び出したりできる。	
モデルフィールドバインド	可視コンポーネントの値の格納/検出場所となるバインド先のモデルフィールド。このプロパティを設定するには、事前に「モデル参照」プロパティを設定しておく必要がある。	モデル参照に依存
モデル参照	可視コンポーネントのバインドモデルフィールドが属するモデルへの参照。「モデルフィールドバインド」プロパティを設定するには、事前にこのプロパティを設定しておく必要がある。	
名前	コンポーネントインスタンスの名前	必須
表示	コンポーネントを表示するかどうかを制御する。コンポーネントの `setVisible(boolean)` メソッドを使い、プログラムから設定することもできる。	

第20章

テキストフィールド

テキストフィールドは、自由形式の単一行テキスト入力フィールドです。

プロパティ名	説明	注
初期値	インスタンス化したときに可視コンポーネントに初期設定する値。バインドモデルフィールドに値が存在する場合、この初期値によって上書きされることに注意。モデルフィールドの値を上書きすることなくコンポーネントに値を設定する場合は、このプロパティを使わずに、overwrite パラメータに false を設定した setValue(Object value,boolean overwrite) メソッドを使用する。このメソッドは必要に応じてコード (イベントハンドラなど) から呼び出したり、同じコンポーネントの初期化後コードプロパティから呼び出したりできる。	
モデルフィールドバインド	可視コンポーネントの値の格納/検出場所となるバインド先のモデルフィールド。このプロパティを設定するには、事前に「モデル参照」プロパティを設定しておく必要がある。	モデル参照に依存
モデル参照	可視コンポーネントのバインドモデルフィールドが属するモデルへの参照。「モデルフィールドバインド」プロパティを設定するには、事前にこのプロパティを設定しておく必要がある。	
名前	コンポーネントインスタンスの名前	必須
表示	コンポーネントを表示するかどうかを制御する。コンポーネントの setVisible(boolean) メソッドを使い、プログラムから設定することもできる。	

第21章

テキスト領域

テキスト領域コンポーネントは、自由形式の複数行テキスト入力フィールドです。

プロパティ名	説明	注
初期値	インスタンス化したときに可視コンポーネントに初期設定する値。バインドモデルフィールドに値が存在する場合、この初期値によって上書きされることに注意。モデルフィールドの値を上書きすることなくコンポーネントに値を設定する場合は、このプロパティを使わずに、overwrite パラメータに false を設定した setValue(Object value,boolean overwrite) メソッドを使用する。このメソッドは必要に応じてコード (イベントハンドラなど) から呼び出したり、同じコンポーネントの初期化後コードプロパティから呼び出したりできる。	
モデルフィールドバインド	可視コンポーネントの値の格納/検出場所となるバインド先のモデルフィールド。このプロパティを設定するには、事前に「モデル参照」プロパティを設定しておく必要がある。	モデル参照に依存
モデル参照	可視コンポーネントのバインドモデルフィールドが属するモデルへの参照。「モデルフィールドバインド」プロパティを設定するには、事前にこのプロパティを設定しておく必要がある。	必須
名前	コンポーネントインスタンスの名前	
表示	コンポーネントを表示するかどうかを制御する。コンポーネントの setVisible(boolean) メソッドを使い、プログラムから設定することもできる。	

第 22 章

検査テキストフィールド

検査テキストフィールドコンポーネントは、関連付けられている妥当性検査コンポーネントでユーザー提供のテキストの妥当性を検査する機能を持つ、自由形式の単一行テキスト入力フィールドです。

プロパティ名	説明	注
初期値	インスタンス化したときに可視コンポーネントに初期設定する値。バインドモデルフィールドに値が存在する場合、この初期値によって上書きされることに注意。モデルフィールドの値を上書きすることなくコンポーネントに値を設定する場合は、このプロパティを使わずに、overwrite パラメータに false を設定した setValue(Object value,boolean overwrite) メソッドを使用する。このメソッドは必要に応じてコード (イベントハンドラなど) から呼び出したり、同じコンポーネントの初期化後コードプロパティから呼び出したりできる。	
モデルフィールドバインド	可視コンポーネントの値の格納/検出場所となるバインド先のモデルフィールド。このプロパティを設定するには、事前に「モデル参照」プロパティを設定しておく必要がある。	モデル参照に依存
モデル参照	可視コンポーネントのバインドモデルフィールドが属するモデルへの参照。「モデルフィールドバインド」プロパティを設定するには、事前にこのプロパティを設定しておく必要がある。	
名前	コンポーネントインスタンスの名前	必須
検査障害メッセージを表示	このコンポーネントに対する検査が失敗した場合に、「検査障害メッセージ」の値を表示するかどうかを定義する。	

プロパティ名	説明	注
検査障害メッセージ	オプションで、フィールドに対する検査が失敗した場合にユーザー向けに表示するメッセージテキスト。この値には、マークアップを含めることができる。	
妥当性検査	このフィールドに関連付けられている妥当性検査コンポーネント	
表示	コンポーネントを表示するかどうかを制御する。コンポーネントの `setVisible(boolean)` メソッドを使い、プログラムから設定することもできる。	

第23章

検査テキスト領域

検査テキスト領域コンポーネントは、関連付けられている妥当性検査コンポーネントでユーザー提供のテキストの妥当性を検査する機能を持つ、自由形式の複数行テキスト入力フィールドです。

プロパティ名	説明	注
初期値	インスタンス化したときに可視コンポーネントに初期設定する値。バインドモデルフィールドに値が存在する場合、この初期値によって上書きされることに注意。モデルフィールドの値を上書きすることなくコンポーネントに値を設定する場合は、このプロパティを使わずに、overwrite パラメータに false を設定した setValue(Object value,boolean overwrite) メソッドを使用する。このメソッドは必要に応じてコード (イベントハンドラなど) から呼び出したり、同じコンポーネントの初期化後コードプロパティから呼び出したりできる。	
モデルフィールドバインド	可視コンポーネントの値の格納/検出場所となるバインド先のモデルフィールド。このプロパティを設定するには、事前に「モデル参照」プロパティを設定しておく必要がある。	モデル参照に依存
モデル参照	可視コンポーネントのバインドモデルフィールドが属するモデルへの参照。「モデルフィールドバインド」プロパティを設定するには、事前にこのプロパティを設定しておく必要がある。	
名前	コンポーネントインスタンスの名前	必須
検査障害メッセージを表示	このコンポーネントに対する検査が失敗した場合に、「検査障害メッセージ」の値を表示するかどうかを定義する。	
検査障害メッセージ	オプションで、フィールドに対する検査が失敗した場合にユーザー向けに表示するメッセージテキスト。この値には、マークアップを含めることができる。	
妥当性検査	このフィールドに関連付けられている妥当性検査コンポーネント	
表示	コンポーネントを表示するかどうかを制御する。コンポーネントの setVisible(boolean) メソッドを使い、プログラムから設定することもできる。	

第24章

マスク付きテキストフィールド

マスク付きテキストフィールドは、自由形式の単一行テキスト入力フィールドです。このコンポーネントによって、特定のマスク表現との照合による入力の検証のサポートが追加されます。

マスク表現は次のように指定されます。

- # 文字は数字 (0 ～ 9) を表わす
- 文字 A は大文字を表わす
- 文字 a (小文字の a) は小文字を表わす
- 文字 B は文字 (大文字または小文字) を表わす
- 文字 . は任意の文字を表わす
- 文字 * は任意の文字または数字を表わす
- その他の文字 (\ を除く) はそのまま使用される
- エスケープが必要な文字 (a、#、A、\ のみ) の前には文字 \ を付ける

たとえば、マスク表現「###-##-####」は社会保障番号の入力の検証に使用できます。マスク表現「\##B#\#」は、先頭と末尾が「#」で、その間は 2 つの数字に挟まれた 1 つの文字となる入力を示しています (例「#9k1#」)。

注 – このコンポーネント内で使用される Java および JavaScript のため、エスケープ文字として使用されるバックスラッシュ文字「\」には、もう 1 つエスケープが必要です。たとえば、入力マスク「\#aA*」を表わすには、コンポーネントに「\\\\ #aA*」と入力する必要があります。

入力マスク表現を使用しない場合、このコンポーネントは標準のテキストフィールドと同様の動作になります。

プロパティ名	説明	注
初期値	インスタンス化したときに可視コンポーネントに初期設定する値。バインドモデルフィールドに値が存在する場合、この初期値によって上書きされることに注意。モデルフィールドの値を上書きすることなくコンポーネントに値を設定する場合は、このプロパティを使わずに、overwrite パラメータに false を設定した setValue(Object value,boolean overwrite) メソッドを使用する。このメソッドは必要に応じてコード (イベントハンドラなど) から呼び出したり、同じコンポーネントの初期化後コードプロパティから呼び出したりできる。	
モデルフィールドバインド	可視コンポーネントの値の格納/検出場所となるバインド先のモデルフィールド。このプロパティを設定するには、事前に「モデル参照」プロパティを設定しておく必要がある。	モデル参照に依存
モデル参照	可視コンポーネントのバインドモデルフィールドが属するモデルへの参照。「モデルフィールドバインド」プロパティを設定するには、事前にこのプロパティを設定しておく必要がある。	
名前	コンポーネントインスタンスの名前	必須
表示	コンポーネントを表示するかどうかを制御する。コンポーネントの setVisible(boolean) メソッドを使い、プログラムから設定することもできる。	
マスク表現	このコンポーネントが検証で照合するマスク付き表現値	

第25章

日付表示

日付表示コンポーネントは、コンボボックスで表わされたフィールド (月、日、年) の日付を表示する可視表示フィールドコンポーネントです。JavaScript のミニカレンダポップアップも含まれています。

日付表示コンポーネントはコンポーネントパレットから使用でき、他のコンポーネントと同様にコンテナビューに追加されます。日付表示コンポーネントは、モデルフィールドにバインドされた値の日付部分を表示します。JavaScript ミニカレンダポップアップは、「ミニカレンダを表示」プロパティでこのコンポーネントと同時に構成できます。さらに 2 つの構成プロパティ (年選択の最大表示値、年選択の最小表示値) によって、年フィールドコンボボックスのサイズ制限を設定できます。

プロパティ名	説明	注
モデル参照	可視コンポーネントのバインドモデルフィールドが属するモデルへの参照。「モデルフィールドバインド」プロパティを設定するには、事前にこのプロパティを設定しておく必要がある。	必須
モデルフィールドバインド	可視コンポーネントの値の格納/検出場所となるバインド先のモデルフィールド。このプロパティを設定するには、事前に「モデル参照」プロパティを設定しておく必要がある。	モデル参照に依存
名前	コンポーネントインスタンスの名前	
年選択の最大表示値	年フィールドコンボボックスに表示する最大の年を指定する	デフォルトは <現在の年> + 50
年選択の最小表示値	年フィールドコンボボックスに表示する最小の年を指定する	デフォルトは <現在の年> -50
ミニカレンダを表示	ミニカレンダポップアップを起動するボタンを表示するかどうかを指定する	デフォルトは False

第 26 章

時刻表示

時刻表示コンポーネントは、コンボボックスで表わされたフィールド (時、分など) の時刻を表示する可視表示フィールドコンポーネントです。

時刻表示コンポーネントはコンポーネントパレットから使用でき、他のコンポーネントと同様にコンテナビューに追加されます。時刻表示コンポーネントは、モデルフィールドにバインドされた値の時刻部分を表示します。軍用時刻表示形式では、2 つのコンボボックスが使用されて、時間フィールドと分フィールドが表現されます。標準の時刻表示形式では、3 つのコンボボックスが使用されて、時間、分、午前/午後の各フィールドが表現されます。

プロパティ名	説明	注
モデル参照	可視コンポーネントのバインドモデルフィールドが属するモデルへの参照。「モデルフィールドバインド」プロパティを設定するには、事前にこのプロパティを設定しておく必要がある。	必須
モデルフィールドバインド	可視コンポーネントの値の格納/検出場所となるバインド先のモデルフィールド。このプロパティを設定するには、事前に「モデル参照」プロパティを設定しておく必要がある。	モデル参照に依存
名前	コンポーネントインスタンスの名前	
表示	コンポーネントを表示するかどうかを制御する。コンポーネントの `setVisible(boolean)` メソッドを使い、プログラムから設定することもできる。	
分間隔表示	分表示の間隔を 1、5、10、15、または 30 分間で指定する。	デフォルトは 1
時刻書式	表示形式を軍用時刻 (24 時間制) と標準時刻 (12 時間制) のどちらにするかを指定する。	デフォルトは 12 時間制

第27章

日時表示

日時表示コンポーネントは、コンボボックスで表わされたフィールド (月、日、年、時、分など) の日付を表示する可視表示フィールドコンポーネントです。JavaScript のミニカレンダポップアップも含まれています。

日時表示コンポーネントはコンポーネントパレットから使用でき、他のコンポーネントと同様にコンテナビューに追加されます。日時表示コンポーネントは、モデルフィールドにバインドされた値の日時部分を表示します。JavaScript ミニカレンダポップアップは、「ミニカレンダを表示」プロパティでこのコンポーネントと同時に構成できます。さらに 2 つの構成プロパティ (年選択の最大表示値、年選択の最小表示値) によって、年フィールドコンボボックスのサイズ制限を設定できます。軍用時刻表示形式では、2 つのコンボボックスが使用され、時間フィールドと分フィールドが表現されます。標準の時刻表示形式では、3 つのコンボボックスが使用されて、時間、分、午前/午後の各フィールドが表現されます。

プロパティ名	説明	注
モデル参照	可視コンポーネントのバインドモデルフィールドが属するモデルへの参照。「モデルフィールドバインド」プロパティを設定するには、事前にこのプロパティを設定しておく必要がある。	必須
モデルフィールドバインド	可視コンポーネントの値の格納/検出場所となるバインド先のモデルフィールド。このプロパティを設定するには、事前に「モデル参照」プロパティを設定しておく必要がある。	モデル参照に依存
名前	コンポーネントインスタンスの名前	
年選択の最大表示値	年フィールドコンボボックスに表示する最大の年を指定する	デフォルトは <現在の年> + 50
年選択の最小表示値	年フィールドコンボボックスに表示する最小の年を指定する	デフォルトは <現在の年> -50
ミニカレンダを表示	ミニカレンダポップアップを起動するボタンを表示するかどうかを指定する。	デフォルトは False

プロパティ名	説明	注
分間隔表示	分表示の間隔を 1、5、10、15、または 30 分間で指定する。	デフォルトは 1
時刻書式	表示形式を軍用時刻 (24 時間制) と標準時刻 (12 時間制) のどちらにするかを指定する。	デフォルトは 12 時間制

第 28 章

ページ移動 (リンク)

ページ移動 (リンク) コンポーネントは、Goto View Bean コマンド記述子を使用するよう事前に構成された、HREF 表示フィールドコンポーネントです。

ページ移動 (リンク) コンポーネントは、ハイパーリンク (HREF) コンポーネントと大体同じように動作します。唯一の違いは、ページ移動 (リンク) コンポーネントは Goto View Bean コマンド記述子に明示的に事前定義されているという点です。この記述子は、このコンポーネントの「ターゲット ViewBean クラス名」プロパティからそのパラメータを受け取ります。

プロパティ名	説明	注
モデル参照	可視コンポーネントのバインドモデルフィールドが属するモデルへの参照。「モデルフィールドバインド」プロパティを設定するには、事前にこのプロパティを設定しておく必要がある。	必須
モデルフィールドバインド	可視コンポーネントの値の格納/検出場所となるバインド先のモデルフィールド。このプロパティを設定するには、事前に「モデル参照」プロパティを設定しておく必要がある。	モデル参照に依存
名前	コンポーネントインスタンスの名前	
表示	コンポーネントを表示するかどうかを制御する。コンポーネントの `setVisible(boolean)` メソッドを使い、プログラムから設定することもできる。	
ターゲット ViewBean クラス名	要求の終わりに表示するビュー Bean のクラス名	必須

第29章

メニュー

メニューは、ユーザーが Web アプリケーションにドロップダウンメニュー機能を設計するためのビューコンポーネントです。メニューコンポーネントは、メニューバーとサブメニュー項目から構成されます。メニューコンポーネントはコンポーネントパレットに用意され、設計時にスタイルと表示を構成できます。

メニュー項目を XML で定義すると、メニュー項目を階層化された集合として構成できます。実行時に、この XML はメニュー定義を読み込むための入力として使用されます。

この XML の作成方法についての詳細は、DTD とサンプル XML を参照してください。

プロパティ名	説明	注
背景色	メニューの背景色を定義する。すべてのメニュー項目は、このプロパティで指定された色で表示される。	
マウスカーソルが重なったときの色	マウスカーソルが重なったときのメニューの色を定義する。マウスカーソルがメニュー項目に重なると、このプロパティで指定された色に変化する。	
フォント名	メニュー項目のテキストに使用するフォントファミリを定義する。メニュー項目のすべてのテキストは、このプロパティで指定されたフォントで表示される。	
フォントサイズ	メニュー項目のテキストに使用するフォントサイズを定義する。メニュー項目のすべてのテキストは、このプロパティで指定されたフォントサイズで表示される。	
フォントスタイル	メニュー項目のテキストに使用するフォントスタイルを定義する。メニュー項目のすべてのテキストは、このプロパティで指定されたフォントスタイルで表示される。	

プロパティ名	説明	注
メニュースタイル	メニューのスタイルを定義する (水平と垂直のどちらか)	
メニュー定義ファイル	メニュー定義の格納された XML ファイルの名前を定義する。XML ファイルの位置は、アプリケーションのコンテキストルートから相対的な位置にする必要がある。	必須
キャッシュ付きメニュー定義の属性名	XML DOM ツリーが保存されるアプリケーションスコープの属性名を定義する。これを設定すると、このプロパティで指定された属性に格納された DOM ツリーからメニュー定義が読み込まれる。	

第30章

静的パンくずリスト

パンくずリストナビゲーションは、サイト構造内での現在のページのコンテキストを表示します。また、情報をグループ化されている方法をユーザーが理解しやすくなり、グループ間を移動したり、情報の構造を理解したりすることも可能になります。静的パンくずリストはパンくずリストの一種で、現在のページに対して表示されているパンくずリストが、ユーザーのナビゲーションに依存しないで、サイト開発者が指定したページの論理的グループ化に依存します。

静的パンくずリストはビューコンポーネントで、静的パンくずリストコンポーネントをユーザーが自分の Web アプリケーションにドラッグアンドドロップできます。Web アプリケーションは、パンくずリスト項目の階層化集合にマッピングでき、XML でそれを構成します。実行時に、この XML は再とマップ構造を読み取るための入力として使用され、現在表示されているページに依存して、パンくずリストが構造化されます。

この XML の作成方法についての詳細は、DTD とサンプル XML を参照してください。

プロパティ名	説明	注
接頭辞テキスト	ブレッドクラブの接頭辞として表現されるテキストを定義する。 たとえば、「接頭辞テキスト」プロパティが「現在の場所」の場合、ブラウザには 現在の場所： ホーム > ショッピング > 男性用品 と表現される。	
区切り画像	ブレッドクラブ項目間で使用される区切り画像を定義する。相対 URL と絶対 URL のどちらも指定できる。相対 URL の場合、画像ファイルの位置はアプリケーションのコンテキストルートから相対的な位置にする必要がある。	区切り画像が指定されない場合は、区切り文字列が使用される。
区切り文字列	ブレッドクラブ項目間で使用される区切り文字列を定義する。 たとえば、「区切り文字列」プロパティが「>>」の場合、ブラウザには 現在の場所：ホーム >> ショッピング >> 男性用品 と表現される。	このプロパティに指定した文字列は、ブラウザで画像を表現できない場合の代替区切りとしても使用される。
サイトマップファイル	サイトマップ構造の格納された XML ファイルの名前を定義する。XML ファイルの位置は、アプリケーションのコンテキストルートから相対的な位置にする必要がある。	
キャッシュ付きサイトマップの属性名	XML DOM ツリーが保存されるアプリケーションスコープの属性名を定義する。これを設定すると、このプロパティで指定された属性に格納された DOM ツリーからサイトマップ定義が読み込まれる。	

第31章

データセットナビゲータ

データセットナビゲータコンポーネントは、基本 ViewBean、基本タイルビュー、基本コンテナビューと同様に、コンテナビューから一連のデータのページネーション (ナビゲーション) コントロールを可能にする 4 つのコマンドフィールドのセットです。4 つのコマンドフィールドはそれぞれ、「先頭」、「前」、「次」、「最後」という 4 つの種類のナビゲーションを表しています。コマンドフィールドは、操作しても表示が変化しない場合に自動的に非表示、または自動的に無効になるよう構成できます。たとえば、現在表示されているのが最後のフィールドの場合は、「次」のレコードが存在しないため、「次」と「最後」のコマンドフィールドを非表示または無効にします。

プロパティ名	説明	注
自動的に無効	対応する操作をしても表示されるデータが変化しない場合に、コマンドフィールドを自動的に非表示/無効にできる。	
無効時に非表示	無効モードのコマンドフィールドを表示しない	自動的に無効に依存
名前	コンポーネントのクラス名	

プロパティ名	説明	注
ターゲットコンテナビューのパス	親 (ContainerView、TiledView、ViewBean のどれか) に宣言されている ContainerView 参照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設定できる。パス内のすべてのコンポーネントは、ContainerView (TiledView、ContainerView、ViewBean、あるいはこれらのタイプのカスタムバージョンまたは他社バージョンの派生物) を参照。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で始まる場合は、名前はルートビュー (ViewBean) への相対パスと見なされる。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**）	

第 32 章

データセットロケータ

データセットロケータコンポーネントは、表示されたデータの「レコード」を表示します (たとえば、レコード 1 〜 10 / 53)。

「〜 ##」が表示されるのは、ページで同時に複数のレコードが表示されている場合だけです。

「/ ##」が表示されるのは、計算可能な場合だけです。一部のモデルでは `PaginatingModel` インタフェース、特に `getTotalDataCount` メソッドが完全には実装されていないため、この情報を表示できません。要求のたびにデータセット全体を読み込むモデルでは、`getTotalDataCount` メソッドが実装されていなくても、常に合計を表示できます。

ページにレコードが表示されない場合、データセットロケータはデフォルトで「(データは見つかりませんでした)」と表示します。このメッセージは、JSP ページレット「com/sun/jatox/view/DatasetLocator.jsp」を編集することでカスタマイズできます。このようになるのは、モデルで `getTotalDataCount` メソッドが実装されず、1 ページに表示されるレコード数で実際の合計データ数が均等に割り切れる場合です。合計レコード数が 50 で、1 ページに 10 レコード表示される場合の最後のページには、「レコード 41 〜 50」と表示され、合計データ数が不明なために「次」ボタンと「最後」ボタンが有効になります。ユーザーがどちらかのボタンをクリックすると、「(データは見つかりませんでした)」とだけ表示されたページが表示され、「次」ボタンと「最後」ボタンが無効になります。

ページ上のデータセットロケータコンポーネントの物理的レイアウトによって、ページが機能するかどうかが決まることがあります。これは、内容が生成された順序と、アプリケーション内での実際のモデルの実行順序の問題です。たとえば、実際にデータを表示するフィールドよりも前にデータセットロケータを配置した場合、モデルがまだ実行されていない可能性があります。実行されていない場合は、データ表示の後にデータセットロケータを移動するか、データセットロケータを表示する前にモデルを手動で実行します。

プロパティ名	説明	注
データセット合計を表示	データセット内のレコードの絶対的合計数 (最終レコードのインデックス番号) を表示する。これが機能するのは、`PaginatingModel` インタフェースから `getTotalDataSize` メソッドが正しく実装されたモデル、またはレコードの合計数を計算で判定できるモデルのみ。レコードの合計数を判定できない場合、このコントロールの合計レコード数部分は、プロパティが False に設定された場合と同様に表示されない。	
名前	コンポーネントのクラス名	
ターゲットコンテナビューのパス	親 (ContainerView、TiledView、ViewBean のどれか) に宣言されている ContainerView 参照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設定できる。パス内のすべてのコンポーネントは、ContainerView (TiledView、ContainerView、ViewBean、あるいはこれらのタイプのカスタムバージョンまたはサードパーティバージョンの派生物) を参照。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で始まる場合は、名前はルートビュー (ViewBean) への相対パスと見なされる。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 : / （**このコンポーネントの親でもある、同じコンテナビューを対象とする**） `/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**）	

第 33 章

Bean アダプタモデル

Bean アダプタモデルは、モデル用のバッキングデータストアとして 1 つ以上の JavaBean を使用できるようにします。このモデルによって表示フィールドを JavaBean プロパティにバインドすることができます。アプリケーションオブジェクトモデルがあり、ビューへのオブジェクトの自動バインドを利用する場合に便利な手法では、EJB クライアントライブラリとの統合に理想的な方法です。クライアント EJB インタフェースを設計する際の一般的で推奨される手法は、EJB ビジネスメソッドの入力、出力、および戻り値パラメータがプリミティブか JavaBean、あるいは同一のもののまとまりである場合に転送オブジェクトパターンを使用する方法です。

プロパティ名	説明	注
Bean クラス	モデルが「適用」する (あるいはモデルのもととなる) JavaBean の完全限定クラス名	
Bean スコープ	適用される Bean の J2EE スコープか場所で、選択肢は「要求」、「セッション」、「アプリケーション」、「なし」、「任意」のいずれか。このモデルは Bean を作成するわけではないことに注意。スコープが何であれ、Bean が存在することは、ベースクラス実装は関知しない。開発者は「なし」を選択し、適用される Bean をプログラムで割り当てることができる。	
Bean スコープ属性名	スコープが要求、セッション、アプリケーションのいずれかの場合の属性名。たとえば Bean のスコープがセッションの場合、このプロパティには HTTP セッション属性の名前を設定する。	
名前	コンポーネントのクラス名	必須

Bean アダプタモデルのデザインアクション

「フィールドを更新」

Bean クラスプロパティの自動的な検査と Bean クラスの JavaBean プロパティごとのモデルフィールドの作成を行うことができます。この機構は JavaBean イントロスペクションを使用して、プロパティを特定します。イントロスペクションでは BeanInfo をキャッシュすることができるため、通常、この設計アクションを繰り返し起動すると、同じ組み合わせの JavaBean プロパティが生成されます。適用される Bean クラスそのものが変更された場合は、その Bean クラスの Studio ファイルシステムを再マウントして、Bean のイントロスペクションで最新のプロパティ記述子が取得されるようにする必要があります。

フィールド

プロパティ名	説明	注
Bean プロパティ名	このフィールド用の JavaBean プロパティの名前。Bean アダプタモデルのモデルフィールドはそれぞれ 1 つの JavaBean プロパティを表す。モデルフィールド名プロパティに基づいて JavaBean プロパティ名も表す場合は、このプロパティを null (無設定) に設定できる。このプロパティ値を設定する場合は、JavaBean プロパティの記述子名と完全に一致する必要がある。	
名前	モデルフィールドの論理名	必須

第 34 章

カスタムモデル

カスタムモデル拡張可能コンポーネントは、完全に新しく任意のモデル実装を作成する必要がある開発者をサポートします。新しいモデル実装を手動でコーディングすることは可能でしたが、カスタムモデルを使用すると、新しいモデルとそのフィールド、オペレーションを IDE で自動的に表示させることができます。カスタムモデルは、モデルデータに既存のインフラストラクチャをほとんど提供することがなく、記憶領域も提供しません。このモデルは、モデルインタフェースを最小限実装して最初から新しいモデルをコーディングした経験がある開発者に役立ちます。

プロパティ名	説明	注
デフォルトのオペレーション名	`execute()` メソッドの ModelExecutionContext パラメータにオペレーション名が指定されていない場合に呼び出すオペレーション。新しいモデルの `execute()` メソッド実装は、開発者が提供する必要があることに注意。新しいモデルに対する `execute()` の動作を無効にするか、省略した場合、このプロパティは意味を持たない。つまり、このプロパティを使用するかどうかは、コンポーネントの作成者の任意である。	
名前	コンポーネントのクラス名	必須

フィールド

プロパティ名	説明	注
フィールドクラス	モデル実装をコーディングする開発者が使用するフィールドのクラス型 (String、Integer、Boolean など)	
名前	モデルフィールドの論理名	必須

オペレーション

プロパティ名	説明	注
名前	モデルオペレーション名	必須

第35章

単純カスタムモデル

単純カスタムモデルは、高度なデータセット管理およびページングサポートを必要とする新しいモデルに対する土台を提供します。単純カスタムモデルは、以前に com.iplanet.jato.model.DefaultModel を特殊化した新しいモデルをコーディングしたことがある場合に役立ちます。他のタイプのカスタムモデルと異なり、この種のカスタムモデルはフィールド値記憶やその他の動作を提供します。開発者は既存の機能を定義するのではなく、カスタマイズするだけでよいことになります。

プロパティ名	説明	注
値の型を強制	true の場合、モデル実装はモデルフィールドに設定された値をそのフィールドに指定された型に変換しようとする。たとえばフィールドの型が Boolean で、HTML フォーム上で表示フィールドを使って String 値がモデルにマップされている場合は、その String 値を強制的に Boolean 型に変換する。この機能は、あらゆる型の変換に com.iplanet.jato.util.TypeConverter クラスを使用するため、開発者は新しい型変換アルゴリズムを登録することができる。	
デフォルトのオペレーション名	`execute()` メソッドの ModelExecutionContext パラメータにオペレーション名が指定されていない場合に呼び出すオペレーション。新しいモデルの `execute()` メソッド実装は、開発者が提供する必要があることに注意。新しいモデルに対する `execute()` の動作を無効にするか、省略した場合、このプロパティは意味を持たない。つまり、このプロパティを使用するかどうかは、コンポーネントの作成者の任意である。	
名前	コンポーネントのクラス名	必須

フィールド

プロパティ名	説明	注
フィールドクラス	モデル実装をコーディングする開発者が使用するフィールドのクラス型 (String、Integer、Boolean など)	
名前	モデルフィールドの論理名	必須

オペレーション

プロパティ名	説明	注
名前	モデルオペレーション名	必須

第 36 章

カスタムツリーモデル

カスタムツリーモデルは、XML ドキュメントや LDAP リポジトリ、ファイルシステムなどの、ツリーやディレクトリなどの階層データ構造を利用するデータストアを利用することを可能にします。

プロパティ名	説明	注
名前	コンポーネントのクラス名	必須

第 37 章

HTTP セッションモデル

HTTP セッションモデルは、そのバッキングデータストアとして HTTP セッションを使用します。HTTP セッションモデルのフィールドは、直接 HTTP セッションの属性に対応します。他の大部分のモデルと異なり、HTTP セッションモデルのフィールドは単に値をセッションの属性にそのまま渡す手段です。言い替えれば、フィールドに値を設定すると、モデルを実行して実際のバッキングデータストアを更新しなくても、その値が直ちにセッションの属性にプッシュされます。このモデルには、セッションの属性に対する統一インタフェースである内部記憶領域はありません。

Web アプリケーションフレームワークアプリケーション内には、HTTP セッションとの対話に HTTP セッションモデルは必要ありません。HTTP セッションの API は、あらゆる Web アプリケーションと同じく完全利用できます。このモデルは、アプリケーションの設計中に、バインドで使用し、内部で HttpSession へのアダプタまたはインターセプタとしてセッション属性を正式に宣言する手段を提供します。

プロパティ名	説明	注
等しい値の設定を許可	アプリケーションサーバーによっては、`HttpSession.setAttribute()` を呼び出すことによって、結果的に `HttpSessionBindingListeners` が起動されたり、持続セッションストアでトランザクションが発生したりなどの影響が出ることがある。このプロパティは、冗長な `setValue()` 呼び出しが `HttpSession.setAttribute()` にそのまま渡されるのを防ぐ手段を提供する。	
名前	コンポーネントのクラス名	必須

フィールド

プロパティ名	説明	注
属性クラス	HTTP セッション属性のクラスの型。setValue() の HTTP セッションモデル実装では、このプロパティを使用して、値の型を強制する。	
属性名	モデルフィールドが対応する HTTP セッション属性の名前。モデルフィールド名プロパティに基づいて HTTP セッション属性名も表す場合は、このプロパティを null (無設定) に設定できる。このプロパティ値を設定する場合は、セッション属性に予定されている名前と完全に一致する必要がある。	
名前	モデルフィールドの論理名	必須
任意の初期値	初期値をフィールド宣言でカプセル化することができる。このプロパティに基いてセッション属性を初期化するには、ModelManager からこのモデルを取得し、`ensureInitialValues()` メソッドを起動し、HttpSession セッションにこの初期値をプッシュすることを推奨する。この呼び出しは、アプリケーションサーブレットの `onNewSession()` メソッドから行うのが適切。	
一時的な属性として格納	`true` に設定した場合、値を管理する JavaBean 内の実際の値は、このフィールドに対する `setValue()` の呼び出しによって一時的なメンバーとしてラップされる。ここでの値は、アプリケーションサーバーが HttpSession のパッシベーションまたは持続記憶をサポートしている場合に、このプロパティを使用して、属性値のメモリーキャッシュへの格納をサポートすることができる。たとえば、ユーザーセッション全体にわたって必要とされる業務データが大量にある場合、開発者はその大量のデータ構造をディスクまたは持続ストアにプッシュするのを回避することができる。このプロパティは高度であり、有効にするときには注意する必要がある。HTTP セッションに対して高可用性サポートを提供するアプリケーションの場合、このプロパティを有効にすると、このセッション属性が高可用でなくなる。	

第 38 章

JDBC SQL 照会モデル

JDBC SQL 照会モデルは、モデル用のバッキングデータストアとして 1 つ以上の RDBMS を使用することを可能にします。この機能によって、表示フィールドをデータベース表内の列にバインドすることができます。SQL オペレーション (表の結合を含む選択、挿入、更新、削除) はすべて、照会モデルを使って行うことができます。

プロパティ名	説明	注
データソース	J2EE コンテナからの接続の取得に使用する JDBC データソース名	必須
照会表を変更	変更を行う照会 (挿入、更新、削除) を生成する際に使用する表の名前。モデルは選択を行う照会に複数の表を使用できるが、変更を行う照会では 1 つしか使用できない。	
名前	コンポーネントのクラス名	必須
SQL テンプレートを選択	検出オペレーションの際の SQL SELECT 文の作成に使用する SQL 選択文のテンプレート。一般には、この文の末尾に where トークン ("__WHERE__") があり、このトークンが、実行時にコンポーネントによって動的に作成される where 句で置き換えられる。	
静的 Where 条件	あらゆる SQL オペレーション (挿入を除く) で使用する where 句	

フィールド

プロパティ名	説明	注
列名	モデルフィールドが対応する、表内の列の実際の名前	必須
解析フィールド	モデルフィールドが解析フィールド (SQL の集計機能) に対応する場合に `true`。デフォルトの設定は `false`。	
空の式	フィールドに値がない場合に値を供給する SQL 式を指定する。ただし、この指定は、「空の値のポリシー」プロパティが「式を使用」に設定されている場合にのみ有効。	

プロパティ名	説明	注
空の値のポリシー	更新オペレーション中のフィールドへの値の供給に使用するポリシー。選択肢は除外、null を送信、式を使用のいずれか。デフォルトの設定は除外。	
フィールドの型	モデルフィールドの Java クラスの型 (java.lang.String、java.lang.Integer、java.lang.Boolean など)	必須
挿入式	フィールドに値がない場合に値を供給する SQL 式を指定する。ただし、この指定は、挿入値のソースプロパティが式を使用に設定されている場合にのみ有効。	
挿入値のソース	挿入オペレーション中のフィールドへの値の供給に使用するポリシー。選択肢は「アプリケーション」、「データベース」、「式を使用」のいずれか。デフォルトの設定はアプリケーション。	
キーフィールド	モデルの対応する列がキーフィールドであるかどうかを指定する。更新、挿入、あるいは削除オペレーションにこのモデルが使用される場合、このプロパティは必須である。ウィザードを使って照会モデルを作成すると、必ずしも、JDBC ドライバメタデータ内でキーフィールドを見つけられないことがある。たとえば、Oracle データソースが非常に矛盾なく動作しているのに、しばしば PointBase データソースがキーフィールドであることを示すものを明らかにできないことがある。このフィールドが設定されていない状態で更新、挿入、削除オペレーションが起動されると、SQL 例外になることがある。	
名前	モデルフィールドの論理名	必須
限定列名	モデルフィールドが対応している列の完全限定名 (<*table*>.<*column*>)	必須
サポートされるオペレーション	モデルフィールドが関与する SQL オペレーション (選択、挿入、更新、削除)。デフォルトの設定は「選択, 挿入, 更新, 削除」。	

第 39 章

JDBC ストアドプロシージャモデル

JDBC ストアドプロシージャモデルでは、ストアドプロシージャを実行することができます。このモデルでは、ストアドプロシージャ内のパラメータと結果列 (ベンダーがサポートしている場合) に表示フィールドをバインドすることができます。

プロパティ名	説明	注
データソース	J2EE コンテナからの接続の取得に使用する JDBC データソース名	必須
名前	コンポーネントのクラス名	必須
プロシージャ名	このモデルが呼び出す RDBMS 内のストアドプロシージャの名前	必須

結果セットの列のフィールド

プロパティ名	説明	注
列名	モデルフィールドが対応するストアドプロシージャ内の結果列の名前。	必須
フィールドの型	モデルフィールドの Java クラスの型 (java.lang.String、java.lang.Integer、java.lang.Boolean など)	必須
名前	モデルフィールドの論理名	必須

プロシージャのパラメータのフィールド

プロパティ名	説明	注
パラメータクラス	モデルフィールドの Java クラスの型 (java.lang.String、java.lang.Integer、java.lang.Boolean など) これは、データベース内の実際のパラメータの SQL 型でないことに注意。	必須
パラメータ名	このフィールドのバインド先の、ストアドプロシージャ内のパラメータの名前	必須
パラメータ型	ストアドプロシージャのパラメータの型 (IN、IN_OUT、OUT、RESULT、RETURN、UNKNOWN)	必須
名前	モデルフィールドの論理名	必須
SQL 型	パラメータの SQL 型 (`java.sql.Types` にある型で、VARCHAR、TIMESTAMP、SMALLINT など)	必須

第40章

オブジェクトアダプタモデル

オブジェクトアダプタモデルは、オブジェクトメンバーへの「深い」アクセスを指定したパス式を使用する任意のオブジェクトまたはそこに含まれるオブジェクトのフィールド、Bean プロパティ、メソッドへのアクセス手段を提供します。

モデルにオブジェクトクラス名プロパティを設定した後で、そのクラスをコンパイルして、読み込み可能にすると、ブラウズ、あるいはビュー上の表示フィールドへのバインドで一般的なキーパスバインド選択ダイアログが利用できるようになります。ビューは匿名のパス式にバインドすることができますが、開発者は、複雑なパス式の別名として機能する名前付きのモデルフィールドを作成し、モデルのクライアントからのオブジェクトグラフの変更を特定するのに役立てることもできます。こうしたフィールドの作成は、モデルフィールドを追加し、そのプロパティを設定した後で手動で行うことも、あるいは「オブジェクトフィールドバインドをブラウズ/追加」アクションを起動することによって自動的に行うことも可能です。

このコンポーネントは `com.iplanet.jato.model.ExecutingModel` を実装し、IDE 内でモデルオペレーションを宣言することができます。そうしたモデルオペレーションは、適用されたオブジェクトクラスの最上位のメソッドにマップされ、それらメソッドのみモデルオペレーションとして表示することができます。パス式はオブジェクトグラフ内の深いメソッドを起動できますが、現在の実装でサポートされているのは、パラメータがゼロ個か、文字列表現のパラメータを持つメソッドだけです。モデルオペレーションは、新しいモデルオペレーションを追加し、プロパティシートでオペレーション名とパラメータを編集することによって手動で作成することができます。また、オペレーションは、コンテキストメニュー項目の「不完全なオペレーションを補完」を使用して、自動的に追加することもできます。

プロパティ名	説明	注
デフォルトデータセット名	適用されたオブジェクトに複数のデータセット (コレクション) が存在することがあるため、このプロパティはそのうちのどのデータセットをデフォルトと見なすかを宣言する。フィールドがバインドされるとき、含まれるデータセットに達するすべてのパス式は、現在のデータセット名をモデルに設定されている必要がある。このプロパティによって、現在のデータセット名が null の場合にモデルが使用するデータセット名 (含まれるデータセットを示すパス式) を指定できる。	
オブジェクト配列	true の場合は、適用されるオブジェクト型がアレイまたはコレクションであることを示す。デフォルトの設定は false。	
名前	コンポーネントのクラス名	必須
オブジェクトクラス名	適用されるオブジェクト型の完全限定クラス名	必須
オブジェクトファクトリ	開発者が ObjectFactory インタフェースを実装している JavaBean を指定することを可能にする。適用されるオブジェクトがプログラムで設定されていなくて、代わりにオブジェクトファクトリが指定されている場合、モデルはオブジェクトファクトリに委託して、適用されるオブジェクトを実行時に検出する。標準のオブジェクトファクトリによって、標準の J2EE 要求やセッション、あるいいはアプリケーションスコープからオブジェクトを検出できる。	

オブジェクトアダプタモデルのデザインアクション

「不完全なオペレーションを補完」

このアクションを起動すると、適用されたオブジェクト上の最上位の public メソッドを表すモデルオペレーションが少なくとも 1 組存在するようになります。モデルオペレーションのパラメータと戻り値については、記憶領域機構について説明した詳細 JavaDoc を参照してください。

「オブジェクトフィールドバインドをブラウズ/追加」

このアクションを起動すると、バインド選択ダイアログが開き、オブジェクトのプロパティとオペレーションを検査することができます (プロパティはオブジェクトが JavaBean の場合のみ、オペレーションは最上位のオブジェクトメソッドにオペレーションが定義されている場合のみ)。ダイアログには、オブジェクトグラフ上で現在選択されているノードのキーパス式が表示されます。オブジェクトグラフ上の、データセットを表すノードのパス式は、データセット名で表されて強調表示されます。「了解」を選択すると、グラフ上で現在選択されているノードに新しいモデルフィールドが生成されます。「取消し」を選択すると、フィールドを追加することなくダイ

アログが終了します。開発者は、現在のパス式を強調表示させて、クリップボードにコピーすることもできます。この機能は、「デフォルトデータセット名」やタイルビューの「一次データセット名」を設定する際に便利です。

フィールド

プロパティ名	説明	注
キーパス式	適用されたオブジェクトグラフ上のプロパティやメンバー、メソッドを通って、フィールドの値にアクセスする方法を表す式。この式は、実行時に、論理フィールド名を解決して、オブジェクトグラフ内の物理的なフィールド値をするために使用される。キーパス式の構文についての詳細は、JavaDoc を参照。	
名前	モデルフィールドの論理名	必須

オペレーション

プロパティ名	説明	注
名前	モデルオペレーション名	必須
オペレーション名	適用されたオブジェクトクラス上の public メソッドの名前	必須
オペレーションパラメータ	このオペレーションのメソッドのゼロ個以上のパラメータを記述する。	

第41章

リソースバンドルモデル

リソースバンドルモデルは、開発者がリソースバンドルを使用してローカライズした値を検出することを可能にします。このモデルで使用されるモデルフィールド名は、そうしたリソースの名前です。リソースバンドルモデルの一般的な用途の 1 つは、静的なテキスト表示フィールドを含むローカライズした内容を形成し、表示フィールドをリソースバンドルモデルにバインドすることによってページをローカライズすることです。実行時のリソース検出にモデルが使用するロケールは、プログラムで設定することができます。設定されていない場合は、デフォルトのシステムロケールが使用されます。

このモデルコンポーネントには、リソースバンドルが設定されている場合に使用可能なすべてのリソースを表示するモデルフィールド選択機能があります。また、リソース名をカプセル化した事前定義のモデルフィールドもサポートしています。このため、バンドルでリソース名が変更されても、モデルのクライアントがリソース名の変更の影響を受けることはありません。モデルフィールドの可能な用途としては、もう 1 つ、モデルフィールドにバインドされたクライアントに影響を与えることなく、実行時に動的にフィールドのリソース名を変更するという使い方があります。

プロパティ名	説明	注
バンドル名	バンドルの完全限定リソース名 (例 : com/sun/jato/Bundle)	必須
名前	コンポーネントのクラス名	必須

フィールド

プロパティ名	説明	注
名前	モデルフィールドの論理名	必須
リソース名	このモデルに関連付けられているリソースバンドル内のキー。モデルフィールド名プロパティに基いてリソース名も表す場合は、このプロパティを null (無設定) に設定できる。このプロパティ値を設定する場合、この値はバンドルで検出されたリソース名に完全に一致する必要があります。	

第 42 章

Web サービスモデル

Web サービスモデル (WS モデル) は、開発者が Web サービスオペレーションを簡単に実行したり、表示フィールドバインドを使って、それらオペレーションのパラメータを検出、生成することを可能にします。Web サービスモデルは、特に JAX-RPC クライアントスタブを処理する特殊なオブジェクトアダプタモデルです。結果は、任意の RPC スタイルの Web サービスに適用されるモデルになります。

Web サービスモデルは、ウィザードによってその作成時に完全に構成されます。Web サービスポート上のすべてのメソッドについて、必要なパラメータがすべて設定され、モデルオペレーションが宣言されます。通常、この後の構成作業は、必要な場合にデフォルトのデータセット名を設定することだけです。

モデルが作成されると、アプリケーションの `WEB-INF/classes/stubs` ディレクトリに JAX-RPC クライアントのスタブが生成され、JAX-RPC サポートライブラリがアプリケーションの `WEB-INF/lib` ディレクトリに自動的に追加されます。

プロパティ名	説明	注
デフォルトデータセット名	適用されたオブジェクトに複数のデータセット (コレクション) が存在することがあるため、このプロパティはそのうちのどのデータセットをデフォルトと見なすかを宣言する。フィールドがバインドされるとき、含まれるデータセットに達するすべてのパス式は、現在のデータセット名をモデルに設定されている必要がある。このプロパティによって、現在のデータセット名が null の場合にモデルが使用するデータセット名 (含まれるデータセットを示すパス式) を指定できる。	
JAX RPC スタブファクトリ	オブジェクトアダプタモデルでは、任意のオブジェクトファクトリ Bean を指定できるのに対し、このモデルでは、`com.iplanet.jato.model.object.JaxRpcStubFactory` 型のオブジェクトファクトリが必要になる。この場合、オブジェクトファクトリは Web サービスモデルが要求スコープの JAX-RPC スタブを検出するのを支援する。このプロパティは Web サービスモデルウィザードによって自動的に設定され、一般には、編集しないことを推奨する。	
名前	コンポーネントのクラス名	必須

フィールド

プロパティ名	説明	注
キーパス式	適用されたオブジェクトグラフ上のプロパティやメンバー、メソッドを通って、フィールドの値にアクセスする方法を表す式。この式は、実行時に、論理フィールド名を解決して、オブジェクトグラフ内の物理的なフィールド値をするために使用される。キーパス式の構文についての詳細は、JavaDoc を参照。	
名前	モデルフィールドの論理名	必須

オペレーション

プロパティ名	説明	注
名前	モデルオペレーション名	必須
オペレーション名	適用されたオブジェクトクラス上の public メソッドの名前	必須
オペレーションパラメータ	このオペレーションのメソッドのゼロ個以上のパラメータを記述する。	

第 43 章

ディレクトリ検索モデル

ディレクトリ検索モデルでは、モデルを LDAP 照会の結果セットのバッキングストアとして使用できます。このモデルによって表示フィールドを Directory 属性にバインドすることができ、ディレクトリストアがあり、ビューへのディレクトリストア値の自動バインドを利用する場合に便利な手法になります。実行時の利点は、モデルが提供するページネーションサポートであることと、結果を双方向にスクロール可能なことです。

プロパティ名	説明	注
複数値の属性を付加	値は True と False のどちらか。1 つの属性に複数の値を追加する場合は True に設定する。	必須
複数値区切り文字	複数の値を追加する場合の区切り文字として使用される。上記のプロパティが False に設定されている場合は無視される。	省略可能
ルートコンテキスト	検索を実行するノード。値は「`ou=People,dc=sun,dc=com`」などとなる。	必須
検索フィルタ	検索用フィルタを指定する。標準のディレクトリ検索フィルタ書式にする必要がある。例：「`(cn=anand*)`」	必須

プログラム上で変更可能なプロパティもあります。そのような属性の例としては、SearchControl、InitialDirContext があります。このような属性の使用についての詳細は、Javadoc を参照してください。

ディレクトリ検索モデルのデザインアクション

最初に、モデルにフィールドを作成します。各フィールドに、LDAP 内の属性名と同じ名前を指定します。または、「属性名」を使用して属性名を保持し、「名前」は単なる論理名とすることもできます。

フィールド

プロパティ名	説明	注
属性名	属性名と一致している必要がある。	省略可能
名前	モデルフィールドの論理名	必須

このコンポーネントの使用方法

1. ディレクトリ検索モデルをアプリケーションへドラッグします。
2. 前述の表の属性を設定します。
3. モデルにフィールドを追加します。モデルフィールド名は、属性名と同じにする必要があります。
4. 複数の値の属性を連結して、1 つの値を生成できます。
5. LDAP サーバーの属性名と一致するよう、各フィールドのモデルフィールドプロパティ「属性」を変更する必要があります。
6. ページレット (タイトル表示) を作成し、それをディレクトリ検索モデルと関連付けます。
7. 必要なフィールドをページレットへドラッグします。
8. ページレットをビュー Bean へドラッグします。
9. ビュー Bean に送信ボタンとテキストフィールドを作成します。
10. ユーザーがテストフィールドに検索フィルタを入力した場合は、値を取得して、それを (ボタンの処理要求メソッドで) モデルに設定します。
11. これで、モデルが新しい結果セットで更新され、それが表示されます。

第 44 章

JDBC 結果セットアダプタモデル

JDBC 結果セットモデルは、渡される結果セットに適応します。ユーザーは、設計時にフィールド名と型を設定できます。アプリケーション開発者は、表示フィールドを直接、作成したこのモデルフィールドにマッピングできます。モデルは、基礎となる結果セットへのアダプタとして機能し、結果セットでカプセル化されたデータの取得と表示だけをサポートします。また、挿入および更新アクションをサポートしていません。

Bean アダプタモデルと同様に、このモデルは値のバインドの前に setResultSet() を呼び出すことでプログラム内で使用できます。また、コンポーネント情報内で定義された構成プロパティを持つことがあり、スコープオブジェクト内で自動的に結果セットを検索します。

プロパティ名	説明	注
結果セット属性スコープ	「要求」、「アプリケーション」、「セッション」を値としてとる。「すべて」を指定した場合、属性はすべての値をとる。	
結果セット属性名	結果セットが格納される属性の名前	上記を設定した場合は必須

JDBC 結果セットアダプタモデルの開発

プロパティ名	説明	注
フィールド名	接続しようとする表の列の名前	省略可能
名前	モデルフィールドの論理名	必須

このモデルの残りのフィールドは、すべて入力の必要がありません。

第 45 章

クライアントセッションモデル

クライアントセッションモデルは、そのバッキングデータストアとして JATO クライアントセッションを使用します。クライアントセッションモデルのフィールドは、直接クライアントセッションの属性に対応します。他の大部分のモデルと異なり、クライアントセッションモデルのフィールドは単に値をクライアントセッションの属性にそのまま渡す手段です。言い替えれば、フィールドに値を設定すると、モデルを実行して実際のバッキングデータストアを更新しなくても、その値が直ちにクライアントセッションの属性にプッシュされます。このモデルには、クライアントセッションの属性に対する統一インタフェースである内部記憶領域はありません。

プロパティ名	説明	注
名前	コンポーネントのクラス名	必須

フィールド

プロパティ名	説明	注
属性クラス	HTTP セッション属性のクラスの型。setValue() の HTTP セッションモデル実装では、このプロパティを使用し、値の型を強制する。	
属性名	モデルフィールドが対応する HTTP セッション属性の名前。モデルフィールド名プロパティに基づいて HTTP セッション属性名も表す場合は、このプロパティを null (無設定) に設定できる。このプロパティ値を設定する場合は、セッション属性に予定されている名前と完全に一致する必要がある。	
名前	モデルフィールドの論理名	必須

第46章

基本コマンド

基本コマンドコンポーネントは、コントローラまたは要求ハンドラコンポーネントです。このコンポーネントは、再利用可能、拡張可能な要求処理オブジェクトを作成するための単純な構造体です。

プロパティ名	説明	注
名前	コンポーネントのクラス名	必須

第 47 章

コマンド連鎖

コマンド連鎖コンポーネントでは、順に起動する複数のコマンドを 1 つにつなげることができます。

プロパティ名	説明	注
コマンド連鎖記述子	順に起動する一群のコマンドコンポーネント	
名前	コンポーネントのクラス名	必須

第48章

アプリケーション属性ファクトリ

アプリケーション属性ファクトリは、アプリケーションスコープからオブジェクトを取得するファクトリです。

プロパティ名	説明	注
属性名	アプリケーションスコープからのオブジェクトの検出に使用する属性の名前	
名前	コンポーネントインスタンスの名前	必須

第 49 章

実行モデルとページ移動コマンド

実行モデルとページ移動コマンド (実行コマンドと移動コマンド) は自動的にモデルを実行し、現在のアプリケーション内のページを表示します。

プロパティ名	説明	注
コマンドクラス名	要求を処理するコマンドクラスの名前。デフォルトでは、標準実装の com.iplanet.jato.view.command.ExecuteAndForwardCommand に設定される。この上級プロパティの設定は、標準コマンドクラス実装のサブクラスを設定する場合にだけ変更することを推奨する。	
実行モデル参照	指定されたオペレーションを実行するモデルの参照。	必須
名前	コンポーネントインスタンスの名前	必須
モデルオペレーション名	実行するモデルオペレーションの名前	
ターゲット ViewBean クラス名	モデルの実行後に表示するビュー Bean のクラス名	必須
ユーザーパラメータ	開発者が定義した一群のパラメータ。Java 式を含むあらゆる Java 型のパラメータを指定可能。このプロパティに指定されたパラメータは、CommandEvent パラメータを介して、指定されたコマンドコンポーネントの execute() メソッドに渡される。また、標準パラメータマップ内で単一の予約パラメータとして渡される。この予約パラメータは com.iplanet.jato.view.command.ViewCommandDescriptorBase.PARAM_USER_PARAMETERS として型付けられ、その型は java.util.Map になる。 この上級プロパティが意味を持つのは、関係する上級プロパティの「コマンドクラス名」にもデフォルト以外の値が設定されていて、「コマンドクラス名」プロパティに指定されたデフォルト以外のクラスが、予約パラメータキーの com.iplanet.jato.view.command.ViewCommandDescriptorBase.PARAM_USER_PARAMETERS を検索するようにコーディングされている場合だけである。	

第50章

モデル実行コマンド

モデル実行コマンドは、起動されると自動的にモデルを実行します。

プロパティ名	説明	注
コマンドクラス名	要求を処理するコマンドクラスの名前。デフォルトでは、標準実装の com.iplanet.jato.view.command.ExecuteModelCommand に設定される。この上級プロパティの設定は、標準コマンドクラス実装のサブクラスを設定する場合にだけ変更することを推奨する。	
実行モデル参照	指定されたオペレーションを実行するモデルの参照	必須
名前	コンポーネントインスタンスの名前	必須
モデルオペレーション名	実行するモデルオペレーションの名前	
ユーザーパラメータ	開発者が定義した一群のパラメータ。Java 式を含むあらゆる Java 型のパラメータを指定可能。このプロパティに指定されたパラメータは、CommandEvent パラメータを介して、指定されたコマンドコンポーネントの execute() メソッドに渡される。また、標準パラメータマップ内で単一の予約パラメータとして渡される。この予約パラメータは com.iplanet.jato.view.command.ViewCommandDescriptorBase.PARAM_USER_PARAMETERS として型付けられ、その型は java.util.Map になる。 この上級プロパティが意味を持つのは、関係する上級プロパティの「コマンドクラス名」にもデフォルト以外の値が設定されていて、「コマンドクラス名」プロパティに指定されたデフォルト以外のクラスが、予約パラメータキーの com.iplanet.jato.view.command.ViewCommandDescriptorBase.PARAM_USER_PARAMETERS を検索するようにコーディングされている場合だけである。	

第51章

転送コマンド

転送コマンドは RequestDispatcher サーブレットを使用して、現在のアプリケーション内のリソースに進みます。

プロパティ名	説明	注
コマンドクラス名	要求を処理するコマンドクラスの名前。デフォルトでは、標準実装の com.iplanet.jato.view.command.ForwardCommand に設定される。この上級プロパティの設定は、標準コマンドクラス実装のサブクラスを設定する場合にだけ変更することを推奨する。	
名前	コンポーネントインスタンスの名前	必須
パス	現在のアプリケーション内のリソースへのパス。一般には、このパスはサーブレットか JSP、HTML、その他ファイルのいずれか。	必須
ユーザーパラメータ	開発者が定義した一群のパラメータ。Java 式を含むあらゆる Java 型のパラメータを指定可能。このプロパティに指定されたパラメータは、CommandEvent パラメータを介して、指定されたコマンドコンポーネントの execute() メソッドに渡される。また、標準パラメータマップ内で単一の予約パラメータとして渡される。この予約パラメータは com.iplanet.jato.view.command.ViewCommandDescriptorBase.PARAM_USER_PARAMETERS として型付けられ、その型は java.util.Map になる。 この上級プロパティが意味を持つのは、関係する上級プロパティの「コマンドクラス名」にもデフォルト以外の値が設定されていて、「コマンドクラス名」プロパティに指定されたデフォルト以外のクラスが、予約パラメータキーの com.iplanet.jato.view.command.ViewCommandDescriptorBase.PARAM_USER_PARAMETERS を検索するようにコーディングされている場合だけである。	

第 52 章

ページ移動コマンド

ページ移動コマンドは、起動されるとページコンポーネントを表示します。

プロパティ名	説明	注
コマンドクラス名	要求を処理するコマンドクラスの名前。デフォルトでは、標準実装の com.iplanet.jato.view.command.GotoViewBeanCommand に設定される。この上級プロパティの設定は、標準コマンドクラス実装のサブクラスを設定する場合にだけ変更することを推奨する。	
名前	コンポーネントインスタンスの名前	必須
ターゲット ViewBean クラス名	要求の終わりに表示するビュー Bean のクラス名	必須
ユーザーパラメータ	開発者が定義した一群のパラメータ。Java 式を含むあらゆる Java 型のパラメータを指定可能。このプロパティに指定されたパラメータは、CommandEvent パラメータを介して、指定されたコマンドコンポーネントの execute() メソッドに渡される。また、標準パラメータマップ内で単一の予約パラメータとして渡される。この予約パラメータは com.iplanet.jato.view.command.ViewCommandDescriptorBase.PARAM_USER_PARAMETERS として型付けられ、その型は java.util.Map になる。 この上級プロパティが意味を持つのは、関係する上級プロパティのコマンドクラス名にもデフォルト以外の値が設定されていて、コマンドクラス名プロパティに指定されたデフォルト以外のクラスが、予約パラメータキーの com.iplanet.jato.view.command.ViewCommandDescriptorBase.PARAM_USER_PARAMETERS を検索するようにコーディングされている場合だけである。	

第53章

取り込みコマンド

取り込みコマンドは RequestDispatcher サーブレットを使用して、現在のアプリケーション内のリソースの取り込みを行います。

プロパティ名	説明	注
コマンドクラス名	要求を処理するコマンドクラスの名前。デフォルトでは、標準実装の com.iplanet.jato.view.command.IncludeCommand に設定される。この上級プロパティの設定は、標準コマンドクラス実装のサブクラスを設定する場合にだけ変更することを推奨する。	
名前	コンポーネントインスタンスの名前	必須
パス	現在のアプリケーション内のリソースへのパス。一般には、このパスはサーブレットか JSP、HTML、その他ファイルのいずれか。	必須
ユーザーパラメータ	開発者が定義した一群のパラメータ。Java 式を含むあらゆる Java 型のパラメータを指定可能。このプロパティに指定されたパラメータは、CommandEvent パラメータを介して、指定されたコマンドコンポーネントの execute() メソッドに渡される。また、標準パラメータマップ内で単一の予約パラメータとして渡される。この予約パラメータは com.iplanet.jato.view.command.ViewCommandDescriptorBase.PARAM_USER_PARAMETERS として型付けられ、その型は java.util.Map になる。 この上級プロパティが意味を持つのは、関係する上級プロパティの「コマンドクラス名」にもデフォルト以外の値が設定されていて、「コマンドクラス名」プロパティに指定されたデフォルト以外のクラスが、予約パラメータキーの com.iplanet.jato.view.command.ViewCommandDescriptorBase.PARAM_USER_PARAMETERS を検索するようにコーディングされている場合だけである。	

第54章

リダイレクトコマンド

リダイレクトコマンドは、HTTP 302 リダイレクト応答を使用して現在の要求を内部または外部の URL にリダイレクトします。

プロパティ名	説明	注
コマンドクラス名	要求を処理するコマンドクラスの名前。デフォルトでは、標準実装の com.iplanet.jato.view.command.RedirectCommand に設定される。この上級プロパティの設定は、標準コマンドクラス実装のサブクラスを設定する場合にだけ変更することを推奨する。	
名前	コンポーネントインスタンスの名前	必須
ユーザーパラメータ	開発者が定義した一群のパラメータ。Java 式を含むあらゆる Java 型のパラメータを指定可能。このプロパティに指定されたパラメータは、CommandEvent パラメータを介して、指定されたコマンドコンポーネントの execute() メソッドに渡される。また、標準パラメータマップ内で単一の予約パラメータとして渡される。この予約パラメータは com.iplanet.jato.view.command.ViewCommandDescriptorBase.PARAM_USER_PARAMETERS として型付けられ、その型は java.util.Map になる。 この上級プロパティが意味を持つのは、関係する上級プロパティの「コマンドクラス名」にもデフォルト以外の値が設定されていて、「コマンドクラス名」プロパティに指定されたデフォルト以外のクラスが、予約パラメータキーの com.iplanet.jato.view.command.ViewCommandDescriptorBase.PARAM_USER_PARAMETERS を検索するようにコーディングされている場合だけである。	
URL	要求のリダイレクト先の URL	必須

第55章

正規表現妥当性検査

正規表現妥当性検査は、JDK 1.4 正規表現を使用して値の妥当性を検査するコンポーネントです。

プロパティ名	説明	注
名前	コンポーネントインスタンスの名前	必須
検査規則	妥当性検査に使用する JDK 1.4 正規表現。データを検査する前に、データは、`com.iplanet.jato.util.TypeConverter` クラスを使って文字列に変換される。	必須

第56章

要求属性ファクトリ

要求属性ファクトリは、要求スコープからオブジェクトを取得するファクトリです。

プロパティ名	説明	注
属性名	要求スコープからのオブジェクトの検出に使用する属性の名前	
名前	コンポーネントインスタンスの名前	必須

第57章

セッション属性ファクトリ

セッション属性ファクトリは、セッションスコープからオブジェクトを取得するファクトリです。

プロパティ名	説明	注
属性名	セッションスコープからのオブジェクトの検出に使用する属性の名前	
名前	コンポーネントインスタンスの名前	必須

第58章

単純選択

単純選択の実装です。

プロパティ名	説明	注
ラベル	ユーザー向けに表示する選択肢のラベル	必須
名前	コンポーネントインスタンスの名前	必須
値	選択肢の値	必須

第59章

モデル参照

モデル参照は、com.iplanet.jato.model.SimpleModelReference のインスタンスを構成します。

プロパティ名	説明	注
インスタンス名	この要求内のモデルインスタンスの名前。インスタンス名が省略された場合は、デフォルトのインスタンスが使用される。同じインスタンス名 (デフォルト名を含む) を指定したあらゆる参照は、同じモデルインスタンスを共有する。	
セッションを検索	HTTP セッションからモデルを取得するかどうかを指定する。この値が true で、セッションからモデルを得られない場合は、新しいモデルインスタンスが作成される。また、「セッションに格納」プロパティが true に設定されている場合、そのインスタンスはセッションに格納される。	
モデルクラス名	モデルクラスの完全限定名	必須
名前	コンポーネントインスタンスの名前	必須
セッションに格納	モデルの新しいインスタンスが作成された場合に、モデルを HTTP セッションに格納するかどうかを指定する (「セッションを検索」プロパティが、true に設定されている場合、新しいインスタンスは作成されない)。モデルは、指定されたインスタンス名かデフォルトのインスタンス名でセッションに格納される。	

第60章

型妥当性検査

型妥当性検査は、指定された型への変換の成否に基づいて妥当性を検査します。

プロパティ名	説明	注
名前	コンポーネントインスタンスの名前	必須
検査規則	検査に使用する完全限定クラス名。検査中、提供された値は、`com.iplanet.jato.util.TypeConverter` クラスを使って、この型に変換される。この型変換が失敗した場合、検査は失敗する。	必須

第61章

ユーザー定義コマンド

ユーザー定義コマンドは、現在のアプリケーションまたはそのコンポーネントライブラリ内の任意コマンドコンポーネントへの参照を表します。

プロパティ名	説明	注
コマンドクラス名	コマンドクラスの完全限定名	必須
名前	コンポーネントインスタンスの名前	必須
オペレーション名	CommandEvent パラメータを介して、指定されたコマンドコンポーネントの execute() メソッドに渡されるオペレーションの名前。	
パラメータ	開発者が定義した一群のパラメータ。Java 式を含むあらゆる Java 型のパラメータを指定可能。このプロパティに指定されたパラメータは、CommandEvent パラメータを介して、指定されたコマンドコンポーネントの execute() メソッドに渡される。	

第 62 章

WebAction コマンド

WebAction コマンドは、指定された WebActionHandler コンポーネントに対して WebAction を起動します。

プロパティ名	説明	注
コマンドクラス名	要求を処理するコマンドクラスの名前。デフォルトでは、標準実装の `com.iplanet.jato.view.command.WebActionCommand` に設定される。この上級プロパティの設定は、標準コマンドクラス実装のサブクラスを設定する場合にだけ変更することを推奨する。	
名前	コンポーネントインスタンスの名前	必須
オペレーション名	実行する WebAction	必須
WebActionHandler パス	起動する WebActionHandler 可視コンポーネントを示す限定パス。このパスは、このコマンドを起動したコマンドフィールドコンポーネントの親を基準に解決される。たとえば、このコマンドにコンテナビューコンポーネント内のボタンが関連付けられている場合、WebAction は親のコンテナビューに対して起動される。 このパスの構文は、区切り文字としてスラッシュ (/) を使用する標準のビュー名パス式の構文に従う。パス内の、最後のコンポーネント以外のすべてのコンポーネントは、コンテナビューまたはコンテナビューからの派生物 (タイルビューなど) 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、ページ (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。	

索引