

Taglibユーザーズガイド

Webエンジン Ver.3.2対応

リリース 1.1 版

初版:2003 年 4 月

改訂:2003 年 6 月



*Muratec Information
Systems.LTD.*

Taglib ユーザーズガイド

原本部品番号:W1D0101-02

原本名: TaglibUser's Guide,Release1. 1

原本著者: 長谷川 和彦

編集: 久田 雅子

Copyright © 2002, MURATEC INFORMATION SYSTEMS,LTD. All rights reserved.

Printed in Japan

制限付権利の説明

プログラム(ソフトウェアおよびドキュメントを含む)の使用、複製または開示は、ムラテック情報システムとの契約に記された制約条件に従うものとします。著作権、特許権およびその他の知的財産権に関する法律により保護されています。

当プログラムのリバース・エンジニアリング等は禁止されております。

このドキュメントの情報は、予告なしに変更されることがあります。ムラテック情報システムは本ドキュメントの無謬性を保証しません。

* ムラテック情報システムとは、ムラテック情報システム株式会社を指します。

危険な用途への使用について

ムラテック情報システム製品は、原子力、航空産業、大量輸送、医療あるいはその他の危険が伴うアプリケーションを用途として開発されておりません。ムラテック情報システム社製品を上述のようなアプリケーションに使用することについての安全確保は、顧客各位の責任と費用により行ってください。万一かかる用途での使用によりクレームや損害が発生いたしましても、ムラテック情報システムおよびその関連会社は一切責任を負いかねます。

このドキュメントに記載されているその他の会社名および製品名は、あくまでその製品および会社を識別する目的にのみ使用されており、それぞれの所有者の商標または登録商標です。

目次

第 I 部	Taglibの概要	1
第1章	Taglibの概要	2
1.	Taglibの概要.....	2
2.	Taglibの特徴.....	2
第2章	Taglib一覧.....	4
1.	拡張ファイル一覧.....	4
2.	Taglib名称一覧.....	7
3.	Taglib属性一覧.....	13
第 II 部	Taglibリファレンス	25
第3章	Taglibリファレンス	26
1.	mis:column	26
2.	mis:input.....	29
3.	mis:select	34
4.	mis:option.....	37
5.	mis:queryOption.....	40
6.	mis:orderBy	41
7.	mis:button	43
8.	mis:textarea.....	45
9.	mis:query	47
10.	mis:view.....	50
11.	mis:entry	53
12.	mis:columnSet.....	54
13.	mis:update.....	56
14.	mis:upload.....	57
15.	mis:link	58
16.	mis:hidden	61
17.	mis:debug.....	62
18.	mis>equals.....	63
19.	mis:message.....	64
20.	mis:writeTable	66
21.	mis:space.....	68
22.	mis:plsqlUpdate.....	70
23.	mis:mailSender	73
24.	mis:mailReceiver.....	74

25.	mis:writeCheck.....	75
26.	mis:appear.....	75
27.	mis:readTable.....	76
28.	mis:where.....	78
29.	mis:and.....	80
30.	mis:viewLink.....	82
31.	mis:columnCheck.....	84
32.	mis:userInfo.....	85
33.	mis:frame.....	86
34.	mis:text.....	88
35.	mis:help.....	89
36.	mis:entryQuery.....	90
37.	mis:requestCache.....	92
38.	mis:shell.....	92
39.	mis:errorMessage.....	93
40.	mis:forward.....	94
41.	mis:columnMarker.....	95
42.	mis:viewMarker.....	96
43.	mis:file.....	97
44.	mis:fileQuery.....	98
45.	mis:fileOption.....	101
46.	mis:fileWhere.....	101
47.	mis:img.....	102
48.	mis:sql.....	104
49.	mis:tbody.....	105
50.	mis:tfoot.....	105
51.	mis:thead.....	106
52.	mis:columnEditor.....	106
53.	mis:favoriteLink.....	108
54.	mis:writeChart.....	109
55.	mis:comment.....	110
56.	mis:statusBar.....	111
57.	mis:report.....	112
58.	mis:value.....	114
59.	mis:schedule.....	115
60.	mis:ldapQuery.....	116
61.	mis:regQuery.....	119
62.	mis:submit.....	121
63.	mis:responseHeader.....	123

64.	mis:bodyHidden.....	124
65.	commonforward.....	125
66.	backGamen.....	126
第 III 部 Taglib属性リファレンス.....		129
第4章	カラム属性.....	130
1.	表示種別.....	130
2.	入力種別.....	132
3.	データベースタイプ.....	133
第5章	ViewFormクラス.....	135
第6章	Queryクラス.....	137
第7章	Writer／Readerクラス.....	139
1.	DBTableWriter.....	139
2.	DBTableReader クラス.....	140
3.	ChartDBTableWriter クラス.....	140

はじめに

MISでは、Webアプリケーション用のカスタム・タグライブラリー(以下 Taglib)を作成しました。Webアプリケーションで開発するための共通機能をTaglibとして提供することで、開発効率と品質の向上を実現しています。

このマニュアルでは、Taglibの概要について説明し、また、各Taglibについても詳細に解説します。また、後半には、Taglibリファレンスをご用意しましたので、マニュアルとして、ご活用ください。

《対象読者》

このマニュアルは、次の作業を行うシステム開発者、システムデザイナー、テクニカル・ユーザー、システム設計者、およびシステム管理者の方を対象としています。

- JSPで画面開発するシステム開発者
- JSP画面をカスタマイズする画面デザイナーおよび、テクニカルユーザー
- Webアプリケーションを利用して開発する、システム設計者および、システム管理者

《本文の表記規則》

本文中には、特別な用語が一目でわかるように様々な表記規則が使用されています。次の表は、本文の表記規則を示しています。

規則	意味
太字	太字は、本文中に定義されている用語または用語集に含まれている用語、あるいはその両方を示します。この句を指定する場合は、索引構成表を作成します。
大文字	大文字は、システムにより指定される要素を示します。
小文字	小文字は、実行可能ファイル、ファイル名、ディレクトリ名およびサンプルのユーザー指定要素を示します。 注意: 一部のプログラム要素には、大文字と小文字の両方が使用されます。この場合は、記載されているとおりに入力してください。
イタリック	イタリックは、プレースフォルダまたは変数を示します。

《コード例の表記規則》

次の表は、コード例の記載上の表記規則を示しています。

規則	意味
[]	大カッコで囲まれている項目は、1 つ以上のオプション項目を示します。大カッコ自体は入力しないでください。
{ }	中カッコで囲まれている項目は、そのうちの 1 つのみが必要であることを示します。中カッコ自体は入力しないでください。
	縦線は、大カッコまたは中カッコ内の複数の選択肢を区切るために使用します。オプションのうち 1 つを入力します。縦線自体は入力しないでください。
… ⋮ ⋮	省略記号は、例に直接関係のないコード部分が省略されていることを示します。

《アイコン》

本文中には、特別な情報を知らせるために、次のアイコンが用意されています。



ヒント

提案や秘訣を示し、これらによって、時間の節約や手順の容易化などを実現できる場合があります。



警告

システムに致命的な影響を及ぼす可能性のあるアクションについて、注意が必要であることを示します。



コラム

関連する基礎知識や細かい技などを解説しています。

第 I 部

Taglibの概要

『私は、実はたわいのない夢を大切にすることから革新が生まれると思っている』
井深大（ソニー創業者）

Webアプリケーションで開発するための共通機能は画面の雛型にあらかじめ設定されており、各画面の機能を実装する場合にTaglibを使用します。（本書でとりあげるTaglibとは、Webアプリケーションにおけるカスタムタグ・ライブラリーを意味します。）ここでは、Taglibの概要について説明します。
構成は、次のとおりです。

第 1 章 Taglibの概要

Taglibの概要、また、使用するにあたっての基本的な使い方、標準的な制約を説明します。

第 2 章 Taglib一覧

拡張ファイル、Taglib名称、Taglib属性をそれぞれ一覧表形式で見やすく示します。
尚、各Taglibの詳細内容については、『第 II 部 Taglibリファレンス』を参照してください。

Web
Web
アプリケーション

第1章 Taglibの概要

この章では、Taglibの概要、また、使用するにあたっての基本的な使い方、標準的な制約を説明します。(本書でとりあげるTaglibとは、MIS で作成したWebアプリケーション用のカスタムタグ・ライブラリーを意味します。)

1. Taglibの概要

Webアプリケーションで開発するための共通機能は画面の雛型にあらかじめ設定されており、各画面の機能を実装する場合にTaglibを使用します。Webアプリケーションでは、JSPを利用して開発するにふさわしい非常に強力な独自タグを提供しています。特に、シンプルPDMにおけるWebアプリケーションフレームワークの一環としてその機能を業務ロジックに特化することにより、汎用タグでは得られない機能性を有しています。

2. Taglibの特徴

①簡易的な属性セット

request 情報を、{@XXXX} (XXXX は、request のパラメーターを取り出すキー) で属性にセットすることができます。

【従来】

```
<input name="PN" value="<%= request.getParameter("PN") %>" />
```

【TagLib】

```
<mis:input name="PN" value="{@PN}" />
```

②XHTML1.0 準拠

XHTML1.0準拠のタグ形式でタグが記述できるようにしました。これにより、通常のタグで、画面イメージを作成した後、<mis:xxxx > タグに自動変換したり、その逆を行うことが可能になります。

③XML形式のJSPファイル

JSPファイルそのものも、XML形式で作成します。JSPファイルの管理、検索、動的作成など、XMLの機能をJSPファイルの作成に使用することができます。以下にJSPファイルをXML形式で作成する場合の注意点を示します。

- 開始タグと終了タグが必ず存在します。
- 空要素の場合も、変換後は <input ... /> とタグを終了させます。
- 属性値は、引用符で囲います。
- 属性は必ず「属性="値"」の形式で変換します。
checked ⇒ checked="checked"
- 要素名と属性名は大文字、小文字が区別されます。

④国際化対応

画面で使用するラベル類については、リソースファイルで提供することにより、コーディング中に直接メッセージを書き込まないようにしました。これにより、各国言語への対応は、言語ごとのリソースファイルを用意するだけで、動的に切替えることができます。

msg 属性は、MessageResource よりその lang 属性で指定された言語のリソースに変換します。同様に lbl 属性は、LabelResource より変換します。

lang 属性は、無指定の場合は、session オブジェクトの "hybs_lang" の値をデフォルトで使用します。(これは、UserInfo クラスの lang を Login 時にセットします。) option タグでは、code 属性と lbls 属性が使用できます。

code 属性は、CodeResource に応じたメニューを作成できます。通常、データベースの選択オプションとして使用できます。

lbls 属性は、カンマで区切られた複数のラベルを分割して、LabelResource より取り込み、ロケールに応じたラベルを取り込んだ後に再び合成します。通常、データベースの Order By 句の様に複数のラベルを合成して利用したいときに、便利です。

⑤多彩なプルダウンメニュー (code 属性)

コンボボックスやリストボックスを作成する場合、メニューは option タグで作成します。Taglib には、豊富な Option タグの機能により、多彩なプルダウンメニューを実現しています。

- code 属性

option タグの code 属性に値をセットすることで、CodeResource に応じたメニューを作成できます。通常、データベースの選択オプションとして使用できます。code 属性とセットで、value 属性を登録すれば、メニューのデフォルト値を設定できます。

(ただし、リストボックスのように複数選ばれた状態にしておくことはできません。)

```
<mis:select name="HDIV" >
  <mis:option value="{@HDIV}" code="HDIV" />
</mis:select>
```

- msg 属性、lbl 属性

msg 属性 をキーにして、MessageResource よりロケールに応じたリソースを読み取ります。同様に、lbl 属性は、LabelResource よりロケールに応じたリソースを読み取ります。

```
<mis:select name="ORDER_BY" >
  <mis:option value="ITEM" lbl="ITEM">品番</mis:option>
  <mis:option value="HDIV,CSIG" lbls="HDIV,CSIG" selected="selected">
  <mis:option value="CSIG,HDIV" lbls="CSIG,HDIV">
</mis:select>
```

- queryOption タグ

option タグの仲間で、データベースからデータを検索して、option タグを作成するのが、この、queryOption タグです。基本的には、query と同様の使い方をします。これは、検索結果の 1 番目を value 属性に、それ以降を、スペースで連結して BODY 部書き込みます。

下記の例は、検索結果に、ITEM が2度現れていますが、間違いではありません。

```
<mis:select name="ITEM" >
  <mis:queryOption>SELECT ITEM,ITEM,HDIV,CSIG FROM OITMSTRU</mis:queryOption>
</mis:select>
```

第2章 Taglib一覧

この章では、拡張ファイル、Taglib名称、Taglib属性をそれぞれ一覧表形式で見やすく示します。尚、各Taglibの詳細内容については、『第II部 Taglibリファレンス』を参照してください。

1. 拡張ファイル一覧

①拡張ファイルとは

拡張ファイルとは、あらかじめ標準機能を1つのファイルにまとめており、単独または、インクルードして使用することができる、JSPファイルのことです。それぞれのファイルは機能別にディレクトリに保管されています。

- common ディレクトリ
Webアプリケーションフレームワーク共通で使用する接続ファイルです。このディレクトリ内では、ファイルを修正したり、追加しないようにしてください。
- custom ディレクトリ
各業務アプリケーションで独自に作成した共通接続ファイルを保管します。common ディレクトリのファイルを修正する場合は、custom にコピー後修正してください。
- debug ディレクトリ
デバッグ時に使用します。
- menu ディレクトリ
プルダウンメニュー等のJSPが複数保管されています。編集しないでください。

②拡張ファイル一覧

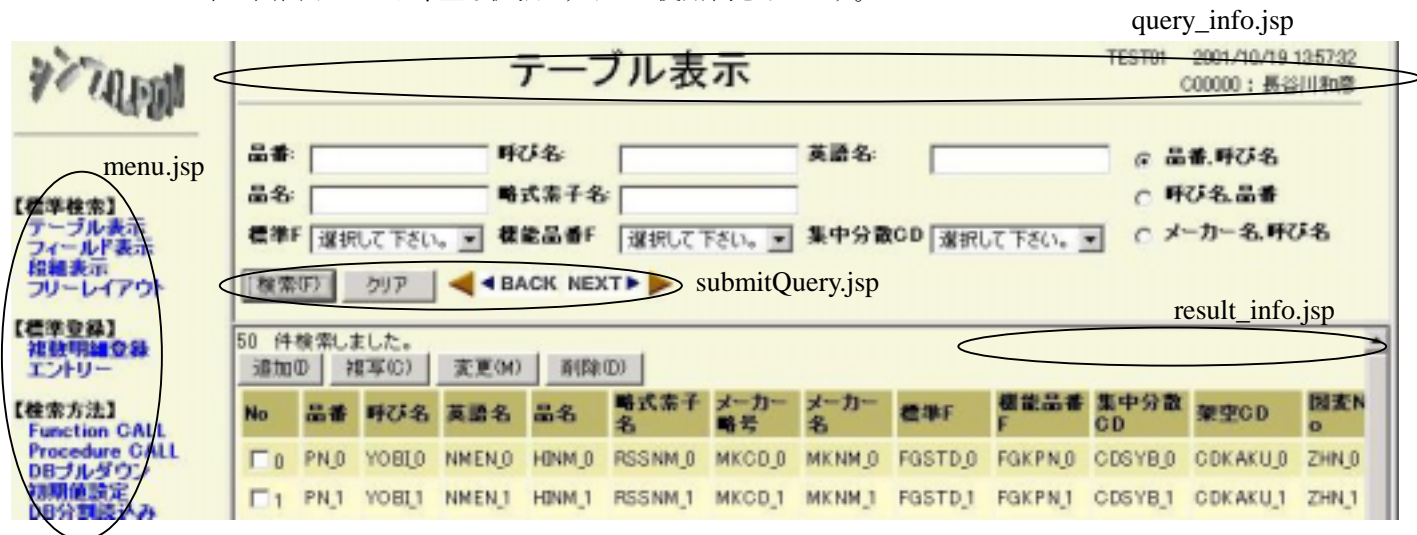
次に拡張ファイル一覧を示します。

ディレクトリ	名称	概要説明
common/	authenticate.jsp	独自の認証を行う場合に組み込みます。
	calendar.css	カレンダー表示を行うときの、画面のCSSファイルです。
	check.jsp	セッションの有無、ユーザーのアクセス許可をチェックします。何らかのエラーの場合は、error.jsp 画面へ転送します。
	default.css	全画面共通に使用されるデフォルトCSSファイルです。
	default.js	Query 画面に共通に使用されるJavaScriptです。先頭のフォームにフォーカスを移動させます。
	statusBar.js	ステータスバーに文字を表示させる為に利用する共通のJavaScriptです。
	treeBOM.js	階層メニュー形式でテーブル表示を行う場合に使用するJavaScriptです。
	encode.jsp	JSP画面のエンコードを指定します。JSP画面は、UTF-8に統一されました。(エンジンVer3より導入。)
	error.jsp	何らかのエラー時(チェックや異常処理等)に呼ばれます。
	errorMessage.jsp	ワーニング発生時に使用するポップアップ画面です。
	Excel_direct.jsp	タブ区切りファイルをテンポラリに出力したり、そこから呼び出したりする機能を集めたファイルです。サーバーのテンポラリディレクトリと直接やり取りできるクライアントからのみ使用できます。
	Excel_insert.jsp	サーバーのテンポラリディレクトリのタブ区切りファイルを追加モードで読み取ります。
	Excel_update.jsp	サーバーのテンポラリディレクトリのタブ区切りファイルを更新モードで読み取ります。
	HTML.jsp	HTML のヘッダーまでの情報を記述しています。各画面共通に組み込みます。
	htmlend.jsp	JSP画面のXMLとしてのタグの整合性と、XHTMLファイルとしての整合性を取る為、各画面の最後にインクルードします。
	jsp_init.jsp	JSP 画面を使用する場合に必ずインクルードして下さい。(通常は、共通拡張ファイル:例えば query_info.jsp 等にインクルードして使用します。)
	log.jsp	画面アクセスごとにログファイルに書き出します。
	prev_next.jsp	submitQuery.jsp / submitFieldQuery.jsp にインクルードされる 進む・戻るボタンです。
	session_init.jsp	ユーザーログイン時に一度だけ呼ばれて、ユーザー情報の設定を行います。
	start_jsp_init.jsp	ユーザーのログインチェックを行い、必要なオブジェクトを初期化します。
	submitClearQuery.jsp	検索開始ボタンで、クリアボタンで初期化します。
	submitFieldQuery.jsp	検索開始ボタンで、テキストフィールド表示時に使用します。
	submitQuery.jsp	通常の実行開始ボタンです。 検索(F)/クリア ボタンとネクストリンクを作成します。
	submitSelectQuery.jsp	検索開始ボタンです。ネクストリンクはありません。
submitSet.jsp	登録化医師ボタンです。ネクストリンクはありません。	

ディレクトリ	名称	概要説明
custom/	custom.css	アプリケーション固有にカスタマイズされたCSSファイルです。これは、自動的に default.css よりも優先して使用されます。
	init.jsp	ユーザーログイン時に初期設定を行う場合に使用します。Index.jsp等にインクルードして使用します。
	query_info.jsp	QUERY 画面(検索条件入力画面)に対して、画面名やユーザー情報などを表示します。
	result_info.jsp	RESULT 画面(検索条結果表示画面)に対して、書き込み許可が ON のときのみ、追加(I)/複写(C)/変更(M)/削除(D) ボタンを作成します。
	logout.html	ログアウトメニュー使用時にこのHTMLが使用されます。
	stopFile.html	フィルター機能で運用を中止した場合に、このHTMLが使用されます。
menu/	fw_menu.js	ヘッダーメニュー使用時のJavaScriptです。
	GmisTree.js	G-MIS折りたたみメニュー使用時のJavaScriptです。
	menu.jsp	標準的なメニュー画面を作成します。
	menu1.jsp	ツリー階層メニューを作成します。
	menu2.jsp	G-MIS折りたたみメニューを作成します。
	headermenu.jsp	ヘッダーメニューを作成します。
	headermenu2.jsp	ヘッダーメニューのスタートポイントファイルです。
	popupmenu.jsp	ポップアップメニューを作成します。
	dir.jsp	ディレクトリ階層をツリー階層で表示するメニューを作成します。
helpMenu.jsp	ヘルプメニューを作成します。	
debug/	printConfig.jsp	ServletConfig オブジェクト情報を表示します。
	printContext.jsp	ServletContext オブジェクト情報を表示します。
	printHeader.jsp	リクエストのヘッダー情報を表示します。
	printPageContext.jsp	PageContext オブジェクト情報を表示します。
	printRequesMethodt.jsp	request オブジェクトのメソッドでアクセスできる値を表示します。
	printRequest.jsp	ServletRequest のサブクラス情報を表示します。
	printSettion.jsp	HttpSession オブジェクト情報を表示します。

③画面イメージ

下の画面イメージは、主な拡張ファイルの使用例を示します。



2. Taglib名称一覧

Webアプリケーションには、多数のTaglibが用意されています。下表はその名称と概要一覧を示しています。通常のHTML Tagを元に独自機能を付加したTagについては、表の「HTML拡張」欄に○をつけています。尚、各Taglibの属性については、次項「3. Taglib属性一覧」、また詳細内容については、『第II部 Taglibリファレンス』を参照してください。また、各表のNOはそれぞれ対応しています。

①Taglib名称一覧

No	名称	概要説明	HTML 拡張
1	column	インプットフィールド(入力フィールド)を作成する簡易タグです。name属性をキーにして、DBCColumnResourceよりリソースを読み取ります。	○
2	input	ユーザーがデータの入力や選択、データの送信などを行うための部品(入力フィールドやボタン)を作成するタグです。name属性にLabelResource.propertiesのキーを与えることで、ロケールにあわせたリソースを使用して、画面に表示します。	○
3	select	プルダウンメニューを作成する簡易タグです。メニュー内の選択肢は、selectタグの内容であるOptionTag、またはQueryOptionTagによって指定されます。name属性にLabelResource.propertiesのキーを与えることで、ロケールにあわせたリソースを使用して、画面に表示します。	○

第2章 Taglib一覧

No	名称	概要説明	HTML 拡張
4	option	SelectTag の中で使用される項目選択用オプションを作成するタグです。プルダウンメニューの選択項目を作成します。name 属性は、LabelResource.properties のキーを与えることで、使用する上位のタグのロケールにあわせたリソースを使用して、画面に表示します。よって、このタグでは ロケールは指定できません。	○
5	queryOption	OptionTag の仲間で、データベースからデータを検索して、項目選択用オプションを作成します。基本的には、QueryTag と同様の使い方をします。プルダウンメニュー内の選択肢を SELECT 文の結果から作成します。このオブジェクトに、queryId を与えることにより、queryId に対応した Query オブジェクト(のサブクラスのオブジェクト)が作成されます。	×
6	orderby	ラジオボタンで表示順(Order By 句)を指定する簡易タグです。name 属性値は、“ORDER_BY” 固定です。	×
7	button	HTML の button タグと同等で、国際化対応とボタンの値を渡すことができます。これにより、ボタンごとに必要な画面を選び出して制御を転送することが可能です。	○
8	textarea	複数行の入力が可能なテキスト入力フィールドを作成するタグです。	○
9	query	データベースの検索を行うためのタグです。このタグの内容に SQL 文を記述します。	×
10	view	データベースの検索結果を表示します。設定によりテキストフィールドやテーブルの形で表示したり、入力可/不可、表示可/不可の設定を行うことが可能です。	×
11	entry	DBTableModel オブジェクトを操作する、共通オブジェクトです。	×
12	columnSet	入力カラムに対して初期値を設定します。DBTableModel オブジェクトを操作する、共通オブジェクトです。	×
13	update	データベースに追加/更新します。(推奨していません)使用しないでください。データベースに追加/更新するときは、PLSQL を CALL して下さい。	×
14	upload	MultipartRequest を利用して、ファイルをサーバーにアップロードします。	×
15	link	リンクを作成するタグです。HTML の a タグと同等で、国際化対応と引数の受け渡しが可能です。	○
16	hidden	隠しフィールドを作成するタグです。name 属性をキーに value 属性をセットします。	×
17	debug	デバッグ時にタグで囲まれた内容を画面に表示します。	×

No	名称	概要説明	HTML 拡張
18	equals	val1 属性 と val2 属性の文字列表現の比較により BODY を表示/ 非表示を切り替える簡易タグです。	×
19	message	画面にメッセージを表示させるタグです。msg 属性に MessageResource.properties のキーを与えることで、ロケールにあわ せたリソースを使用して、画面に表示します。	×
20	writeTable	データベースの検索結果をファイルに出力します。DBTableModel オ ブジェクトを表示する、共通オブジェクトです。	×
21	space	テーブル表示で InputTag 等の存在しない個所に挿入して、テーブ ルの行列の個数を合わせるために使用します。	×
22	plsqlUpdate	PLSQL を CALL してデータベースにアクセスします。DBTableModel 内 のデータを 配列で PL/SQL に渡して DB 登録します。	×
23	mailSender	メールを送信するための簡易タグです。	×
24	mailReceiver	メールを受信するための簡易タグです。	×
25	writeCheck	サブMITボタンの表示を制御します。ユーザー情報と画面情報を元 に、ロール/グループの権限と RW(読取/書込)モードを比較して、条件 が成立しない場合は、その BODY 部を表示しないように できます。	×
26	appear	指定された値が 設定されている場合のみ表示されるタグです。 SqlAndTag と類似の特性をもっています。逆にいえば、このタグの特殊 ケースが SqlAndTag だといえます。	×
27	readTable	DBTableModel オブジェクトを表示する、共通オブジェクトです。 このオブジェクトに、データ(DBTableModel)と、コントローラ (ReadTableTagForm)を与えて、外部からコントロールすることで、各種 形式で データ(DBTableModel)を表示させることが 可能です。 ReadTableTag に対して、コマンドを与えることにより、内部のコントロー ラの実装に対応した、形式でデータを作成します。	×
28	where	SqlAndTag で指定された条件を元に、WHERE 句を作成します。	×
29	and	Where 句を作成するための条件を指定します。	×
30	viewLink	検索結果に対してリンクを表示するときに使用します。DBTableModel オブジェクトを表示する、共通オブジェクトです。	×
31	columnCheck	登録データの物理チェックを行うタグです。	×
32	userInfo	UserInfo オブジェクトに所定のキーで文字列を設定させるタグです。	×
33	frame	フレームを作成する簡易タグです。HTML の flame タグと同等で、国 際化対応と引数の受け渡しが可能です。	○

第2章 Taglib一覧

No	名称	概要説明	HTML 拡張
34	text	BODY 部に記述された文字列を、ISO8859-1 コードから、ユーザー情報のエンコーディングに変換します。また、request 引数{@xxxx}も、変換されます。これは、XML 形式で日本語コードのパーサーが、JSP エンジン (Tomcat) でサポートされるまでの、暫定的なタグです。	×
35	help	ヘルプ画面 (主に HTML で作成したファイル) を表示するリンクを自動作成します。	×
36	entryQuery	EntryQueryTag は、DBTableModel オブジェクトを作成する、共通オブジェクトです。このオブジェクトに、dbid を与えることにより、dbid に対応した Query オブジェクト (のサブクラスのオブジェクト) が作成されます。	×
37	requestCache	request オブジェクトを キャッシュすることにより、再検索時や、各画面遷移時の項目の持ち回りを行います。	×
38	shell	val1 属性 と val2 属性の文字列表現の比較により BODY を表示/非表示を切り替える簡易タグです。	×
39	errorMessage	エラーメッセージを 表形式で表示するタグオブジェクトです。	×
40	forward	フォワードを実行するタグです。フォワード時に、指定の引数をセットできます。	×
41	columnMarker	検索結果に対して様々な属性を付加するときに使用します。DBTableModel のカラムの属性を利用して、自由に HTML のタグを組み込むことが可能になります。	×
42	viewMarker	検索結果に対して様々な属性を付加するときに使用します。DBTableModel オブジェクトの表示に HTML タグをマークするオブジェクトです。子タグとして、ColumnMarkerTag の Body タグをパースして、タグを作成します。	×
43	file	ファイル関連の取扱いをおこなうタグです。各種アクション に応じた振る舞いを行います。	×
44	fileQuery	ファイル検索をおこなうタグです。指定の SELECT 条件の項目をディレクトリより所得します。	×
45	fileOption	ファイルリストの作成をおこなうタグです。本タグは Select タグの BODY 部に指定します。	×
46	fileWhere	fileQuery で検索するときに使用する検索条件を指定します。	×
47	img	GIF 等のイメージファイルを表示するタグです。	○
48	sql	SQLTag は、画面上にデータベースの検索結果を表示するタグです。	×
49	tbody	検索結果に対して表示フォーマットを変更するときに使用します。theadTag は、ボディー部のフォーマットを定義するタグです。	×

No	名称	概要説明	HTML 拡張
50	tfoot	検索結果に対して表示フォーマットを変更するときに使用します。theadTag は、フッター部のフォーマットを定義するタグです。	×
51	thead	検索結果に対して表示フォーマットを変更するときに使用します。theadTag は、ヘッダー部のフォーマットを定義するタグです。	×
52	columnEditor	検索結果に対して様々な属性を付加するときに使用します。DBTableModel オブジェクトの表示に HTML タグをマークするオブジェクトです。子タグとして、ColumnMarkerTag の Body タグをパースして、タグを作成します。	×
53	favoriteLink	お気に入りリンクを作成する簡易タグです。画面検索時の引数やユーザー情報を元にして、ダイレクトに再検索できるリンクを作成します。	×
54	writeChart	DBTableModel オブジェクトをチャートで表示する、カスタムタグです。このオブジェクトに、データ(DBTableModel)と、コントローラ(DBTableWriter)を与えて、外部からコントロールすることで、各種形式でデータ(DBTableModel)を表示させることが可能です。	×
55	comment	JSP アプリケーションに対してバージョン管理できるようにコメントを記述するためのタグです。	×
56	statusBar	管理者が全てのユーザーにメッセージを同報できるタグオブジェクトです。	×
57	report	DBTableModel オブジェクトをレポート形式に返還するタグリブクラスです。このオブジェクトに、データ(DBTableModel)と、コントローラ(DBTableReport クラス)を与えて、外部からコントロールすることで、各種形式でデータ(DBTableModel)を表示させることが可能です。	×
58	value	ValueTag は JSP 上からキー、値を設定することにより、リクエスト情報にキー、値を記録させ、キーにより取り出し可能なカスタムタグです。	×
59	schedule	Context (アプリケーション) 共有のタイマー機能を用いて、スケジュールを、設定します。	×
60	ldapQuery	LDAP にアクセスするための、ldapQuery タグです。このタグで検索した結果は、DBTableModel にセットされるため、JDBCQuery と同様に、view タグで表示させることが可能です。	×
61	regQuery	Windows レジストリにアクセスするための、regQuery タグです。このタグで検索した結果は、DBTableModel にセットされるため、JDBCQuery と同様に、view タグで表示させることが可能です。	×
62	submit	サブミットボタンを表示するタグです。value 属性 value 属性に、forward したい JSP ファイル名を記述します。JSP ファイル名は、標準で、insert,copy,modify,delete などと指定します。実際には、各 JSP 画面(insert.jsp,copy.jsp,modify.jsp,delete.jsp)にリクエストが転送されます。	○

No	名称	概要説明	HTML 拡張
63	responseHeader	HttpServletResponse オブジェクトに対して、キャッシュコントロールするヘッダー情報をセットするタグです。Cache-Control、Content-Type、Location、Refresh、status、redirect をセットできます。	×
64	bodyHidden	隠しフィールドを作成する簡易タグです。	×
65	commonforward	フォワードキャッシュによりページ転送するタグです。	×
66	backGamen	戻るボタン/戻るリンクを表示するタグです。	○

②Taglib画面イメージ

下の画面イメージは、主なTaglibの使用例を示します。

The screenshot shows a web application interface with several annotated components:

- Input / column:** Search form with fields for 品番 (Item No.), 呼び名 (Call Name), 英語名 (English Name), 品名 (Item Name), 略式素子名 (Short Name), 標準F (Standard F), 機能品番F (Function Item No. F), and 集中分散CD (Concentration/Dispersion CD).
- input type=radio:** Radio buttons for 品番,呼び名 (Item No., Call Name), 呼び名,品番 (Call Name, Item No.), and メーカー名,呼び名 (Manufacturer Name, Call Name).
- select:** A dropdown menu for 集中分散CD with options like 購入品 (Purchased Item) and 製造品 (Manufactured Item).
- button:** Buttons for 検索(F) (Search), クリア (Clear), BACK, and NEXT.
- message:** A message box indicating 50 items were searched.
- submit:** Action buttons like 追加 (Add), 複製(C) (Copy), 変更(M) (Change), and 削除(D) (Delete).
- view:** A table with columns for No, 品番 (Item No.), 呼び名 (Call Name), 英語名 (English Name), 品名 (Item Name), 略式素子名 (Short Name), メーカー略号 (Manufacturer Abbreviation), メーカー名 (Manufacturer Name), 標準F (Standard F), 機能品番F (Function Item No. F), 集中分散CD (Concentration/Dispersion CD), 架空CD (Virtual CD), and 図表 (Chart).

3. Taglib属性一覧

TaglibはXML形式で表され、属性を使用します。属性として使用できる文字列は、タグの名称ごとに異なります。

- ボディ部ありの場合

```
<mis:タグの名称  
  属性 1="値 1"  
  属性 1="値 1"  
  ...>
```

ボディ部

```
<mis:タグの名称>
```

- ボディ部なしの場合

```
<mis:タグの名称  
  属性 1="値 1"  
  属性 1="値 1"  
  .../>
```

①専用属性一覧

下表は各 Taglib の属性を示しています。

尚、汎用属性は、HTML拡張タグに共通で使用される属性のため、該当する Taglib について「汎用属性」欄に○をつけて示し、詳細表示を割愛しています。

また、属性の値にも入力可能な条件があります。それについては、『第Ⅲ部 Tagribリファレンス』を参照ください。

NO	1	2	3	4	5
名称	column	input	select	option	queryOption
class	ColumnTag	InputaTag	SerectTag	OptionTag	QueryOption Tag
body	×	×	○	×	○
汎用属性	○	○	○	○	×
属性	name	type	name	selected	tableId
	value	name	size	value	dbid
	td	value	multiple	label	skipRowCount
	addNoValue	size	msg	code	maxRowCount
	lbl	maxlength	lbl	lbls	language
	must	checked	language	msg	value
	writable	src	must	lbl	separator
	colspan	td	td	language	
	maxlength	alt	colspan		
	size	accept			
	language	usemap			
	dfaultVal	ismap			
	useDefaultVal	msg			
		lbl			
		language			
		debug			
		lbls			
	colspan				

NO	6	7	8	9	10
tld	orderby	button	textarea	query	view
class	OrderbyTag	ButtonTag	TextareaTag	QueryTag	ViewFormTag
body	×	○	○	○	○
汎用属性	×	○	○	×	×
属性	columns	name	name	tableId	viewFormId
	checked	value	rows	dbid	command
	lbls	type	cols	command	startNo
	checkedValue	msg	msg	skipRowCount	pageSize
		lbl	lbl	maxRowCount	language
		language	language	queryType	tableId
			debug	scope	columnWritable
				debug	columnDisplay
				displayMsg	viewFormType
				names	scope
				stopZero	noWritable
					noDisplay
					writable
					checked
					pagePlus
					rowspan
					skip
					viewLinkId
					selectedType
					optionType Attributes
				noMessage	
				backLinkCount	

第2章 Taglib一覧

NO	11	12	13	14	15
tld	entry	columnSet	update	upload	link
class	EntryTag	columnSet Tag	UpdateTag	FileUploadTag	LinkTag
body	○	×	○	×	○
汎用属性	×	×	×	×	○
属性	tableId	command	tableId	maxPostSize	name
	command	value	dbid	language	href
	rows	action	command	tableId	hreflang
	scope	columnId	tableName	fileURL	type
		columnNo	columns		rel
			checkFile		rev
			setFile		charset
			queryType		shape
			scope		coords
			debug		target
			displayMsg		msg
			names		lbl
					language
					column
					keys
					value
					onLink
					fileCheck
				filePath	

NO	16	17	18	19	20
tld	hidden	debug	equals	message	writeTable
class	HiddenTag	DebugTag	EqualsTag	MessageTag	WriteTableTag
body	○	○	○	○	×
汎用属性	×	×	×	×	×
属性	name	debug	val1	msg	separator
	value		val2	lbl	header Sequence
	names		ignoreCase	language	url
	vals		startsWith	comment	filename
			nullCheck	val0	encode
			notEquals	val1	fileAppend
			debug	val2	tableId
				val3	direct
				val4	langusge
				val5	fleURL
				val6	writerClass
				val7	zip
				val8	scope
			val9	disposition	

NO	21	22	23	24	25
tld	space	plsqlUpdate	mailSender	mailReceiver	writeCheck
class	SpaceTag	PlsqlUpdateTa g	MailSenderTag	MailReceiver Tag	writeCheckTag
body	×	○	○	×	○
汎用属性	×	×	×	×	×
属性	labelWidth	tableId	charset	user	
	valueWidth	queryType	from	password	
		dbid	to	columnList	
		scope	cc	host	
		command	bcc	tableId	
		maxRowCount	subject	debug	
		skipRowCount	host	command	
		debug	filename		
		displayMsg	debug		
		names			
		dbType			
		selectedAll			

第2章 Taglib一覽

NO	26	27	28	29	30
tld	appear	readTable	where	and	viewLink
class	AppearTag	ReadTableTag	SqlWhereTag	SqlAndTag	ViewLinkTag
body	×	×	○	×	○
汎用属性	×	×	×	×	×
属性	startKey	separator	startKey	startKey	command
	value	fileURL		value	viewLinkId
	defaultVal	filename		multi	
		encode			
		readerClass			
		maxRowCount			
		language			
		tableId			
		command			
		modifyType			
		displayMsg			

NO	31	32	33	34	35
tld	columnCheck	userInfo	frame	text	help
class	ColumnCheck Tag	UserInfoTag	FrameTag	TextTag	HelpTag
body	×	○	×	○	×
汎用属性	×	×	○	×	×
属性	command	command	src	value	guiInfoKey
	columns	key	name		extension
	nullCheck	value	longdesc		msg
	checkType	dbid	marginwidth		fileCheck
	maxRowCount	query	marginheight		target
	minRowCount	nullSet	noresize		viewMsg
			scrolling		
			frameborder		
			changeSrc		
			keys		
			dbTable		

NO	36	37	38	39	40
tld	entryQuery	requestCache	shell	errorMessage	forward
class	EntryQuery Tag	RequestCache Tag	ShellTag	ErrorMessage Tag	ForwardTag
body	○	×	×	×	×
汎用属性	×	×	×	×	×
属性	tableId		program	command	page
	queryType		useBatch	clear	dbkeys
	dbid		wait	viewType	keys
	scope		stdout	debug	vals
	command		stderr	displayMsg	scope
	maxRowCount		debug	warningMsg	
	skipRowCount				
	debug				
	displayMsg				
	names				

NO	41	42	43	44	45
tld	columnMarker	viewMarker	file	fileQuery	fileOption
class	ColumnMarker Tag	ViewMarkerTa g	FileTag	FileQueryTag	FileOptionTag
body	○	○	○	○	○
汎用属性	×	×	×	×	×
属性	column	command	fileURL	select	select
	columns		file1	form	from
	onMark		file2	orderBy	value
			columns	desc	orderBy
			action	multi	desc
			tableId	tableId	fileType
				level	
				command	
				maxRowCount	
				displayMsg	
			overflowMsg		

第2章 Taglib一覽

NO	46	47	48	49	50
tld	fileWhere	img	sql	tbody	tfoot
class	FileWhereTag	ImageTag	SQLTag	TbodyTag	TfootTag
body	○	×	○	○	○
汎用属性	×	○	×	×	×
属性	name	name	command	rowspan	rowspan
	value	src	defaultVal		
	action	alt	thead		
	ignoreCase	longdesc			
		width			
		height			
		usemap			
		ismap			
		align			
		border			
		hspace			
		vspace			

NO	51	52	53	54	55
tld	thead	columnEditor	favoriteLink	writeChart	comment
class	TheadTag	ColumnEditor Tag	FavoriteLink Tag	WriteChart Tag	CommentTag
body	○	○	○	×	○
汎用属性	×	×	×	×	×
属性	rowspan	lang	target	writerClass	title
		column	direct	title	version
		label	method	width	build
		className	title	height	date
		size	msg	maxRange	author
		writable	lbl	minRange	
		renderer		noteDisplay	
		editor		language	
		codeSel		tableId	
		dbType		scope	
		defaultVal			
		parameter			
		must			
		optionAttributes			
		addNoValue			

NO	56	57	58	59	60
tld	statusBar	report	value	schedule	ldapQuery
class	StatusBarTag	ReportTable Tag	ValueTag	ScheduleTag	LDAPQuery Tag
body	○	×	○	×	×
汎用属性	×	×	×	×	×
属性	action	headerKeys	command	command	initctx
	timer	headerVals	action	timerTask	providerURL
	brink	footerKeys	key	name	entrydn
	image	footerVals	value	comment	searchbase
	src	pageEndCut	defaultVal	singleton	filter
		fileURL	nullSet	delay	searchScope
		filename	dbid	period	timeLimit
		outFileURL	parameter	fixedDelay	derefLinkFlag
		outFilename	scope	commonforward	maxRowCount
		encode	debug		returningObj Flag
		reportClass			attributes
		language			orderBy
		tableId			desc
		scope			tableId
		debug			command
					displayMsg
					overflowMsg
					debug
				scope	

第2章 Taglib一覽

NO	61	62	63	64	65
tld	regQuery	submit	responseHeader	bodyHidden	commonforward
class	RegistryQuery Tag	SubmitTag	ResponseHeader Tag	BodyHidden Tag	Commonforward Tag
body	×	×	×	○	×
汎用属性	×	○	×	×	×
属性	regQuery	value	cacheControl	name	
	baseKey	command	contentType	value	
	key	gameId	refresh		
	value	msg	refreshURL		
	maxRowCount	lbl	redirect		
	orderBy	language	status		
	maxLevel		location		
	like				
	tableId				
	command				
	displayMsg				
	overflowMsg				
	debug				
	scope				

NO	66
tld	backGamen
class	BackGamen Tag
body	×
汎用属性	○
属性	type
	command
	gamenId
	keys
	vals
	href
	target
	id
	lang
	dir
	title
	style
	tabindex
	accesskey
	clazz
	msg
lbl	
language	

②汎用属性一覧

下表に汎用属性一覧を示します。taglibの汎用属性は、HTMLタグの共通属性をまとめて設定しています。タグによっては、設定できても無効な属性がありますので、ご注意ください。

汎用属性名称
clazz
id
title
style
lang
dir
readonly
disabled
tabindex
accesskey
optionAffributes

第 II 部

Taglibリファレンス

『自らの運命をコントロールせよ。
さもなければ、他人にコントロールされることになるだろう。』
ジャック・ウェルチ（ゼネラルエレクトロニックCEO）

構成は、次のとおりです。
第 I 部で紹介したTaglibについて、詳細内容を解説します。

第 3 章 Taglibリファレンス
各Taglibの詳細内容、属性について、例文を交えながら解説します。

Web
Web
アプリケーション

第3章 Taglibリファレンス

この章では、各Taglibの詳細内容、属性について、例文を交えながら解説します。

1. mis:column

①概要

形式	<code><mis:column name=… value=… /></code>
body	なし

②説明

ColumnTag は、インプットフィールド(入力フィールド)を作成する簡易タグです。name 属性をキーにして、DBCColumnResource よりリソースを読み取ります。

name 属性に DBCColumnResource.properties のキーを与えることで、そのリソースに指定した表示形式で画面に表示します。あくまでデフォルトの DB 項目と同じ条件のデータを簡易的に表示させることが主目的ですので、細かい設定 (size、maxlength など) を設定したい場合は inputTag をお使い下さい。

③属性

属性	意味	必須
name	この部品の名前を指定します。 <code><mis:column name="PN" /></code>	○
value	初期値を指定します。 <code><mis:column name="PN" value="syokichi" /></code>	
td	テーブル形式の <code><td></code> タグを使用するかどうかを指定します。 デフォルトは、使用する("yes") です。 <code><td><mis:column name="PN" td="no" /> </td></code>	
addNoValue	コンボボックスの場合、値なしの Option を Select に含めるかどうかを指定します。 デフォルトは、含める("true") です。 <code><mis:column name="CDC" addNoValue="false" /></code>	

属性	意味	必須
lbl	ラベルを指定します。ラベルを変更するときに、lbl 属性を使います。ラベル ID は、所定の language に基づく ResourceManager の getLabel(id) を呼び出し、その結果を msglbl に登録します。getMsglbl() で取り出せます。ラベル ID とメッセージ ID は同時には登録できません。	
must	必須入力を表す色に変えるかどうか指定します。デフォルトは、必須でない("false") です。(注意)必須チェックとは連動していません。 <mis:column name="PN" must="true" />	
writable	書き込みタイプかどうかを指定します。デフォルトは、書き込みタイプ("true") です。 <mis:column name="PN" writable="false" />	
colspan	フィールド部分の colspan を指定します。通常は、3 を指定して、フィールド、ラベル、フィールドの 組み合わせ部分に、長いフィールドが適用されるように使用します。	
id	id セット	
maxlength	最大入力文字数を指定します。 <mis:column name="PN" maxlength="20" />	
size	横幅を指定します。 <mis:column name="PN" size="20" />	
clazz	class セット。Java の言語使用上、class で作成できないため、代用として clazz を使用しています。html で作成される属性は、class で作成されます。	
lang	lang セット。	
dir	dir セット。	
title	title セット。	
style	style セット。	
readonly	readonly セット。readonly="readonly" 以外の値はセットできないように 制限をかけます。(大文字/小文字も判別します)	
disabled	disabled セット。disabled="disabled" 以外の値はセットできないように 制限をかけます。(大文字/小文字も判別します)	
tabindex	tabindex セット。	
accesskey	accesskey セット。	
language	言語コードを指定します。この言語コードに基づいて、表示のラベルをリソースから作成します。	

属性	意味	必須
optionAttributes	optionAttributes セット。JavaScript などの HTML 基本タグ以外の属性を、そのまま タグとして使用します。そのときに、(') は、(")に置き換えます。	
defaultVal	処理コマンド。value 属性に値がセットされていないときに、デフォルト値をセットします。	
useDefaultVal	処理コマンド。デフォルト値がセットされていないときに、DBCColumnResource.properties ファイルの デフォルト値の値を使うかどうかを指定します。使う場合は、true / 使わない場合は false (デフォルト)をセットします。	

④使用例

例 1 編集種別が Textの場合は、テキストフィールドが表示されます。
通常は NAME 属性を指定するだけです。

```
<mis:column name="PN" />
```

例 2 編集種別が Menu の場合は、メニューが表示されます。

```
<mis:column name="CDC" />
```

例 3 初期値をセットするときは value 属性を使います。

```
<mis:column name="PN" value="syokichi" />
```

例 4 ラベルを変更するときは lbl 属性を使います。

```
<mis:column name="PN" lbl="PNL" />
```

例 5 リソースファイルからコンボボックスも自動作成されます。

```
<mis:column name="CDC" />
```

例 6 NULL 値を消すときは addNoValue 属性を使います。

```
<mis:column name="CDC" addNoValue="false" />
```

例 7 属性は、複数組み合わせで使えます。

```
<mis:column name="CDC" addNoValue="false" must="true" />
```

例 8 色を変更するときは must 属性を使います。

```
<mis:column name="PN" must="true" />
```

例9 自動作成される td タグやラベルを外すときは td 属性を使います。

```
<mis:column name="PN" td="false" />
```

例10 自動作成される td タグを外し、ラベルとフィールドは残します。

```
<mis:column name="PN" td="no" />
```

例 11 非入力にするときは writable 属性を使います。

```
<mis:column name="PN" writable="false" />
```

例 12 強制的に SIZE の変更をするときは size 属性を使います。

```
<mis:column name="PN" size="20" />
```

2. mis:input

①概要

形式	<mis:input name=… value=… />
body	なし

②説明

InputTag は、ユーザーがデータの入力や選択、データの送信などを行うための部品(入力フィールドやボタン)を作成するタグです。name 属性に LabelResource.properties のキーを与えることで、ロケールにあわせたリソースを使用して、画面に表示します。ロケールは、session 登録項目の MISSystem#LANG_KEY をデフォルトで使用し、language 属性で定義されていれば、そちらを優先します。

より簡易的な入力用タグとして、ColumnTag がありますので、通常は、ColumnTag をお使い下さい。各属性は、{@xxxx} 変数が使用できます。これは、ServletRequest から、xxxx をキーに値を取り出し、この変数に 割り当てます。つまり、この xxxx をキーにリクエストすれば、この変数に値をセットすることができます。

```
http://localhost/query.jsp?KEY1=VAL1&KEY2=VAL2
```

のようなリクエストで、{@KEY1} とすれば、VAL1 がセットされます。

③属性

属性	意味	必須
type ※1	この属性は、作成するコントロールの形式を指定します。 表示形式は (text/password/checkbox/radio/submit/reset/button/image/ file) が指定できます。 指定しなければ、type="text"として動作します。	
name	この属性は、コントロール名を示します。 <mis:input name="PN" />	○
value	この属性は、当該コントロールの初期値を指定します。type 属 性の値が"radio"か"checkbox"以外の場合、この属性は省略可 能です。初期値を指定します。 <mis:input name="PN" value="syokichi" />	
size	この属性は、ユーザエージェントに当該コントロールの初期幅 指定を示します。この幅はピクセル値だが、type 属性が"text" または"password"の場合は、文字数を示す整数を値とします。 横幅を指定します。 <mis:input name="PN" size="20" />	
maxlength	type 属性の値が"text"または"password"の場合、この属性は ユーザが入力できる最大文字数を指定します。最大文字数は size 指定を超え得るが、超えた場合ユーザエージェントはスク ロール機構を提供する必要がある。この属性のデフォルト値は 「無制限」です。最大入力文字数を指定します。 <mis:input name="PN" maxlength="20" />	
checked	この論理型属性は、type 属性値が"radio"または"checkbox"の 場合に、“入”であることを指定します。ユーザエージェントは、 他のコントロール形式においては本属性を無視しなければならない。 チェックボックスやラジオボタンのとき、初めからチェックされた 状態を表示します。 check 値が "checked" の場合のみ、実行されます。 <mis:input name="PN" type="checkbox" check="checked" />	
src	この属性は、type 属性値が"image"の場合、グラフィカルな提 出ボタンの装飾に用いる画像の所在を指定します。src セット。	
td ※2	テーブル形式の <td> タグを使用するかどうかを指定します。 (独自タグ) デフォルトは、使用する("yes") です。 <mis:input name="PN" td="false" />	
id	文書内識別子 id セット	
lang	言語情報 lang セット	
dir	テキスト方向 dir セット	
title	要素情報 title セット	
style	行内スタイル情報 style セット	

属性	意味	必須
alt	代替テキスト alt セット	
accept	サーバに対する正当な MIME タイプ accept セット	
readonly	読み出し専用指定 readonly セット readonly="readonly" 以外の値はセットできないように 制限を かけます。(大文字/小文字も判別します)	
disabled	選択不能指定 disabled セット disabled="disabled" 以外の値はセットできないように 制限を かけます。(大文字/小文字も判別します)	
tabindex	タブ移動 tabindex セット	
accesskey	アクセスキーaccesskey セット	
usemap	クライアント側イメージマップ usemap セット	
ismap	サーバ側イメージマップ ismap セット	
	組込みイベント onfocus、onblur、onselect、onchange、onclick、ondblclick、 onmousedown、onmouseup、onmouseover、onmousemove、 onmouseout、onkeypress、onkeydown、onkeyup	
clazz	class 属性が使えないための代替。HTML 化したときは class に なります。class セット Java の言語使用上、class で作成できないため、代用として clazz を使用しています。html で作成される属性は、class で 作成されます。	
msg	messageResource より取り込むキー メッセージを指定します。 メッセージ ID は、所定の language に基づく ResourceManager の getMessage(id) を呼び出し、その結果 を msglbl に登録します。getMsglbl() で取り出せます。ラベル ID とメッセージ ID は同時には登録できません。	
lbl	LabelResource より取り込むキーラベルを指定します。 ラベル ID は、所定の language に基づく ResourceManager の getLabel(id) を呼び出し、その結果を msglbl に登録しま す。getMsglbl() で取り出せます。ラベル ID とメッセージ ID は同 時には登録できません。	
language	Resource の言語を指定します。言語コードを指定します。 この言語コードに基づいて、表示のラベルをリソースから作成し ます。	
debug	デバッグ状態にします。 true/false デバッグ情報を 出力する/しないを指定します。	

属性	意味	必須
lbls	ラベル(複数)を指定します。 シングルラベルの lbl 属性との違いは、ここではラベルを複数カンマ区切りで 渡すことが可能であることです。これにより、“A,B,C” という値に対して、“A のラベル表示、B のラベル表示、C のラベル表示” という具合に文字列を 再合成します。これは、SQL 文の Ordr By 句で、ソート順を指定する場合などに便利です。 <mis:orderBy columns="MKNMJ, MKCD, MKNMEN" lbls="MKNMJ, MKCD, MKNMEN" />	
optionAttributes	optionAttributes セット JavaScript などの HTML 基本タグ以外の属性を、そのまま タグとして使用します。そのときに、(')は、(")に置き換えます。	
colspan	フィールド部分の colspan を指定します。 通常は、3 を指定して、フィールド、ラベル、フィールドの 組み合わせ部分に、長いフィールドが適用されるように使用します。 <mis:input name="PN" colspan="3" />	

※ 1 type 属性

input タグの type 属性には、以下の値を設定することができます。

input の type 属性	意味	必須
text	1 行だけのテキスト入力コントロールを作成します。	
password	このコントロール形式はパスワードのようにデリケートな内容の入力に用います。	
checkbox	チェックボックスを作成します。	
radio	ラジオボタンを作成します。	
submit	提出ボタンを作成します。	
image	グラフィカルな提出ボタンを作成します。src 属性値でボタンを装飾する画像の URI を指定します。アクセシ性を考慮し、著者は alt 属性によって画像の代替テキストを提供する必要があります。	
reset	リセットボタンを作成します。	
button	押しボタンを作成します。ユーザエージェントは、ボタンのラベルとして value 属性値を用いる必要があります。	
hidden	隠れコントロールを作成します。	
file	ファイル選択コントロールを作成します。ユーザエージェントは value 属性値をファイル名の初期値としてもよいです。。	

type が submit、button、hidden の場合には、各 Taglib を利用して下さい。

※2 td 属性

input タグでは、使用者の便宜を図るため、あらかじめラベルを一体化した input タグを作成します。ラベルは、国際化を考慮して、msg 属性 または、lbl 属性で指定します。その場合、表示の体裁を整えるため、table タグの td タグを組み込んでいます。

出力結果 type="radio" OR "checkbox" の場合
 type="text" OR "password" OR "image" OR "file" の場合

④使用例

例 1 lbl属性が指定されていない場合は、name 属性をキーにして LabelResource を読み取ります。

```
<mis:input name="OYA" />
```

例 2 lbl属性を指定することで、ラベルだけを付替えることができます。

```
<mis:input name="OYA" lbl="KO"/>
```

例 3 InputTag では、size 属性、maxlength 属性を使用できます。

```
<mis:input name="PN" size="13" maxlength="11"/>
```

例 4 通常は NAME 属性を指定するだけです。

```
<mis:input name="OYA"/>
```

例 5 ラベルを変更するときは lbl 属性を使います。

```
<mis:inputname="PN" lbl="KO"/>
```

例 6 強制的に SIZE の変更をするときは size 属性を使います。

```
<mis:inputname="PN" size="13" maxlength="11"/>
```

例 7 自動作成される td タグやラベルを外すときは td 属性を使います。

```
<mis:input name="OYA" td="false"/>
```

例 8 自動作成される td タグを外し、ラベルとフィールドは残します。

```
<mis:input name="OYA" td="no"/>
```

例 9 avaScript のイベントをコーディングするときは、optionAttributes 属性を使います。

```
<mis:input name="PN" optionAttributes="onBlur='blr()' onFocus='fcs()'/>
```

例 10 ラベル部分と入力フィールド部分がテーブルタグの<td>により左右に分割されます。

```
<mis:input type="radio" name="OYA"/>
```

例 11 HTML 表示時は、前後に<tr>タグで囲って、整形できます。

```
<table><tr><mis:input name="PN" value="{@PN}"/></tr>
<tr><mis:input name="CD" value="{@CD}"/></tr>
</table>
```

例 12 BODY 部分は、無視されます。コメント等に使用できます。HTML ファイルには、コメント部分は出力されません。

```
<table><tr><mis:input name="PN" value="{@PN}">部品入力フィールド</mis:input></tr>
<tr><mis:input name="CD" value="{@CD}">コードフィールド</mis:input></tr>
</table>
```

3. mis:select

①概要

形式	<mis:select name=・・・ > ・・・ </mis:select>
body	あり

②説明

SelectTag は、プルダウンメニューを作成する簡易タグです。メニュー内の選択肢は、selectタグの内容である OptionTag、または QueryOptionTag によって指定されます。name 属性に LabelResource.properties のキーを与えることで、ロケールにあわせてリソースを使用して、画面に表示します。ロケールは、ユーザー情報の lang 属性をデフォルトで使用し、セットされていない場合は、リクエスト情報のロケールから取得します。

language 属性で定義されていれば、そちらを優先します。

各属性は、{@xxxx} 変数を使用できます。これは、ServletRequest から xxxx をキーに値を取り出し、この変数に割り当てます。つまり、この xxxx をキーにリクエストすれば、この変数に値をセットすることができます。

http://localhost/query.jsp?KEY1=VAL1&KEY2=VAL2
 のようなリクエストで、{@KEY?}1} とすれば、VAL1 がセットされます。

このタグは、BODY 部に option タグを持つことができます。option タグを記入した場合は、option タグを元に、プルダウンメニューを作成します。

逆に、option タグがセットされない場合は、name 属性の、CodeResource.properties から Option 文字列を作成して、プルダウンメニューを作成します。

③属性

属性	意味	必須
name	この属性はコントロール名を示します。メニューの名称を指定します。	○
size	Select タグがスクロール式リストボックスとして表示される場合、この属性はリスト中で一望させたい行数を指定します。視覚系ユーザエージェントは select タグを常にリストボックスとして表示する必要はなく、例えばプルダウンメニューなど他の機構を用いて表すこともできます。横幅を指定します。	
multiple	この論理型属性が設定された場合、複数選択が許容されます。設定されない場合、1つの select タグでは1つの選択しか許されません。複数選択を可能にします。multiple 値が "multiple" の場合のみ、実行されます。	
id	文書内識別子 id セット	
lang	言語情報 lang セット	
dir	テキスト方向 dir セット	

属性	意味	必須
title	要素情報 title セット	
style	行内スタイル情報 style セット	
disabled	選択不能指定 disabled セット disabled="disabled" 以外の値はセットできないように 制限を かけます。(大文字/小文字も判別します)	
tabindex	タブ移動 tabindex セット	
	組込みイベント onclick 、 ondblclick 、 onmousedown 、 onmouseup 、 onmouseover 、 onmousemove 、 onmouseout 、 onkeypress 、 onkeydown、onkeyup	
clazz	class 属性が使えないための代替。HTML 化したときは class に なります。class セット Java の言語使用上、class で作成できな いため、代用として clazz を使用しています。 html で作成される属性は、class で作成されます。	
msg	MessageResource より取り込むキーメッセージを指定します。 メッセージ ID は、所定の language に基づく ResourceManager の getMessage(id) を呼び出し、その結果 を msglbl に登録します。getMsglbl() で取り出せます。ラベル ID とメッセージ ID は同時には登録できません。	
lbl	LabelResource より取り込むキーラベルを指定します。ラベルを 変更するときに、lbl 属性を使います。ラベル ID は、所定の language に基づく ResourceManager の getLabel(id) を呼 び出し、その結果を msglbl に登録します。getMsglbl() で取り 出せます。ラベル ID とメッセージ ID は同時には登録できませ ん。	
language	Resource の言語を指定します。言語コードを指定します。この 言語コードに基づいて、表示のラベルをリソースから作成しま す。	
must	必須入力を表す色に変わるかどうか指定します。デフォルトは、 必須でない("false") です。(注意)必須チェックとは連動してい ません。	
td	テーブル形式の <td> タグを使用するかどうかを指定します。 デフォルトは、使用する("yes") です	
optionAttribut es	optionAttributes セット JavaScript などの HTML 基本タグ以 外の属性を、そのまま タグとして使用します。そのときに、(') は、(") に置き換えます。	
colspan	フィールド部分の colspan を指定します。通常は、3 を指定し て、フィールド、ラベル、フィールドの 組み合わせ部分に、長 いフィールドが適用されるように使用します。 <mis:select name="CDK" colspan="3" />	

ヒント



select タグはメニューを生成します。メニュー中の各選択肢は option タグによって提示されます。ひと
つの select タグには最低 1 つの option タグが含まれている必要があります。

④使用例

例1 CodeResource プロパティの値を使用したいときは code 属性を使います。

```
<mis:select name="CDC">
  <mis:option code="CDC" />
</mis:select>
```

例2 選択項目の一番上に msg 属性を用いて、項目を付け足すことも可能です。

```
<mis:select name="CDC">
  <mis:option msg="MSG0001" />
  <mis:option code="CDC" />
</mis:select>
```

例3 選択肢の中から複数を選択できるようにする場合に指定します。

```
<mis:select name="CDC" multiple="multiple">
  <mis:option code="CDC"/>
</mis:select>
```

例4 選択メニューをデータベースの検索結果から作成することが可能です。

```
<mis:select name="CDC">
  <mis:queryOption>
    <jsp:text>
      select ITEM,ITEM from OITMSTRU
    </jsp:text>
  </mis:queryOption>
</mis:select>
```

例5 プロパティの値を使用したいときは lbl 属性を使います。

```
<mis:select name="CDC">
  <mis:option lbl="CDC1"/>
</mis:select>
```

例10 選択不可にするときは disabled 属性を使います。

```
<mis:select name="CDC">
  <mis:option msg="MSG0001" disabled="disabled" />
  <mis:option code="CDC" />
</mis:select>
```

4. mis:option

①概要

形式	<mis:option code="..." />
body	なし

②説明

OptionTag は、SelectTag の中で使用される項目選択用オプションを作成するタグです。プルダウンメニューの選択項目を作成します。name 属性は、LabelResource.properties のキーを与えることで、使用する上位のタグのロケールにあわせたリソースを使用して、画面に表示します。よって、このタグではロケールは指定できません。selected 属性は、そのタグが選ばれている場合を、“true” で指定します。デフォルトは、“false” です。

各属性は、{@xxxx} 変数が使用できます。これは、ServletRequest から、xxxx をキーに値を取り出し、この変数に割り当てます。つまり、この xxxx をキーにリクエストすれば、この変数に値をセットすることができます。

`http://localhost/query.jsp?KEY1=VAL1&KEY2=VAL2`

のようなリクエストで、{@KEY}1} とすれば、VAL1 がセットされます。

③属性

属性	意味	必須
selected	この論理型属性が設定されていた場合、当該選択肢が選択済みであることを指定します。オプションを選択済みの状態にセットします。ここでのラベルは、option タグのラベルであって、何ら変換は行いません。	
value	この属性は、当該コントロールの初期値を指定します。この属性が設定されていない場合、初期値は option タグの内容となります。値を指定します。	
label	この属性により、著者は、当該項目に関して option タグの内容そのものよりも短いラベルを指定できる。これが設定されている場合、ユーザーエージェントは当該項目のラベルとして、option タグの内容ではなく本属性の値を用いる必要があります。ラベルを指定します。ここでのラベルは、option タグのラベルであって、何ら変換は行いません。	
code	ResourceManager.getOption() で取得できる オプション文字列のキーです。コード ID をセットします。選択済み初期値を与えたい場合は、value で初期値を指定してください。	

属性	意味	必須
lbls	ラベルを複数 カンマ区切りでラベル ID をセットします。ラベル (複数)を指定します。シングルラベルの lbl 属性との違いは、ここではラベルを複数 カンマ区切りで 渡すことが可能であることです。これにより、“A,B,C” という値に対して、“A のラベル表示、B のラベル表示、C のラベル表示” という具合に文字列を再合成します。これは、SQL 文の Ordr By 句で、ソート順を指定します場合などに便利です。 <pre><mis:orderBy columns="MKNMJP,MKCD,MKNMEN" lbls="MKNMJP,MKCD,MKNMEN" /></pre>	
id	文書内識別子 id セット	
lang	言語情報 lang セット	
dir	テキスト方向 dir セット	
title	要素情報 title セット	
style	行内スタイル情報 style セット	
disabled	選択不能指定 disabled セット disabled="disabled" 以外の値はセットできないように 制限をかけます。(大文字/小文字も判別します)	
	組込みイベント onclick、ondblclick、onmousedown、onmouseup、onmouseover、onmousemove、onmouseout、onkeypress、onkeydown、onkeyup	
clazz	class 属性が使えないための代替。HTML 化したときは class にります。class セット。Java の言語使用上、class で作成できないため、代用として clazz を使用しています。html で作成される属性は、class で作成されます。	
msg	messageResource より取り込むキーメッセージを指定します。メッセージ ID は、所定の language に基づく ResourceManager の getMessage(id) を呼び出し、その結果を msglbl に登録します。getMsglbl() で取り出せます。ラベル ID とメッセージ ID は同時には登録できません。	
lbl	LabelResource より取り込むキーラベルを指定します。ラベル ID は、所定の language に基づく ResourceManager の getLabel(id) を呼び出し、その結果を msglbl に登録します。getMsglbl() で取り出せます。ラベル ID とメッセージ ID は同時には登録できません。	
language	Resource の言語を指定します。言語コードを指定します。この言語コードに基づいて、表示のラベルをリソースから作成します。	



ヒント

select タグはメニューを生成します。メニュー中の各選択肢は option タグによって提示され、1 つの select タグには最低 1 つの option タグが含まれている必要があります。

④使用例

例1 プロパティの値を使用したいときは code 属性を使います。

```
<mis:select name="CDC"> <!-- CodeResource -->
  <mis:option code="CDC"/>
</mis:select>
```

例2 選択項目の一番上に項目を付け足すことも可能です。

```
<mis:select name="CDC">
```

例3 プロパティの値を使用したいときは msg 属性を使います。

```
<mis:option msg="MSG0001"/> <!-- MessageResource -->
  <mis:option code="CDC"/>
</mis:select>
```

例4 選択項目の一番上に項目を付け足すことも可能です。

```
<mis:select name="CDC">
```

例5 LabelResource プロパティの値を使用したいときは lbl 属性を使います。

```
<mis:option lbl="CDC"/>
  <mis:option code="CDC"/>
</mis:select>
```

例6 選択肢の中から複数選択できるようにするときは multiple 属性を使います。

```
<mis:select name="CDC" multiple="multiple">
  <mis:option code="CDC"/>
</mis:select>
```

例7 選択不可にするときは disabled 属性を使います。

```
<mis:select name="CDC" disabled="disabled">
  <mis:option code="CDC"/>
</mis:select>
```

例8 選択肢を SELECT 文の結果から作成したいときは queryOption タグと組み合わせて使います。

```
<mis:select name="CDC">
  <mis:queryOption>
    <jsp:text>select NOSYN,NOSYN,':',NMSYN from DB01O ORDER BY 1</jsp:text>
  </mis:queryOption>
</mis:select>
```

5. mis:queryOption

①概要

形式	<pre><mis:queryOption> <jsp:text>SQL 文</jsp:text> </mis:queryOption></pre>
body	あり

②説明

QueryOptionTag は、OptionTag の仲間で、データベースからデータを検索して、項目選択用オプションを作成します。基本的には、QueryTag と同様の使い方をします。プルダウンメニュー内の選択肢を SELECT 文の結果から作成します。このオブジェクトに、queryId を与えることにより、queryId に対応した Query オブジェクト(のサブクラスのオブジェクト)が作成されます。ここで指定する SELECT 文は、『SELECT KEY,LABEL1,LABEL2,... FROM TABLE ...』形式 をしている必要があります。特別なケースとして、『SELECT KEY FROM TABLE ...』形式の場合は、LABEL に KEY が 使用されます。

各属性は、{@xxxx} 変数を使用できます。これは、ServletRequest から、xxxx をキーに値を取り出し、この変数に 割り当てます。つまり、この xxxx をキーにリクエストすれば、この変数に値をセットすることができます。

http://localhost/query.jsp?KEY1=VAL1&KEY2=VAL2

のようなリクエストで、{@KEY1} とすれば、VAL1 がセットされます。

③属性

属性	意味	必須
tableId	結果を DBTableModel に書き込んで、session に登録するときのキーです。デフォルト"hybs_dbTableModel"	
dbid	Query オブジェクトを作成する時の DB 接続 ID	
skipRowCount	データの読み始めの初期値を指定(オプション)	
maxRowCount	データの最大読み込み件数を指定(オプション)	
language	Resource の言語を指定します。	
value	Option の初期値で選ばれる値を指定します。	
separator	複数のラベルを合成するときに使用する項目区切り文字をセットします。デフォルト スペース	

④使用例

```
<mis:select name="CDC">
  <mis:queryOption>
    <jsp:text>
      selectNOSYN,NOSYN,NMSYNfromDB01ORDERBY1
    </jsp:text>
  </mis:queryOption>
</mis:select>
```

```
<mis:select name="CDC">選択項目の一番上に空白をセットしたいとき option タグを組合せることも可能
  <mis:option msg=""/>初期値を設定したいときは value 属性を使います
  <mis:queryOption value="61200" separator=":">
    <jsp:text>
      selectCDBK,CDBK,NMBKfromDB02ORDERBY1
    </jsp:text>
  </mis:queryOption>
</mis:select>
```

6. mis:orderBy

①概要

形式	<mis:ordeBy columns"...",..." />
body	なし

②説明

OrderByTag は、ラジオボタンで表示順(Order By 句)を指定する簡易タグです。name 属性値は、"ORDER_BY" 固定です。columns 属性は、カンマ区切りのカラム属性を設定します。ロケールは、ユーザー情報の lang 属性をデフォルトで使用し、セットされていない場合は、リクエスト情報のロケールから取得します。

③属性

属性	意味	必須
columns	Label を複数 カンマ区切りで渡すことができます。表示順対象のカラムを指定します。	○
checked	デフォルトでチェックを指定する場合に使用します。初めからチェックされた状態で表示します。check 値が "checked" の場合のみ、実行されます。	

属性	意味	必須
lbls	ラベル(複数)を指定します。ラベル(複数)を変更するときに、lbl属性を使います。シングルラベルの lbl 属性との違いは、ここではラベルを複数 カンマ区切りで 渡すことが可能であることです。これにより、“A,B,C” という値に対して、“A のラベル表示,B のラベル表示、C のラベル表示” という具合に文字列を再合成します。これは、SQL 文の Ordr By 句で、ソート順を指定する場合などに便利です。	
checkedValue	初めからチェックされた状態を表示します。チェックバリューが、自分自身のカラム名と同じであれば、チェックが付きます。通常、checkedValue = “{@ORDER_BY}” で、自動的にチェックされます。	

④使用例

例 1

```
<tr><mis:orderBy columns="PN,YOBI" checked="checked" /></tr>
<tr><mis:orderBy columns="YOBI,PN" /></tr>
<tr><mis:orderBy columns="MKNM,YOBI" /></tr>
```

例 2

```
<mis:orderBy columns="PN,YOBI" checked="checked" />
<mis:orderBy columns="YOBI,PN" />
<mis:orderBy columns="MKNM,YOBI" />
```

7. mis:button

①概要

形式	<mis:button type="..." />
body	あり

②説明

ButtonTag は、HTML の button タグと同等で名前と値を渡すことができます。HTML の button タグと同等で、国際化対応とボタンの値を渡すことができます。これにより、ボタンごとに必要な画面を選び出して制御を転送することが可能です。

BODY 部分が、ボタンの表示に使えます。つまり、ここにラベルや強調文字などのタグを記述すれば、そのままボタンとして表示されます。

③属性

属性	意味	必須
name	この属性は、コントロール名を示します。ボタンの名称を指定します。	
value	この属性は、ボタンの初期値を示します。ボタンの値を指定します。	
type	この属性は、ボタンの形式を宣言します。取り得る値は以下。 submit: 提出ボタンを作成します。デフォルト値です。 reset: リセットボタンを作成します。 button: 押しボタンを作成します。 ボタンのタイプを指定します。(submit/reset/button) ここでは、submit、reset、button のみを許可します。	○
id	文書内識別子 id セット	
lang	言語情報 lang セット	
dir	テキスト方向 dir セット	
title	要素情報 title セット	
style	行内スタイル情報 style セット	
disabled	選択不能指定 disabled セット disabled="disabled" 以外の値はセットできないように 制限を かけます。(大文字/小文字も判別します)	
accesskey	アクセスキー accesskey セット	
tabindex	タブ移動 tabindex セット	
	組込みイベント onfocus、onblur、onclick、ondblclick、onmousedown、 onmouseup、onmouseover、onmousemove、onmouseout、 onkeypress、onkeydown、onkeyup	

属性	意味	必須
clazz	class 属性が使えないための代替。HTML 化したときは class になります。class セット Java の言語使用上、class で作成できないため、代用として clazz を使用しています。html で作成される属性は、class で作成されます。	
msg	messageResource より取り込むキーメッセージを指定します。メッセージ ID は、所定の language に基づく ResourceManager の getMessage(id) を呼び出し、その結果を msglbl に登録します。getMsglbl() で取り出せます。ラベル ID とメッセージ ID は同時には登録できません。	
lbl	LabelResource より取り込むキーラベルを指定します。ラベルを変更するときに、lbl 属性を使います。ラベル ID は、所定の language に基づく ResourceManager の getLabel(id) を呼び出し、その結果を msglbl に登録します。getMsglbl() で取り出せます。ラベル ID とメッセージ ID は同時には登録できません。	
language	Resource の言語を指定します。言語コードを指定します。この言語コードに基づいて、表示のラベルをリソースから作成します。	



ヒント

button タグが生成するボタンと input タグが生成するボタンは、機能的には同等ですが、button タグの方がレンダリング能力があり、内容を持てます。例えば、画像を内容に持つ button の機能は、type 属性が "image" の input タグとそっくり同じですが、button タグは内容を持てます。

④使用例

例 1 プロパティのラベル使用したいときは lbl 属性に、LabelResource を使います。

```
<mis:button type="submit" name="test" value="1" lbl="test" >
```

例 2 プロパティのラベル使用したいときは msg 属性を使います。

```
<mis:button type="submit" name="test" value="1" msg="MSG0103" />
```

8. mis:textarea

①概要

形式	<mis:textarea name=… value=… /> <jsp:text > BODY 文 </jsp:text > </ mis:textarea>
body	あり

②説明

TextareaTag は、複数行の入力が可能なテキスト入力フィールドを作成するタグです。

③属性

属性	意味	必須
name	この属性はコントロール名を示します。名前を指定します。	○
rows	この属性は、同時に表示するテキスト行数を指定します。この行数を超える入力をユーザが可能でなければならないので、ユーザエージェントは、入力内容が表示領域を超える場合当該コントロールの内容をスクロールするなどの機構を提供する必要があります。高さ(行数)を指定します。	
cols	この属性は平均的文字幅に基づく文字数で、表示領域の幅を指定します。この幅より長い行をユーザが入力可能でなければならないので、ユーザエージェントは、入力内容が表示領域を超える場合当該コントロールの内容をスクロールするなどの機構を提供する必要があります。ユーザエージェントは長い行をスクロールによらずに折り返して表示してもよい。横幅を指定します。	
id	文書内識別子 id セット	
lang	言語情報 lang セット	
dir	テキスト方向 dir セット	
title	要素情報 title セット	
style	行内スタイル情報 style セット	
readonly	読み出し専用指定 readonly セット readonly="readonly" 以外の値はセットできないように 制限を かけます。(大文字/小文字も判別します)	
disabled	選択不能指定 disabled セット disabled="disabled" 以外の値はセットできないように 制限を かけます。(大文字/小文字も判別します)	
tabindex	タブ移動 tabindex セット	
accesskey	アクセスキー accesskey セット	

属性	意味	必須
	組込みイベント onfocus、onblur、onselect、onchange、onclick、ondblclick、 onmousedown、onmouseup、onmouseover、onmousemove、 onmouseout、onkeypress、onkeydown、onkeyup	
clazz	class 属性が使えないための代替。HTML 化したときは class に なります。class セット Java の言語使用上、class で作成できな いため、代用として clazz を使用しています。html で作成され る属性は、class で作成されます。	
msg	messageResource より取り込むキーメッセージを指定します。 メッセージ ID は、所定の language に基づく ResourceManager の getMessage(id) を呼び出し、その結果 を msglbl に登録します。getMsglbl() で取り出せます。ラベル ID とメッセージ ID は同時には登録できません。	
lbl	LabelResource より取り込むキーラベルを指定します。ラベルを 変更するときに、lbl 属性を使います。ラベル ID は、所定の language に基づく ResourceManager の getLabel(id) を呼 び出し、その結果を msglbl に登録します。getMsglbl() で取り 出せます。ラベル ID とメッセージ ID は同時には登録できませ ん。	
language	Resource の言語を指定します。言語コードを指定します。この 言語コードに基づいて、表示のラベルをリソースから作成しま す。	
debug	デバッグ状態にします。 true/false デバッグ情報を 出力する ／しないを指定します。	



ヒント

tesxtarea タグは、複数行のテキスト入力コントロールを生成します。ユーザエージェントはタグの内容を当該コントロールの初期値とする必要があり、最初にこれをレンダリングしなければなりません。

④使用例

例1 テキストエリアを作成します。

```
<mis:textarea name="BUN" />
```

例2 初期値に値をセットしたいとき

```
<mis:textarea name="BUN">
  <jsp:text>test</jsp:text>
</mis:textarea name="BUN">
```

9. mis:query

①概要

形式	<pre> <mis:query> <jsp:text> SQL 文 </jsp:text> </mis:query> <mis:query command="NEW" > <jsptext > SELECT 文 </mis:query> </mis:query> <mis:query command="NEW" names="..." queryType="JDBCErrMsg" > <jsptext > { call PL/SQL(?,?,?,?) } </jsp:text> </mis:query> </pre>
body	あり

②説明

QueryTag は、データベースの検索を行うためのタグです。このタグの内容に SQL 文を記述します。WhereTag、AndTag を使うと実行される SQL 文が異なります(使用例参照)。また、PL/SQL の SP で検索を行うときもこのタグを使います。

③属性

属性	意味	必須
tableId	結果を DBTableModel に書き込んで、session に登録するときのキーです。(通常は使いません)結果を DBTableModel に書き込んで、session に登録するときのキーを指定します。デフォルトは、MISSystem.DB_TABLE_MODEL_KEY です。	
dbid	Query オブジェクトを作成する時の DB 接続 ID(通常は使いません)Query オブジェクトを作成する時の DB 接続 ID を指定します。	
command	処理コマンド(NEW のみ)(オプション)コマンドをセットします。コマンドは、HTML から (get/post) 指定されますので、setCommand()メソッドで 登録します。	
skipRowCount	データの読み始めの初期値を指定 (オプション通常は使いません)データの読み始めの初期値を指定します。データベース自体の検索は、指定された SQL の全件を検索しますが、DBTableModel のデータとしては、スキップ件数分は登録されません。サーバーのメモリ資源と応答時間の確保のためです。	

属性	意味	必須
maxRowCount	データの最大読み込み件数を指定(オプション通常は使いません)データの最大読み込み件数を指定します。データベース自体の検索は、指定された SQL の全件を検索しますが、DBTableModel のデータとして登録する最大件数をこの値に設定します。サーバーのメモリ資源と応答時間の確保のためです。	
queryType	検索を実行する手段を指定します。ストアプロシージャ等を実行する場合に、queryType 属性を使用します。その場合、queryType="JDBCErrMsg" を指定する必要があります。デフォルトは、JDBCQuery です。	
scope	キャッシュする場合の範囲を指定します。 "request","page","session","applicaton" が指定できます。	
debug	デバッグ情報を 出力する/しないを指定します。	
displayMsg	検索結果をの件数や登録された件数を画面上に表示するかどうかを指定します。ここでは、検索結果の件数や登録された件数をまず出力し、その次に、ここで指定したメッセージをリソースから取得して 表示します。表示させたくない場合は、displayMsg = "" をセットしてください。デフォルトは、検索件数を表示します。	
names	PL/SQL を利用する場合の引数にセットすべき データの名称を指定します。複数ある場合は、カンマ区切り文字で渡します。	
stopZero	検索結果が0件のとき処理を続行するかどうかを指定します。デフォルトは、フォルス(続行する)です。	

④使用例

例1 指定のSQL文を実行します。

```
<mis:query command="NEW" >
  <jsp:text>
    select PN,YOBI,NMEN,HINM,RSSNM,MKCD,MKNM,FGSTD,FGKPN,CDSYB,CDKAKU
    from OITMSTRU
    where PN like "{@PN}%"
    order by PN
  </jsp:text>
</mis:query>
```

例2 指定のSQL文を実行します。

```
<mis:query command="NEW">
  <jsp:text>
    select PN,YOBI,NMEN,HINM from XX01 where PN = '{@PN}' order by PN
  </jsp:text>
</mis:query>
```

①検索条件が入力された時({@PN}が NOT NULL のとき)

作成される SQL 文⇒select PN,YOBI,NMEN,HINM from XX01 where PN = 'AAA' order by PN

②検索条件が入力されなかった時(@PN)が NULL のとき

作成される SQL 文⇒select PN,YOBI,NMEN,HINM from XX01 where PN = '' order by PN

```
-----
<mis:query command="NEW">
  <jsp:text>
    select PN,YOBI,NMEN,HINM from XX01
  </jsp:text>
  <mis:where>
    <mis:and value="PN = '{@PN}%" />
    <mis:and value="YOBI like '{@YOBI}%" />
  </mis:where>
  <jsp:text>
    order by PN
  </jsp:text>
</mis:query>
```

①検索条件が入力された時(@PN)が NOT NULL のとき

作成される SQL 文⇒select PN,YOBI,NMEN,HINM from XX01 PN = 'AAA%'
and YOBI like 'BBB%' order by PN

②検索条件が入力されなかった時(@PN)が NULL のとき) WHERE 句がなくなる。

作成される SQL 文⇒select PN,YOBI,NMEN,HINM from XX01 order by PN

注意:WhereTag を使った場合、下のようにはなりません。

select PN,YOBI,NMEN,HINM from XX01 PN = '' and YOBI like '%' order by PN

```
-----
<mis:query command="NEW">
  <jsp:text>
    select PN,YOBI,NMEN,HINM from XX01 where PN="11111"
  </jsp:text>
  <mis:where startKey="and">
    <mis:and value="YOBI like '{@PN}%" />
  </mis:where>
  <jsp:text>
    order by PN
  </jsp:text>
</mis:query>
```

①検索条件が入力された時(@YOBI)が NOT NULL のとき

作成される SQL 文⇒select PN,YOBI,NMEN,HINM from XX01 PN = '11111'
and YOBI like 'BBB%' order by PN

②検索条件が入力されなかった時(@YOBI)が NULL のとき) WHERE 句がなくなる。

作成される SQL 文⇒select PN,YOBI,NMEN,HINM from
XX01 PN = '11111' order by PN

10. mis:view

①概要

形式	<pre><mis:view /> <mis:view viewFormId = "HTMLTableViewForm" command = "{@command}" startNo = "0" pageSize = "20" language = "ja" /></pre> <p>viewFormId: ViewForm オブジェクトを作成する時のキー (オプション) ViewFormFactory 参照</p> <p>command: 処理コマンド (PREV,NEXT,FIRST,LAST,VIEW) (オプション)</p> <p>startNo: データの書き始めの初期値を指定 (オプション)</p> <p>pageSize: データのページ数 (画面表示件数) を指定 (オプション)</p> <p>language: ロケール (オプション)</p>
body	あり

②説明

ViewFormTag は、データベースの検索結果を表示します。設定によりテキストフィールドやテーブルの形で表示したり、入力可/不可、表示可/不可の設定を行うことが可能です。DBTableModel オブジェクトを表示する、共通オブジェクトです。このオブジェクトに、データ (DBTableModel) と、コントローラ (ViewFormTagForm) を与えて、外部からコントロールすることで、各種形式でデータ (DBTableModel) を表示させることが可能です。

ViewFormTag に対して、コマンドを与えることにより、内部のコントローラの実装に対応した、形式でデータを作成します。

各属性は、{@xxxx} 変数が使用できます。これは、ServletRequest から、xxxx をキーに値を取り出し、この変数に割り当てます。つまり、この xxxx をキーにリクエストすれば、この変数に値をセットすることができます。

http://localhost/query.jsp?KEY1=VAL1&KEY2=VAL2

のようなリクエストで、{@KEY}1} とすれば、VAL1 がセットされます。

③属性

属性	意味	必須
viewFormId	ViewForm オブジェクトを作成する時のキー (ViewFormFactory 参照) session から取得する ViewForm オブジェクトの ID。表示処理後に、(内部ポインタを書き換えた) ViewForm オブジェクトを 同じキーで、session に登録します。デフォルトは、MISSystem.VIEWFORM_KEY です。	

属性	意味	必須
command	処理コマンド(PREV,NEXT,FARST,LAST,VIEW)コマンドをセットします。コマンドは、HTML から(get/post)指定されますので、setCommand()メソッドで登録します。	
startNo	データの書き始めの初期値を指定。表示データを作成する場合の表示の開始行番号をセットします。セットされていない場合は、変更しません。	
pageSize	データのページ数(画面表示件数)を指定。表示データを作成する場合の1ページの行数をセットします。セットされていない場合は、変更しません。	
language	ロケール。言語コードを指定します。この言語コードに基づいて、表示のラベルをリソースから作成します。	
tableId	結果をDBTableModelに書き込んで、sessionに登録するときのキーです。デフォルト"hybs_dbTableModel" "session"から取得するDBTableModelオブジェクトのID。表示処理後に、(内部ポインタを書き換えた)DBTableModelオブジェクトを同じキーで、sessionに登録します。デフォルトは、MISSystem.DB_TABLE_MODEL_KEYです。	
columnWritable	書き込み可能カラム名を、カンマ区切りで与えます。 例:"OYA,KO,HJO,SU,DYSET,DYUPD"	
columnDisplay	表示可能カラム名を、カンマ区切りで与えます。 例:"OYA,KO,HJO,SU,DYSET,DYUPD"	
viewFormType ※1	ViewFormオブジェクトを作成するViewFormFactoryに与えるID。これは、mis.pdm.hayabusa.html以下の****ViewFormクラスの****を与えます。これらは、ViewFormのサブクラスです。	
scope	キャッシュする場合のスコープを指定します。 "request","page","session","applicaton"が指定できます。	
noWritable	書き込み不可カラム名を、カンマ区切りで与えます。 例:"OYA,KO,HJO,SU,DYSET,DYUPD" br />	
noDisplay	表示不可カラム名を、カンマ区切りで与えます。 例:"OYA,KO,HJO,SU,DYSET,DYUPD"	
writable	rowWritable(行が書き込み可能かどうか)を設定します。書き込み許可する場合は、true 許可しない場合は、true 以外を設定します。デフォルト値は、ViewForm.DEFAULT_ROW_WRITABLEです。ここで、falseを設定すると全行書き込み禁止になります。trueを設定した場合、デフォルトで書き込み許可が与えられ、先頭カラムがwritableであれば、その値を取り込んで、行ごとの書き込み許可/不許可が決定されます。	
checked	行のチェックボックスの初期値を指定します。書き込み可能な行(rowWritable == true)のチェックボックスに対して 初期値を選択済みにするか、非選択済みにするかを指定します。	
pagePlus	1ページの行数の増加分をセットします。セットされていない場合は、0(増加なし)です。	

属性	意味	必須
rowspan	表示データを作成する場合のフォーマットの行数をセットします。	
skip	NEXT 等で、選択行以外をスキップするのかどうかを指定します。明細情報の登録/編集時に、必要な個所のみを順番に処理する場合に使用します。	
viewLinkId	request から取得する ViewLink に対応する Attributes オブジェクトの ID。ViewForm オブジェクトに、この Attributes オブジェクトを登録することにより、カラムに対して、リンク情報を付加して表示させます。デフォルトは、MISSystem.VIEWLINK_KEY です	
selectedType	表示時の選択用オブジェクトのタイプを指定します。複数選択可能時は "checkbox" を指定します。一つだけ選ばせる場合は、"radio" を指定します。隠しフィールドで全件を選ぶ場合は、"hidden" を指定します。デフォルトは、"checkbox" です。	
optionType Attributes	テーブル等のチェックボックスに属性を付加します。JavaScript などの HTML 基本タグ以外の属性を、そのまま チェックボックス/ラジオボタン等に使用します。そのときに、(')は、(")に置き換えます。	
noMessage	検索結果メッセージを表示する/しないを設定します。デフォルトは、表示する(true 以外)です。	
backLinkCount	ページの先頭へのリンクの間隔をセットします。セットされていない場合は、0(リンクなし)です。	

※1

viewFormType 属性に指定できる値は、第Ⅲ部『第5章 ViewForm クラス』の中から選択できます。

? ヒント

ViewFormTag は、DBTableModel オブジェクトを表示する、共通オブジェクトです。このオブジェクトに、データ(DBTableModel)と、コントローラ(ViewFormTagForm)を与えて、外部からコントロールすることで、各種形式で データ(DBTableModel)を表示させることが可能です。

ViewFormTag に対して、コマンドを与えることにより、内部のコントローラの実装に対応した、形式でデータを作成します。

④使用例

例1

```
<mis:view viewFormId = "HTMLTable" command = "NEW" startNo = "0" pageSize = "5" />
```

例2

```
< mis:view viewFormId="HTMLTable" command="NEW" startNo="0" pageSize="5" / >
```

11. mis:entry

①概要

形式	①<mis:entry command="..."> <mis:setColumncommand="..." /> </mis:entry> ②<mis:entry command="..." /> setColumn を使わない場合でも result.jsp から次画面 (insert,modify,copy.jsp 等)に DBTableModel をもっていく場合には、必ず②を書いてください。(取消のとき、エンジン内で DBTableModel を操作するのに使用するため)
body	あり

②説明

EntryTag は、DBTableModel オブジェクトを操作する、共通オブジェクトです。

③属性

属性	意味	必須
tableId	結果を DBTableModel に書き込んで、session に登録するときのキーです。session から所得する DBTableModel オブジェクトの ID。表示処理後に、(内部ポインタを書き換えた) DBTableModel オブジェクトを 同じキーで、session に登録します。デフォルトは、MISSystem.DB_TABLE_MODEL_KEY です。	
command	処理コマンド INSERT/COPY/MODIFY/DELETE/ENTRY/CHANGE コマンドをセットします。 コマンドは、HTML から (get/post) 指定されますので、setCommand()メソッドで登録します。	○
rows	変更する行数を指定します。(オプション)書き込み可能口ウを、カンマ区切りで与えます。例："1,2,3,5,8"	
scope	処理コマンド。デフォルト値がセットされていないときに、DBCColumnResource.properties ファイルの デフォルト値の値を使うかどうかを指定します。使う場合は、true / 使わない場合は false (デフォルト)をセットします。	

④使用例

```
<mis:entry command = "{@command}" >
  <mis:columnSet command = "{@command}" columnId = "ECNO" action = "CLEAR" />
```

```
<mis:columnSet command = "{@command}" columnId = "JYOKYO" action = "SET" value = "1" />
</mis:entry>
```

```
<mis:entry command = "MODIFY" rows="1" >
  <mis:columnSet command = "MODIFY" columnId = "key" action = "TBLSET" value = "[key][lang]" />
</mis:entry>
```

DEFAULT カラムリソースで定義したデフォルト値をセットします。

CREAR 値をクリア(ゼロストリング "")します。

ADD 現在の値を +1 します。 0 ⇒ 1, A ⇒ B, 9 ⇒ 10

SET value で設定した値を 新しい値として登録します。

LOWER 小文字に変換します。

UPPER 大文字に変換します。

COPY value にコピー元のカラム ID をセットすれば、その値を代入します。

TBLSET DBTableModel の内容を取り込んで指定の columnId カラムに設定します。[カラム名] で指定できます。また、これは文字列を解析して、value を作成しますので、文字列連結等に使用できます。

12. mis:columnSet

①概要

形式	<pre><mis:columnSet command = "... " /> <mis:entry command="..."> <mis:setColumn command="..." /> </mis:entry></pre>
body	なし

②説明

ColumnSet タグは、入力カラムに対して初期値を設定します。
DBTableModel オブジェクトを操作する、共通オブジェクトです。

③属性

属性	意味	必須
command	処理コマンド。上位の EntryTag のコマンドと一致した場合のみ、処理されます。 INSERT/COPY/MODIFY/DELETE	
value	この属性は、当該コントロールの初期値を指定します。 設定する値をセットします。	

属性	意味	必須
action	value に対する処理コマンド CLEAR/COPY/INCREMENT/DECREMENT アクションをセットします。 [DEFAULT/CREAR/ADD/SET/LOWER/UPPER/COPY/TBLSET] DEFAULT カラムリソースで定義したデフォルト値をセット。 CREAR 値をクリア(ゼロストリング "")します。 ADD 現在の値を +1 します。 0 ⇒ 1, A ⇒ B, 9 ⇒ 10 SET value で設定した値を 新しい値として登録します。 LOWER 小文字に変換します。 UPPER 大文字に変換します。 COPY value にコピー元のカラム ID をセットすれば、その値を代入します。 TBLSET DBTableModel の内容を取り込んで指定の columnId カラムに設定します。[カラム名] で指定できます	
columnId	変更するカラムをカラム名で指定します。columnId をセットします。処理を適用するカラムIDを指定します。カラム名とカラムNo は同時には登録できません。	
columnNo	変更するカラムをカラム No で指定します。columnNo セット処理を適用するカラムNoを指定します。カラム名とカラム No は同時には登録できません。	

④使用例

```

<mis:entry command = "{@command}" >
  <mis:columnSet command = "{@command}" columnId = "ECNO" action = "CLEAR" />
  <mis:columnSet command = "{@command}" columnId = "JYOKYO" action = "SET" value = "1" />
</mis:entry>

<mis:entry command = "MODIFY" rows="1" >
  <mis:columnSet command = "MODIFY" columnId = "key"
    action = "TBLSET" value = "[key][lang]" />
</mis:entry>

```

13. mis:update

①概要

形式	<mis:update command="..." />
body	あり

②説明

データベースに追加/更新します。(推奨していません)使用しないでください。データベースに追加/更新するときは、PLSQL を CALL して下さい。⇒PLSQLUpdate タグ参照。

③属性

属性	意味	必須
tableId	結果をDBTableModelに書き込んで、sessionに登録するときのキーを指定します。	
dbid	Queryオブジェクトを作成する時のDB接続IDを指定します。	
command	処理コマンド (INSERT/COPY/MODIFY/DELETE/ENTRY/CHANGE)	
queryType	検索を実行する手段を指定します。ストアプロシージャ等を実行する場合に、queryType属性を使用します。その場合、queryType="JDBCErrMsg"を指定する必要があります。デフォルトは、JDBCQueryです。	
scope	キャッシュする場合の範囲を指定します。 "request","page","session","applicaton"が指定できます。	
debug	デバッグ情報を出力する/しないを指定します。	
displayMsg	検索結果をの件数や登録された件数を画面上に表示するかどうかを指定します。ここでは、検索結果の件数や登録された件数をまず出力し、その次に、ここで指定したメッセージをリソースから取得して表示します。表示させたくない場合は、displayMsg=""をセットしてください。デフォルトは、検索件数を表示します。	
names	PL/SQLを利用する場合の引数にセットすべきデータの名称を指定します。複数ある場合は、カンマ区切り文字で渡します。	

④使用例

14. mis:upload

①概要

形式	<mis:upload fileURL="..." maxPostSize="..." />
body	なし

②説明

FileUploadTag は、MultipartRequest を利用して、ファイルをサーバーにアップロードします。Upload 後の属性は、DBTableModel に格納することにより、通常のデータと同様の取り出し方が可能です。

③属性

属性	意味	必須
maxPostSize	最大ファイル容量。最大転送サイズを指定します。	
language	ロケール。言語コードを指定します。この言語コードに基づいて、表示のラベルをリソースから作成します。	
tableId	結果を DBTableModel に書き込んで、session に登録するときのキーです。デフォルト"hybs_dbTableModel"	
fileURL	ファイル URL を指定します。	

④使用例

15. mis:link

①概要

形式	<code><mis:link href="..."><jsp:text> ... </jsp:index></mis:link></code>
body	あり

②説明

LinkTag はリンクを作成するタグです。HTML の a タグと同等で、国際化対応と引数の受け渡しが可能です。

③属性

属性	意味	必須
name	このタグをリンクの対象とするための名前を指定します。この属性は、他のリンクからの終点となり得るよう、当該アンカーに名前をつけます。この属性値は固有のアンカー名でなければならない。この名前の有効範囲は、当該文書内に限られます。この属性が、id 属性と同じ名前空間を共有することに注意してください。	
href	リンク先の URL を指定します。この属性は、Web リソースの所在を指定します。これによって当該タグ(始点側アンカー)と、この属性が定める終点側アンカーとの間のリンクが定まります。	
hreflang	リンク先の基本となる言語コードを指定します。この属性は、href 属性が指し示すリソースの基本言語を指定します。この属性は、href 属性が指定されている場合しか用いません。	
type	リンク先の MIME タイプを指定します。この属性は、リンク先アドレスで得られるコンテンツの MIME タイプに関する助言を与えます。サポートしていない MIME タイプのコンテンツを取得することになると助言された場合、ユーザエージェントは、コンテンツを取ってくるのではなく代替メカニズムを用いることを選択できます。	
rel	この文書から見た、href 属性で指定されるリンク先との関係を指定します。この属性は、当該文書から href 属性が指定するアンカーに向けての関係性を述べます。この属性の値は、空白区切りで列挙したリンク形式のリストです。	
rev	href 属性で指定されるリンク先から見た、この文書との関係をしてします。この属性は、href 属性が指定するアンカーから当該タグへの逆方向リンクの記述に用います。この属性の値は、空白区切りで列挙したリンク形式のリストです。	
charset	リンク先の文字コードセットを指定します。この属性は、当該リンクが指し示すリソースの文字符号化方法を指定します。詳しくは文字符号化方法の項を参照してください。	

属性	意味	必須
id	文書内識別子 id セット	
lang	言語情報 lang セット	
dir	テキスト方向 dir セット	
title	要素情報 title セット	
style	行内スタイル情報 style セット	
tabindex	タブ移動 tabindex セット	
accesskey	アクセスキー accesskey セット	
shape	リンクボタンの形状(イメージマップ rect/circle/poly/default)を指定します。	
coords	リンクボタンの座標を指定します。coords セット イメージマップ rect circle poly	
target	目標フレーム情報。リンク先の文書を表示させるフレーム、またはウィンドウの名前を指定します。	
	組込みイベント onfocus、onblur、onclick、ondblclick、onmousedown、 onmouseup、onmouseover、onmousemove、onmouseout、 onkeypress、onkeydown、onkeyup	
clazz	class セット。Java の言語使用上、class で作成できないため、代用として clazz を使用しています。html で作成される属性は、class で作成される。HTML 化したときは class になる。	
msg	メッセージを指定します。メッセージ ID は、所定の language に基づく ResourceManager の getMessage(id) を呼び出し、その結果を msglbl に登録します。getMsglbl() で取り出せます。ラベル ID とメッセージ ID は同時には登録できません。messageResource より取り込むキー	
lbl	ラベルを指定します。ラベルを変更するときに、lbl 属性を使います。ラベル ID は、所定の language に基づく ResourceManager の getLabel(id) を呼び出し、その結果を msglbl に登録します。getMsglbl() で取り出せます。ラベル ID とメッセージ ID は同時には登録できません。LabelResource より取り込むキー	
language	言語コードを指定します。この言語コードに基づいて、表示のラベルをリソースから作成します。Resource の言語を指定します。	
debug	デバッグ状態にします。true/false デバッグ情報を 出力する/しないを指定します。	
name	このタグをリンクの対象とするための名前を指定します。この属性は、他のリンクからの終点となり得るよう、当該アンカーに名前をつけます。この属性値は固有のアンカー名でなければならない。この名前の有効範囲は、当該文書内に限られます。この属性が、id 属性と同じ名前空間を共有することに注意してください。	

属性	意味	必須
column	リンクをつけたいカラムを指定します。このカラム名の TableModel に対して、 ViewLink タグが存在する場合は、リンクを張ります。	
keys	リンク先に渡すキーを指定します。	
value	リンク先に渡す値を指定します。	
onLink	リンクを張る/張らないの設定を指定します。	
fileCheck	ファイルの存在チェックを行うかどうかを指定します。“true”の場合は、ファイルが存在チェックを行い、ファイルがあればリンクを張ります。ファイルが存在しない場合は、何も表示されません。デフォルトは、“false”(ファイルチェックを行わない)です。注意:現状、viewLink タグからの使用はできませんので、ご了承ください。	
filePath	アドレスにファイルのディレクトリ直接のアドレスを指定するかどうか設定します。“true”の場合は、ファイルの物理ディレクトリをリンクします。デフォルトは、“false”(ファイルの直接アドレスを行わない)です。	

④使用例

例1

```
<mis:link name="muratec" href="http://www.muratec.co.jp" target="_top"/>
```

例2 普通のリンク

```
<mis:link href="../32ExcelIn/direct.jsp" keys="location,filename,modifyType"
  value="{@GUI.ADDRESS},{@GUI.ADDRESS}.xls,A" target="RESULT" >
  
  <mis:message msg="MSG0057" />
</mis:link>
```

例3 検索結果テーブル内でのリンク

```
<mis:viewLink command="{@command}">
  <mis:link column="KMZ" href="../RK0251/index.jsp" target="CONTENTS"
    keys="KMZ,ORDER_BY,command" value="[KMZ],'LKIS',NEW" />
</mis:viewLink>
```

16. mis:hidden

①概要

形式	<pre><mis:hidden name="..." /> <mis:hidden name="..." value="..." /> <mis:hidden name="..." /><jsp:text>...</jsp:text></mis:hidden></pre>
body	あり

②説明

HiddenTag は、隠しフィールドを作成するタグです。name 属性をキーに value 属性をセットします。

③属性

属性	意味	必須
name	この部品の名前を指定します。名前を指定します。name 属性と names 属性は同時には登録できません。	
value	入力フィールドの初期値を指定します。値を指定します。	
names	名前を CVS 形式で複数指定します。name 属性と names 属性は同時には登録できません。	
vals	names 属性に対応する値を CVS 形式で複数指定します。	

④使用例

例 1 次画面に PN という名前で{@PN}に入っている値を渡します。

```
<input:hidden name="PN" value="{@PN}" />
```

例 2 次画面で使用する PLSQL と引数を渡します。

```
<input:hidden name="name" value="UNIQ,LKIS,KBLOYA,KBLKO,KBNGK,HJO,KBLKNM" />
```

```
<input:hidden name="SQL" value="{ call RKP0011B.RK0011B(?,?,?,?) }" />
```

↓

```
<mis:plsSqlUpdate
  command = "{@command}"
  names   = "{@names}"
  dbType  = "RK0011ARG"
  queryType = "JDBCPLSQL" >
  <jsp:text>{@SQL}</jsp:text>
</mis:plsSqlUpdate>
```

例 3 次画面に SQL という名前で Body に記述した値を渡します。

```
<mis:hidden name="SQL"
```

```
  <jsp:text>
```

```
    SELECT AAA
```

```
FROM RK01
WHERE UNIQ = ?
</jsp:text>
</mis:hidden>
```

17. mis:debug

①概要

形式	<mis:debug debug="true"or"false"><jsp:text> ... </jsp:index></mis:debug>
body	あり

②説明

DebugTag は、デバッグ時にタグで囲まれた内容を画面に表示します。

③属性

属性	意味	必須
debug	デバッグ状態にします。表示(true)/非表示(false) デバッグ情報を 出力する/しないを指定します。	

デバッグ状態で、BODY を書き込む。

④使用例

```
<mis:debug debug="true" >
  <jsp:text>デバッグ時に表示したい内容</jsp:text>
</mis:debug>
```

debug : 出力する ("true")/しない ("false")(オプション)

例1 debug="true"なので、表示される。

```
<mis:debug debug="true"><jsp:text>test</jsp:text></mis:debug>
```

例2 debug="false"なので、表示されない。

```
<mis:debug debug="false"><jsp:text>test</jsp:text></mis:debug>
```

18. mis:equals

①概要

形式	<mis:equals val1="..." val2="..."> ... </mis:equals>
body	あり

②説明

EqualsTag は、引数 val1 と val2 を比較し、等しければタグの内容を表示(実行)します。
 val1 属性 と val2 属性の文字列表現の比較により BODY を表示/非表示を切り替える簡易タグです。
 val1.equals(val2) が 成立すれば、BODY を表示します。
 val1 が null の場合は、無条件で非表示とします。(val1 == val2) はチェックしないというイメージです。

③属性

属性	意味	必須
val1	equals で比較するときの、第 1 引数を指定します。	○
val2	equals で比較するときの、第 2 引数を指定します。	
ignoreCase	比較する場合に大文字/小文字を区別するかどうかを指定します。デフォルトは、区別しない("true")	
startsWith	この文字列が、指定された接頭辞で始まるかどうかを判定します。デフォルトは、判定しない("false")	
nullCheck	null チェックを行うかどうかを指定します。チェックを行うように指定("true")した場合に、第一引数が null の場合は、タグの Body は実行されます。デフォルトは、行わない(true 以外)です。	
notEquals	判定結果を反転させます。通常の成立条件において、不成立の場合に、BODY を実行します。通常の処理結果の正反対の処理を行います。デフォルトは、通常(true 以外)です。	
debug	デバッグ情報を 出力する/しないを指定します。	

? ヒント

EqualsTag は、val1 属性と val2 属性の文字列表現の比較により BODY を表示/非表示を切り替える簡易タグです。
 val1.equals(val2)が成立すれば、BODY を表示します。
 val1 が null の場合は、無条件で非表示とします。(val1==val2)はチェックしないというイメージです。

④使用例

例1 val1=val2 なので、タグの内容が表示される。
 <mis:equals val1="val" val2="val"><jsp:text>test</jsp:text></mis:equals>

例 2 val1 ≠ val2 なので、タグの内容が表示されない。

```
<mis:equals val1="val1" val2="val2"><jsp:text>test</jsp:text></mis:equals>
```

例 3

```
<mis:equals val1="ABCD" val2="{@value}" >
    val1.equals( val2 ) 時に実行したい内容
</mis:equals>
```

例 4

```
<mis:equals val1="{@value}" nullCheck="true" >
    val1 が null の時に実行したい内容
</mis:equals>
```

19. mis:message

①概要

形式	<code><mis:message msg="..." /></code> <code><mis:message lbl="..." /></code>
body	あり

②説明

画面にメッセージを表示させるタグです。msg 属性に MessageResource.properties のキーを与えることで、ロケールにあわせたリソースを使用して、画面に表示します。lbl 属性に LabelResource.properties のキーを与えることで、ロケールにあわせたリソースを使用して、画面に表示します。ロケールは、ユーザー情報の lang 属性をデフォルトで使用し、セットされていない場合は、リクエスト情報のロケールから取得します。

msg 属性に MessageResource.properties 以外の値をセットした場合や、lbl 属性に LabelResource.properties 以外の値をセットした場合は、セットした値が、そのまま出力されます。これは、ロケールが見つからなかった 場合の標準の対応方法です。

msg 属性と lbl 属性を同時に与えた場合は、MISSystemException を throw します。これは、どちらかの属性しか使えないということを意味します。

各属性は、{@xxxx} 変数が使用できます。これは、ServletRequest から、xxxx をキーに値を取り出し、この変数に 割り当てます。つまり、この xxxx をキーにリクエストすれば、この変数に値をセットすることができます。

③属性

属性	意味	必須
msg	メッセージを指定します。メッセージ ID は、所定の language に基づく ResourceManager の getMessage(id) を呼び出し、その結果を msglbl に登録します。getMsglbl() で取り出せません。ラベル ID とメッセージ ID は同時には登録できません。	
lbl	LabelResource より取り込むキーを指定します。ラベルを指定します。ラベルを変更するときに、lbl 属性を使います。ラベル ID は、所定の language に基づく ResourceManager の getLabel(id) を呼び出し、その結果を msglbl に登録します。getMsglbl() で取り出せません。ラベル ID とメッセージ ID は同時には登録できません。	
language	Resource の言語を指定します。言語コードを指定します。この言語コードに基づいて、表示のラベルをリソースから作成します。	
comment	直接メッセージを表示させたい時に指定します。コメントを指定します。msg 属性でメッセージリソースから読み取りますが、開発途中でリソース登録を済ませていない状態でコメントを入れる場合に、直接コメントをかけるようにします。	
val0	メッセージの引数を指定します。	
val1	メッセージの引数を指定します。	
val2	メッセージの引数を指定します。	
val3	メッセージの引数を指定します。	
val4	メッセージの引数を指定します。	
val5	メッセージの引数を指定します。	
val6	メッセージの引数を指定します。	
val7	メッセージの引数を指定します。	
val8	メッセージの引数を指定します。	
val9	メッセージの引数を指定します。	

 ヒント

MessageTag は、画面にメッセージを表示させるタグです。msg 属性に MessageResource.properties のキーを与えることで、ロケールにあわせたリソースを使用して、画面に表示します。

また、lbl 属性に LabelResource.properties のキーを与えることで、ロケールにあわせたリソースを使用して、画面に表示します。ロケールは、session 登録項目の "hybs_lang" をデフォルトで使用し、language 属性で定義されていれば、そちらを優先します。

④使用例

例1

```
<mis:message comment="コメント" />
<mis:message msg="MSG0032" language="ja" />
<mis:message lbl="PN" language="ja" />
```

msg : MessageResource.properties のキー
 lbl : LabelResource.properties のキー
 language : ロケール(オプション)

例2

```
<mis:message msg="MSG0032" ><jsp:text>検索条件</jsp:text></mis:message>
```

例3

```
<mis:message lbl="PN" ><jsp:text>品番</jsp:text></mis:message>
```

BODY 部分は、無視されます。コメント等に使用できます。
 HTML ファイルには、コメント部分は出力されません。

20. mis:writeTable

①概要

形式	<mis:writeTable name="..."><jsp:text> ... </jsp:index></mis:link>
body	なし

②説明

WriteTableTag は、データベースの検索結果をファイルに出力します。DBTableModel オブジェクトを表示する、共通オブジェクトです。このオブジェクトに、データ(DBTableModel)と、コントローラ(WriteTableTagForm)を与えて、外部からコントロールすることで、各種形式でデータ(DBTableModel)を表示させることが可能です。WriteTableTag に対して、コマンドを与えることにより、内部のコントローラの実装に対応した、形式でデータを作成します。

各属性は、{@xxxx} 変数が使用できます。これは、ServletRequest から、xxxx をキーに値を取り出し、この変数に割り当てます。つまり、この xxxx をキーにリクエストすれば、この変数に値をセットすることができます。

http://localhost/query.jsp?KEY1=VAL1&KEY2=VAL2
 のようなリクエストで、{@KEY}1} とすれば、VAL1 がセットされます。

③属性

属性	意味	必須
separator	固定長ファイルを作成する時の項目区切り文字を指定します。可変長ファイルを作成するときの項目区切り文字をセットします。	
headerSequence	DBTableModel の出力順を指定します。DBTableModel の出力順をセットします。Label,Name,Size,Class,Data の各フィールドの頭文字のアルファベットで 出力順を設定します。	
url	ファイルを Web から取り込む時の基準 URL を指定します。	
filename	ファイルを作成するときのファイル名を指定します。	
encode	ファイルを作成するときのファイルエンコーディング名を指定します。“DEFAULT”,“JISAutoDetect”,“JIS”,“EUC_JP”,“MS932”,“SJIS”,“Windows-31J”,“Shift_JIS”	
writerClass ※1	実際に書き出すクラス名 (の略称) を指定します。	
fileAppend	DBTableModel のファイルを追加モードで書き込むかどうかを指定します。	
tableId	結果を DBTableModel に書き込んで、session に登録するときのキーです。session から所得する DBTableModel オブジェクトの ID。表示処理後に、(内部ポインタを書き換えた) DBTableModel オブジェクトを 同じキーで、session に登録します。デフォルトは、MISSystem.DB_TABLE_MODEL_KEY です。	
direct	結果をダイレクトに EXCEL ファイルとして出力するかどうかを指定します。	
langusge	Resource の言語を指定します。言語コードを指定します。この言語コードに基づいて、表示のラベルをリソースから作成します。	
fileURL	fileURL を セットします。	
writerClass	実際に書き出すクラス名 (の略称) をセットします。	
zip	結果をファイルに出力するときに、ZIP で圧縮するかどうかをセットします。デフォルトは、圧縮しないです。	
scope	キャッシュする場合の範囲を指定します。“request”,“page”,“session”,“applicaton” が指定できます。	

※1 writerClass 属性に指定できる値は、第 III 部 -『第 7 章 Writer/Reader クラス』-『1.DBTableWrite クラス』の中から選択できます。

④使用例

21. mis:space

①概要

形式	<mis:space />
body	なし

②説明

SpaceTag は、InputTag を使用する場合の空きカラムを埋めます。テーブル表示で InputTag 等の存在しない個所に挿入して、テーブルの行列の個数を合わせるために使用します。

通常のタグは、ラベルと値という2つの項目を別々のカラムに登録するように <td></td>タグが、2つ続いた形式をしています。一般に、個数は表示上の都合で任意の個数の<td></td>タグを使用したいケースがあります。

その場合、この SpaceTag を使っておけば大量の作成済み画面を修正しなくても このクラスを修正するだけで、すべての空欄のデザインがキープできます。

③属性

属性	意味	必須
labelWidth	ラベル部分(前半部分)の大きさを指定します。属性は、width 属性に設定されます。	
valueWidth	値部分(後半部分)の大きさを指定します。属性は、width 属性に設定されます。	

? ヒント

input タグを利用する場合、ラベルとオブジェクトを<td>タグを用いてテーブル形式で表示してレイアウトを整えているため、テーブルの空き項目の個所に通常の<td></td>とすると、順番が狂ってきます。

この space タグは、input タグと共に、スペース(空き)領域を作成します。

具体的には、<td></td><td></td> と、2個分の領域を確保します。

将来的に、input タグが、<td>1個分になったり、3個分に拡張された場合でも、この space タグを使用しておけば、画面レイアウトの変更を行う必要はありません。

④使用例

例1 space タグを使用しない場合

```
<table>
  <tr>
    <mis:column name="1-1" />
    <mis:column name="1-2" />
    <mis:column name="1-3" />
  </tr>
  <tr>
    <mis:column name="2-1" />
    <td></td><td></td>
    <mis:column name="2-3" />
  </tr>
</table>
```

例2 space タグを使用した場合

```
<table>
  <tr>
    <mis:column name="1-1" />
    <mis:column name="1-2" />
    <mis:column name="1-3" />
  </tr>
  <tr>
    <mis:column name="2-1" />
    <mis:space />
    <mis:column name="2-3" />
  </tr>
</table>
```

22. mis:plsSqlUpdate

①概要

形式	<pre> <mis:plsSqlUpdate command="..." names="..." dbType="..." queryType="JDBCPLSQL" > <jsp:text> {call plsSql(?,?,?,?)} </jsp:text> </mis:plsSqlUpdate> </pre>
body	あり

②説明

PLSQL を CALL してデータベースにアクセスします。DBTableModel 内のデータを 配列で PL/SQL に渡して DB 登録します。

③属性

属性	意味	必須
tableId	結果を DBTableModel に書き込んで、session に登録するときのキーです。(通常は使いません)結果を DBTableModel に書き込んで、session に登録するときのキーを指定します。デフォルトは、MISSystem.DB_TABLE_MODEL_KEY です。	
queryType	検索を実行する手段を指定します。ストアプロシージャ等を実行する場合に、queryType 属性を使用します。その場合、queryType="JDBCErrMsg" を指定する必要があります。デフォルトは、JDBCQuery です。	
dbid	Query オブジェクトを作成する時の DB 接続 ID(通常は使いません)Query オブジェクトを作成する時の DB 接続 ID を指定します。	
scope	キャッシュする場合の範囲を指定します。"request", "page", "session", "applicaton" が指定できます。	
command	処理コマンド(NEW のみ)(オプション)コマンドをセットします。コマンドは、HTML から(get/post)指定されますので、setCommand()メソッドで 登録します。	
maxRowCount	データの最大読み込み件数を指定(オプション通常は使いません)データの最大読み込み件数を指定します。データベース自体の検索は、指定された SQL の全件を検索しますが、DBTableModel のデータとして登録する最大件数をこの値に設定します。サーバーのメモリ資源と応答時間の確保のためです。	

属性	意味	必須
skipRowCount	データの読み始めの初期値を指定 (オプション通常は使いません) データの読み始めの初期値を指定します。データベース自体の検索は、指定された SQL の全件を検索しますが、DBTableModel のデータとしては、スキップ件数分は登録されません。サーバーのメモリ資源と応答時間の確保のためです。	
debug	デバッグ情報を 出力する/しないを指定します。	
displayMsg	検索結果をの件数や登録された件数を画面上に表示するかどうかを指定します。ここでは、検索結果の件数や登録された件数をまず出力し、その次に、ここで指定したメッセージをリソースから取得して 表示します。表示させたくない場合は、displayMsg = "" をセットしてください。デフォルトは、検索件数を表示します。	
names	PL/SQL を利用する場合の引数にセットすべき データの名称を指定します。複数ある場合は、カンマ区切り文字で渡します。	
dbType	Query オブジェクトに渡す引数のタイプ定義。ここでは、type 定義の PL/SQL 名を指定します。行を表す配列は、type 名 ARRAY という名称です。	
selectedAll	全てのデータを選択済データとして扱って処理します。全件処理する場合に指定します。(true/false)デフォルトは false です。	

④使用例

例 1 引数/プロシジャーを他の JSP から渡す場合

【copy.jsp】

```
<mis:hidden name="names" value="UNIQ,USRID,ECNO,EDBN" />
<mis:hidden name="SQL" value="{ call RKP0271E.RK0271E(?,?,?,?)}" />
```

【entry.jsp】

```
<mis:plsqliUpdate
  command      = "{@command}"
  names        = "{@names}"→PL/SQL に渡す引数(配列)のカラム名
  dbType       = "RK0271ARG"→PL/SQL に渡す引数(配列)の定義ファイル名
  queryType    = "JDBCPLSQL" >
  <jsp:text>
    {@SQL}CALL する PL/SQL
  </jsp:text>
</mis:plsqliUpdate>
```

例 2 引数/プロシジャーを直接書く場合

【entry.jsp】

```
<mis:plsqliUpdate
  command      = "{@command}"
  names        = "UNIQ,USRID,ECNO,EDBN"→PL/SQL に渡す引数(配列)のカラム名
  dbType       = "RK0271ARG"→PL/SQL に渡す引数(配列)の定義ファイル名
  queryType    = "JDBCPLSQL" >
  <jsp:text>
```

```
{ call RKP0271E.RK0271E(?,?,?,?) }→CALL する PL/SQL  
</jsp:text>  
</mis:plsqliUpdate>
```

ヒント

①RKP0271E.RK0271E(?,?,?,?)の「?」の意味

```
(RKP0271E.spc)  
CREATE OR REPLACE PACKAGE RKP0271E AS  
PROCEDURE RK0271E(  
  P_KEKKAOUTNUMBER-- 1 個目の「?」⇒結果 0:正常 1:警告 2:異常  
,P_ERRMSGOUTERR_MSG_ARRAY-- 2 個目の「?」⇒エラーメッセージ配列  
,P_NAMES INVARCHAR2-- 3 個目の「?」⇒カラム名チェック用文字列  
,P_SYSARGSINSYSARG_ARRAY-- 4 個目の「?」⇒登録条件配列  
                                (改廃(A:追加/C:変更/D:削除)等がセットされます)  
,P_RK0271INRK0271ARG_ARRAY-- 5 個目の「?」⇒登録データ配列
```

②RK0271ARG の定義の仕方

```
(RK0271ARG.sql)  
DROP TYPE RK0271ARG_ARRAY;  
CREATE OR REPLACE TYPE RK0271ARG AS OBJECT  
(  
  UNIQVARCHAR2(11)  
,USRIDVARCHAR2(5)  
,ECNOVARCHAR(7)  
,EDBNVARCHAR(2)  
) ;  
/  
CREATE OR REPLACE TYPE RK0271ARG_ARRAY AS VARRAY(100) OF RK0271ARG;  
/
```


23. mis:mailSender

①概要

形式	<mis: mailSender from="..." to="..." subject="..." />
body	あり

②説明

メールを送信するための簡易タグです。

③属性

属性	意味	必須
charset		
from	送信元 (FROM) アドレスを指定します。	
to	送信先 (TO) アドレスを指定します。	
cc	送信先 (CC) アドレスを指定します。	
bcc	送信先 (BCC) アドレスを指定します。	
subject	タイトルを指定します。	
host	メールサーバーを指定します。	
filename	添付ファイル名を指定します。複数ファイルをセットできます。	
debug	デバッグ情報の表示を行うかどうかを指定します。ture:表示 /false:非表示	

④使用例

```
<mis: mailSender from="C00000" to="C99999" subject="{@SUBJECT}" />
```

charset = キャラクタセット (例:iso-2022-jp)
 from = 送信元
 to = To 送信先(複数登録可能)
 cc = Cc 送信先(複数登録可能)
 bcc = Bcc 送信先(複数登録可能)
 host = SMTPサーバー名
 subject = タイトル
 filename = 添付ファイル名(ローカルにセーブされたファイル名)(複数登録可能)
 message = 内容(BODY 部分に書いても良い、ただし、どちらか一方のみ)
 debug = true/false;

24. mis:mailReceiver

①概要

形式	<mis: mailReciver user="..." password="..." command="..." />
body	なし

②説明

メールを受信するための簡易タグです。

③属性

属性	意味	必須
user	受信ユーザーを指定します。	
password	受信パスワードを指定します。	
columnList	項目リストを指定します。FROM,TO,ALLHEADERLINES,...の形式で 項目をカンマで区切って入力します。	
host	メールサーバーを指定します。	
tableId	生成した DBTableModel オブジェクトを session に登録する時の ID。デフォルトは、MISSystem.DB_TABLE_MODEL_KEY です。	
debug	デバッグ情報の表示を行うかどうかを指定します。ture:表示 /false:非表示。	
command	処理コマンド。コマンドは、HTML から(get/post)指定されますので、setCommand()メソッドで 登録します。	

④使用例

```
<mis: mailReciver user="C00000" password="C99999"
    columnList = "MESSAGENUMBER,SUBJECT,FROM,TO,SENTDATE" />
user          = ユーザー名
password      = パスワード
columnList    = 項目リスト
host          = POP3サーバー名
debug         = true/false;
```

25. mis:writeCheck

①概要

形式	<mis:writeCheck />
body	あり

②説明

指定された値が 設定されている場合のみ表示されるタグです。

SqlAndTag と類似の特性をもっています。逆にいえば、このタグの特殊ケースが SqlAndTag だといえます。

③属性

④使用例

```
<mis:writeCheck>
  <mis:input type="submit" name="test" />
</mis:writeCheck>
```

26. mis:appear

①概要

形式	<mis:appear startKey="..." value="..." />
body	なし

②説明

指定された値が 設定されている場合のみ表示されるタグです。SqlAndTag と類似の特性をもっています。逆にいえば、このタグの特殊ケースが SqlAndTag だといえます。

③属性

属性	意味	必須
startKey	固定値の開始文字列を設定します。このキーは、バリューと接続される場合に空白文字を一つ挿入します。この値は、固定値のみ設定可能です。{@xxx}文字は使用できません。	
value	値をセットします。指定された値が 設定されている場合のみ表示されます。これは、一般に value 値が変動する場合に、defaultVal 等に重複する値を 設定したくない場合に使用します。{@xxx}文字が使用できます。	○

属性	意味	必須
defaultVal	初期値を設定します。 value 値が NULL の場合に、この初期値が表示されます。	

④使用例

```

<mis:query command="{@command}" >
  <jsp:text>
    SELECTYOBI,NMEN,NMJP,SIYO1,SIYO2,SIYO3,SIYO4,SIYO5,SIYO6
    FROMRK18
    WHEREFGJ IN ('0','1')
  </jsp:text>
  <mis:where startKey="AND">
    <mis:and value="YOBI LIKE '{@YOBI}%" />
    <mis:and value="NMEN LIKE '{@NMEN}%" />
    <mis:and value="NMJP LIKE '{@NMJP}%" />
  </mis:where>
  <mis:appear startKey = "order by" value = "{@ORDER_BY}" defaultVal = "YOBI" />
</mis:query>

```

27. mis:readTable

①概要

形式	<mis:readTable ... />
body	なし

②説明

DBTableModel オブジェクトを表示する、共通オブジェクトです。

このオブジェクトに、データ(DBTableModel)と、コントローラ(ReadTableTagForm)を与えて、外部からコントロールすることで、各種形式でデータ(DBTableModel)を表示させることが可能です。

ReadTableTag に対して、コマンドを与えることにより、内部のコントローラの実装に対応した、形式でデータを作成します。

各属性は、{@xxxx} 変数が使用できます。これは、ServletRequest から、xxxx をキーに値を取り出し、この変数に割り当てます。つまり、この xxxx をキーにリクエストすれば、この変数に値をセットすることができます。

http://localhost/query.jsp?KEY1=VAL1&KEY2=VAL2
 のようなリクエストで、{@KEY}1} とすれば、VAL1 がセットされます。

③属性

属性	意味	必須
separator	可変長ファイルを作成するときの項目区切り文字をセットします。	
fileURL	fileURL を セットします。	
filename	ファイルを作成するときのファイル名をセットします。	
encode	ファイルを作成するときのファイルエンコーディング名をセットします。“DEFAULT”, “JISAutoDetect”, “JIS”, “EUC_JP”, “MS932”, “SJIS”, “Windows-31J”, “Shift_JIS”	
readerClass ※1	実際に読み出すクラス名 (の略称) をセットします。	
maxRowCount	読取時の最大取り込み件数をセットします。DBTableModel のデータとして登録する最大件数をこの値に設定します。サーバーのメモリ資源と応答時間の確保のためです。	
language	言語コードを指定します。この言語コードに基づいて、表示のラベルをリソースから作成します。	
tableId	session から所得する DBTableModel オブジェクトの ID。表示処理後に、(内部ポインタを書き換えた) DBTableModel オブジェクトを 同じキーで、session に登録します。デフォルトは、MISSystem.DB_TABLE_MODEL_KEY です。	
command	処理コマンドコマンドをセットします。コマンドは、HTML から (get/post)指定されますので、setCommand()メソッドで 登録します。	
modifyType	ファイル取り込み時の モディファイタイプ (A,C,D 属性) を指定します。ファイル読み込み時に、そのデータを A(追加)、C(更新)、D(削除)の モディファイタイプをつけた状態にします。その状態で、そのまま、update することが可能になります。	
displayMsg	query の結果を画面上に表示するかどうかを指定します。ここでは、検索結果の件数や登録された件数をまず出力し、その次に、ここで指定したメッセージをリソースから取得して 表示します。表示させたくない場合は、displayMsg = "" をセットしてください。デフォルトは、検索件数を表示します。	

※1 reader 属性に指定できる値は、第 III 部—『第 7 章 Writer/Reader クラス』—『2. DBTableReader クラス』の中から選択できます。

④使用例

28. mis:where

①概要

形式	<pre><mis:where startKey="..." > <mis:and ... /> <mis:and ... /> </mis:where></pre>
body	あり

②説明

SqlAndTag で指定された条件を元に、WHERE 句を作成します。

各属性は、{@xxxx} 変数が使用できます。これは、ServletRequest から、xxxx をキーに値を取り出し、この変数に割り当てます。つまり、この xxxx をキーにリクエストすれば、この変数に値をセットすることができます。

③属性

属性	意味	必須
startKey	SQL 条件句の最初の演算子を指定します。	

④使用例

```
<mis:query command="NEW">
  <jsp:text>
    select PN,YOBI,NMEN,HINM from XX01 where PN = '{@PN}' order by PN
  </jsp:text>
</mis:query>
```

①検索条件が入力された時(@PN)が NOT NULL のとき

作成される SQL 文⇒select PN,YOBI,NMEN,HINM from XX01 where PN = 'AAA' order by PN

②検索条件が入力されなかった時(@PN)が NULL のとき

作成される SQL 文⇒select PN,YOBI,NMEN,HINM from XX01 where PN = '' order by PN

```
<mis:query command="NEW">
  <jsp:text>
    select PN,YOBI,NMEN,HINM from XX01
  </jsp:text>
  <mis:where>
    <mis:and value="PN = '{@PN}%" />
    <mis:and value="YOBI like '{@YOBI}%" />
  </mis:where>
  <jsp:text>
```

```

        order by PN
    </jsp:text>
</mis:query>

```

①検索条件が入力された時(@PN)が NOT NULL のとき

作成される SQL 文⇒select PN,YOBI,NMEN,HINM from XX01 PN = 'AAA%' and YOBI like 'BBB%' order by PN

②検索条件が入力されなかった時(@PN)が NULL のとき) WHERE 句がなくなる。

作成される SQL 文⇒select PN,YOBI,NMEN,HINM from XX01 order by PN

注意:WhereTag を使った場合、下のようにはなりません。

select PN,YOBI,NMEN,HINM from XX01 PN = '' and YOBI like '%' order by PN

```

<mis:query command="NEW">
    <jsp:text>
        select PN,YOBI,NMEN,HINM from XX01 where PN="11111"
    </jsp:text>
    <mis:where startKey="and">
        <mis:and value="YOBI like '{@PN}%" />
    </mis:where>
    <jsp:text>
        order by PN
    </jsp:text>
</mis:query>

```

①検索条件が入力された時(@YOBI)が NOT NULL のとき

作成される SQL 文⇒select PN,YOBI,NMEN,HINM from XX01 PN = '11111' and YOBI like 'BBB%' order by PN

②検索条件が入力されなかった時(@YOBI)が NULL のとき) WHERE 句がなくなる。

作成される SQL 文⇒select PN,YOBI,NMEN,HINM from XX01 PN = '11111' order by PN

29. mis:and

①概要

形式	<mis:and startKey="..." value="..." multi="...,...,..." />
body	なし

②説明

Where 句を作成するための条件を指定します。
 各属性は、{@xxxx} 変数を使用できます。これは、ServletRequest から、xxxx をキーに値を取り出し、この変数に 割り当てます。つまり、この xxxx をキーにリクエストすれば、この変数に値をセットすることができます。

③属性

属性	意味	必須
startKey	SQL 条件句の最初の演算子を指定します。	
value	value セット	○
multi	複数の引数に対して処理するかどうかを設定します。引数 {@xxxx} が、複数表れた場合に、'xx1','xx2','xx3', ... という形式に変換します。where 条件で言うと、"where PN in ({@PN})" という文字列に対して、"where PN in ('xx1','xx2','xx3')" を作成することになります。デフォルトは、false (マルチ変換しない) です。	

④使用例

```
<mis:query command="NEW">
  <jsp:text>
    select PN,YOBI,NMEN,HINM from XX01 where PN = '{@PN}' order by PN
  </jsp:text>
</mis:query>
```

- ①検索条件が入力された時(@PN)が NOT NULL のとき
 作成される SQL 文⇒select PN,YOBI,NMEN,HINM from XX01 where PN = 'AAA' order by PN
- ②検索条件が入力されなかった時(@PN)が NULL のとき
 作成される SQL 文⇒select PN,YOBI,NMEN,HINM from XX01 where PN = '' order by PN

```
<mis:query command="NEW">
  <jsp:text>
    select PN,YOBI,NMEN,HINM from XX01
  </jsp:text>
  <mis:where>
```



```

    <mis:and value="PN = '{@PN}%" />
    <mis:and value="YOBI like '{@YOBI}%" />
  </mis:where>
  <jsp:text>
    order by PN
  </jsp:text>
</mis:query>

```

①検索条件が入力された時(@PN)が NOT NULL のとき

作成される SQL 文⇒select PN,YOBI,NMEN,HINM from XX01 PN = 'AAA%' and YOBI like 'BBB%' order by PN

②検索条件が入力されなかった時(@PN)が NULL のとき) WHERE 句がなくなる。

作成される SQL 文⇒select PN,YOBI,NMEN,HINM from XX01 order by PN

注意:WhereTag を使った場合、下のようにはなりません。

select PN,YOBI,NMEN,HINM from XX01 PN = '' and YOBI like '%' order by PN

```

<mis:query command="NEW">
  <jsp:text>
    select PN,YOBI,NMEN,HINM from XX01 where PN="11111"
  </jsp:text>
  <mis:where startKey="and">
    <mis:and value="YOBI like '{@PN}%" />
  </mis:where>
  <jsp:text>
    order by PN
  </jsp:text>
</mis:query>

```

①検索条件が入力された時(@YOBI)が NOT NULL のとき

作成される SQL 文⇒select PN,YOBI,NMEN,HINM from XX01 PN = '11111' and YOBI like 'BBB%' order by PN

②検索条件が入力されなかった時(@YOBI)が NULL のとき) WHERE 句がなくなる。

作成される SQL 文⇒select PN,YOBI,NMEN,HINM from XX01 PN = '11111' order by PN

30. mis:viewLink

①概要

形式	<pre><mis:viewLink command="..." > <mis:link name="..." ... /> </mis:viewLink ></pre>
body	あり

②説明

検索結果に対してリンクを表示するときに使用します。DBTableModel オブジェクトを表示する、共通オブジェクトです。このオブジェクトに、データ(DBTableModel)と、コントローラ(EntryTagForm)を与えて、外部からコントロールすることで、各種形式でデータ(DBTableModel)を表示させることが可能です。ViewLinkTag に対して、コマンドを与えることにより、内部のコントローラの実装に対応した、形式でデータを作成します。

各属性は、{@xxxx} 変数が使用できます。これは、ServletRequest から、xxxx をキーに値を取り出し、この変数に割り当てます。つまり、この xxxx をキーにリクエストすれば、この変数に値をセットすることができます。

http://localhost/query.jsp?KEY1=VAL1&KEY2=VAL2
 のようなリクエストで、{@KEY}1} とすれば、VAL1 がセットされます。

③属性

属性	意味	必須
command	処理コマンドコマンドをセットします。コマンドは、HTML から (get/post) 指定されますので、setCommand()メソッドで登録します。	○
viewLinkId	request から取得する ViewLink に対応する Attributes オブジェクトの ID。ViewLink オブジェクトをこのキーで、登録することにより、ViewForm オブジェクトで、リンク情報を付加して表示させる。デフォルトは、MISSystem.VIEWLINK_KEY です。	

④使用例

QueryTag と ViewTag の間に viewLink を使います。

```
<mis:viewLink command="{@command}">
  <mis:link column="NOSYN"
    href="../01PATTERN1/query.jsp" target="QUERY"
    keys="NOSYN,NMSYN,CDBK" value="[NOSYN],[NMSYN],[CDBK]" />
  <mis:link column="NMSYN"
    href="../01PATTERN1/query.jsp" target="QUERY"
    keys="NMSYN" value="[NMSYN]" />
  <mis:link column="CDBK"
    href="../01PATTERN1/query.jsp" target="QUERY"
    keys="NOSYN,NMSYN" value="[NOSYN],[NMSYN]" />
</mis:viewLink>
```

31. mis:columnCheck

①概要

形式	<mis:columnCheck command = "..."/> />
body	なし

②説明

登録データの物理チェックを行うタグです。

③属性

属性	意味	必須
command	処理コマンド。コマンドをセットします。コマンドは、HTML から (get/post)指定されますので、setCommand()メソッドで 登録します。	
columns	チェックすべきカラム列をカンマ区切りで指定します。 columns = "AAA,BBB,CCC,DDD"	
nullCheck	NULL チェックすべきカラム列をカンマ区切りで指定します。 nullCheck = "AAA,BBB,CCC,DDD"	
checkType	チェック対象のデータを指定します。(AUTO/NEW/ENTRY) 通常のリクエストデータは、キーそのものですが、エン트리データは表形式の データを一括で登録するため、(キー+"_" + 行番号)形式のため、バラす必要があります。ENTRY は、DBTableModel のリクエスト情報をチェックします。これは、全データが対照になります。(columns/nullCheck 属性は無効) AUTO は、command が、上記 NEW か ENTRY かを判断して処理を振り分けます。 デフォルトは、AUTO です。	
maxRowCount	ENTRY 時に選択された行数の最大値を設定します。最大選択件数を超えた場合は、エラーメッセージを返します。	
minRowCount	ENTRY 時に選択された行数の最小値を設定します。最小選択件数を超えない場合は、エラーメッセージを返します。	

④使用例

```
< mis:columnCheck
  command = "{@command}"
  columns = "AAA,BBB,CCC"      DB 定義(DBColumn リソース)で定義した項目
                                (桁数,タイプ等)でチェックします。
  nullCheck = "AAA,CCC"      NULL チェックを実行します。
/>
```

32. mis:userInfo

①概要

形式	<mis:userInfo ... />
body	あり

②説明

UserInfo オブジェクトに所定のキーで文字列を設定させるタグです。
 設定した値は、{@USER.xxxx} 形式で 取り出すことができます。
 また、command 属性で 直接画面に値を書き出すことも可能です。

以下の値は UserInfo オブジェクトの項目から取得します。

USER.JNAME ユーザー日本語名称
 USER.ENAME ユーザー英語名称
 USER.ID ユーザーID
 USER.PASSWD ユーザーパスワード
 USER.IDNO USER.ID が5Byte 以上の時のみ先頭1文字を除いたユーザーID
 USER.INFO ユーザー情報(ユーザーID:日本語名称)
 USER.MAILTO メールアドレス
 USER.MAILUSERID メールユーザー名
 USER.MAILPASSWD メールパスワード
 USER.LANG 言語
 USER.ROLE ロール
 USER.GROUP ユーザーグループ
 USER.PROJECT プロジェクト
 USER.IPADDRESS IP アドレス
 USER.LOGINTIME ログイン時刻 以下の値はあらかじめ、動的に作成されます。

USER.YMD 8byte の今日のシステム日付

USER.YMDH 14byte の今日のシステム日時 それ以外は、外部より設定された値です。

③属性

属性	意味	必須
command	処理コマンド。コマンドをセットします。コマンドは、HTML から (get/post)指定されますので、setCommand()メソッドで 登録します。	
key	UserInfo に登録するキーをセットします。	
value	UserInfo に登録する値をセットします。	
dbid	(通常は使いません)Query オブジェクトを作成する時の DB 接続 ID を指定します。	

属性	意味	必須
query	推奨されていません。Query を実行して、value に値をセットするかどうかを指定します。デフォルトは、実行しない ("false")	
nullSet	value が NULL の時に、設定するかどうかを指定します。true の場合は、null のときでもセットします。false の場合は、null のときは、既存の値を置き換えません。デフォルトは、null のときでもセットするです。 ("true")	

④使用例

33. mis:frame

①概要

形式	<mis:frame src="..." name="..." />
body	なし

②説明

フレームを作成する簡易タグです。HTML の frame タグと同等で、国際化対応と引数の受け渡しが可能です。

③属性

属性	意味	必須
src	フレームに表示するソースファイルを指定します。	○
name	フレーム名を指定します。	○
longdesc	Longdesc を指定します。	
marginwidth	フレームの左右余白サイズを指定します。	
marginheight	フレームの上下余白サイズを指定します。	
noresize	フレームサイズを変更できないように指定します。	
scrolling	スクロールバーの表示/非表示を指定します。auto:必要に応じてスクロールバーを表示(デフォルト)yes:常にスクロールバーを表示 no:常にスクロールバーを表示しない	
frameborder	フレームの境界線の表示/非表示を指定します。0:枠を表示しない 1:枠を表示します。(デフォルト)	

属性	意味	必須
changeSrc	推奨されていません。changeGamen タグで、src の指定先の入れ替えを、代替してください。ソース指定を変更します。(廃止)src 指定がデフォルト設定で、changeSrc 属性が設定されている 場合には、この値が優先されます。changeSrc は、完全に指定されている場合のみ、有効になります。例えば、引数 {@xxxx}などを指定しておき、そこに何らかの値が設定 された場合のみ、こちらが使用されます。この属性は、changeGamen 属性と同時に使用できません。	
keys	引数を指定します。URL の引数にセットするキーを カンマ区切りでセットします。	
id	id セット。	
clazz	class セット Java の言語使用上、class で作成できないため、代用として clazz を使用している。html で作成される属性は、class で作成される。	
title	title セット	
style	style セット	
dbTable	前ページで選択したデータ列の情報を次のページに渡すかどうかを指定します。ただし、1行分のデータのみです。(複数選択時は、最初の1件目)yes:渡す no:渡さない。(デフォルト)	

④使用例

```

<frameset>
  <mis:frame marginheight="2" marginwidth="2"
    src="../../RK0310/query.jsp" name="QUERY" keys="PN" dbTable="true" />
  <mis:frame marginheight="2" marginwidth="2"
    src="../../RK0310/forward.jsp" name="RESULT" keys="PN" dbTable="true" />
</frameset>

```

34. mis:text

①概要

形式	<mis:text>・・・</mis:text>
body	あり

②説明

BODY 部に記述された文字列を、ISO8859-1 コードから、ユーザー情報のエンコーディングに変換します。また、request 引数{@xxxx}も、変換されます。これは、XML 形式で日本語コードのパーサが、JSP エンジン (Tomcat) でサポートされるまでの、暫定的なタグです。なお、このタグの内部に存在するカスタムタグは、先に実行されるため漢字コードは、変換されません。

③属性

属性	意味	必須
value	value 値に直接書かれた漢字コードを Shift_JIS に変換します。ここで、value に設定した場合は、BODY 部は無視されます。 <mis:textvalue="あいうえお"/>	

④使用例

例1

```
<mis:text>
  <p>あいうえお:<inputname="PN"value="{@PN}"/></p>
</mis:text>
```

例2

```
<mis:textvalue="あいうえお"/>
```


35. mis:help

①概要

形式	<mis:help msg="HELP" guiInfokey="..." />
body	なし

②説明

③属性

属性	意味	必須
guiInfoKey	GUIInfo のキーを指定します。	○
extension	拡張子を指定します。なにも設定されていない場合は、“html”がデフォルトとなります。ここでは、ピリオドは、含める必要はありません。	
msg	メッセージを指定します。メッセージ ID は、所定の language に基づく ResourceManager の getMessage(id) を呼び出し、その結果を msglbl に登録します。getMsglbl() で取り出せません。ラベル ID とメッセージ ID は同時には登録できません。	
fileCheck	推奨されていません。ファイルの存在チェックを行うかどうかを指定します。この属性のセットの有無に関わらず、ファイルの存在チェックが行われます。この属性は廃止されました。互換性の維持のために残していますが、時期バージョンで、削除されます。	
target	TARGET 属性を指定します。デフォルトは、“_new”として、新規に画面を立ち上げます。CONTENTS 等を指定すれば、コンテンツフレーム(メニューの右側全面)に、RESULT を指定すれば、リザルトフレーム(メニュー右下側)に表示します。	
viewMsg	メッセージを常時表示させるかどうかを指定します。“true”の場合は、常時表示させます。ファイルが、存在した場合は、リンクが張られ、存在しない場合は、リンクが張られません。“false”の場合は、ファイルが、存在した場合は、リンクが張られ、存在しない場合は、なにも表示されません。デフォルトは、“false”(メッセージを常時表示しない)です。	

④使用例

```
<mis:help msg="HELP" guiInfoKey="{@GUI.KEY}" />
```

36. mis:entryQuery

①概要

形式	<mis:entryQuery…>…</mis:entryQuery>
body	あり

②説明

EntryQueryTag は、DBTableModel オブジェクトを作成する、共通オブジェクトです。このオブジェクトに、dbid を与えることにより、dbid に対応した Query オブジェクト(のサブクラスのオブジェクト)が作成されます。

各属性は、{@xxxx} 変数が使用できます。これは、ServletRequest から、xxxx をキーに値を取り出し、この変数に 割り当てます。つまり、この xxxx をキーにリクエストすれば、この変数に値をセットすることができます。

http://localhost/query.jsp?KEY1=VLA1&KEY2=VAL2
 のようなリクエストで、{@KEY}1} とすれば、VAL1 がセットされます。

③属性

属性	意味	必須
tableId	(通常は使いません)結果を DBTableModel に書き込んで、session に登録するときのキーを指定します。デフォルトは、MISSystem.DB_TABLE_MODEL_KEY です。	
queryType		
dbid	通常は使いません)Query オブジェクトを作成する時の DB 接続 ID を指定します。Query オブジェクトを作成する時の DB 接続 ID を指定します。	
scope	キャッシュする場合の範囲を指定します。 "request""page""session""applicaton" が指定できます。	
command	処理コマンドコマンドをセットします。コマンドは、HTML から (get/post)指定されますので、setCommand()メソッドで 登録します。	
maxRowCount	通常は使いません。データの最大読み込み件数を指定します。データベース自体の検索は、指定された SQL の全件を検索しますが、DBTableModel のデータとして登録する最大件数をこの値に設定します。サーバーのメモリ資源と応答時間の確保のためです。	
skipRowCount	通常は使いません。データの読み始めの初期値を指定します。データベース自体の検索は、指定された SQL の全件を検索しますが、DBTableModel のデータとしては、スキップ件数分は登録されません。サーバーのメモリ資源と応答時間の確保のためです。	
debug	デバッグ情報を 出力する/しないを指定します。	

属性	意味	必須
displayMsg	検索結果をの件数や登録された件数を画面上に表示するかどうかを指定します。ここでは、検索結果の件数や登録された件数をまず出力し、その次に、ここで指定したメッセージをリソースから取得して表示します。表示させたくない場合は、displayMsg = "" をセットしてください。デフォルトは、検索件数を表示します	
names	PL/SQL を利用する場合の引数にセットすべき データの名称を指定します。複数ある場合は、カンマ区切り文字で渡します。	

④使用例

例 1

```
<mis:query
  dbid      = "JDBCQuery"
  command   = "{@command}" >
<jsp:text>
  select * from foo where bar like '{@BAR}%'
</jsp:text>
</mis:query>
```

dbid : Query オブジェクトを作成する時のキー(オプション)
QueryFactory 参照

command : 処理コマンド(NEW のみ)(オプション)

skipRowCount : データの読み始めの初期値を指定(オプション)

maxRowCount : データの最大読み込み件数を指定(オプション)

例2

```
<table><tr><mis:input name="PN" value="{@PN}" /></tr>
<tr><mis:input name="CD" value="{@CD}" /></tr>
</table>
```

例3

```
<table><tr><mis:input name="PN" value="{@PN}" >部品入力フィールド</mis:input></tr>
<tr><mis:input name="CD" value="{@CD}" >コードフィールド</mis:input></tr>
</table>
```

BODY 部分は、無視されます。コメント等に使用できます。

HTML ファイルには、コメント部分は出力されません。

37. mis:requestCache

①概要

形式	<mis:requestCache/>
body	なし

②説明

RequestCacheTag は、request オブジェクトを キャッシュすることにより、再検索時や、各画面遷移時の項目の持ち回りを行います。command = "NEW" で、キャッシュし、RENEW で、取り出します。暫定的にこのタグは、共通 JSP ファイルに設定し、HTML そのもののキャッシュ制御も 行うように設定しています。

③属性

④使用例

38. mis:shell

①概要

形式	<mis:shell program="..." />
body	なし

②説明

val1 属性 と val2 属性の文字列表現の比較により BODY を表示/非表示を切り替える簡易タグです。val1.equals(val2) が 成立すれば、BODY を表示します。val1 が null の場合は、無条件で非表示とします。(val1 == val2) はチェックしないというイメージです。

③属性

属性	意味	必須
program	シェルファイルを実行する時に引き渡すコマンド	○
useBatch	EXE ファイルを実行する時に引き渡すコマンド	
wait	プロセスの終了を待つ(true)/待たない(false) (デフォルト true)	
stdout	実行結果の標準出力を出力するかどうか(デフォルト false)	
stderr	実行結果のエラー出力を出力するかどうか(デフォルト false)	
debug	デバッグ情報を 出力する/しないを指定します。	

④使用例

```
<mis:equals val1="ABCD" val2="{@value}" >
  val1.equals( val2 ) 時に表示したい内容
</mis:equals>
```

39. mis:errorMessage

①概要

形式	<mis:errorMessage command = "{@command}" clear = "{@clear}" />;
body	なし

②説明

エラーメッセージを 表形式で表示するタグオブジェクトです。Query 関係の実行時にエラー/ワーニングが発生すると、MISSystem.ERROR_MESSAGE_KEY をキーに ErrorMessage オブジェクト をセッションに登録します。この情報を元に、表題(TITLE)か、内容(BODY)を表示します。基本的には、表題表示時には、リンクを張り、共通エラー表示画面をオープン できる様になっています。各属性は、{@xxxx} 変数が使用できます。これは、ServletRequest から、xxxx をキーに値を取り出し、この変数に 割り当てます。つまり、この xxxx をキーにリクエストすれば、この変数に値をセットすることができます。

③属性

属性	意味	必須
command	処理コマンド。コマンドをセットします。コマンドは、HTML から (get/post)指定されますので、setCommand()メソッドで 登録します。	
clear	メッセージを初期化(クリア)するかどうかを指定します。メッセージは、一般には、エラーメッセージかワーニングです。最終処理でメッセージが無ければ、標準でクリアします。また、command が NEW の場合も、メッセージは自動でクリアされます。デフォルトは、クリアしない (true 以外)です。	
viewType	表示形式『表題(TITLE)か、内容(BODY)』を指定します。一般には、表題(TITLE) を表示しておきます。表題表示時には、リンクを張り、共通エラー表示画面をオープン できる様になっています。	
debug	デバッグ情報を 出力する/しないを指定します。	
displayMsg	plsSqlUpdate の結果を画面上に表示するかどうかを指定します。指定したメッセージをリソースから取得して表示します。表示させたくない場合は、displayMsg = "" をセットしてください。デフォルトは、『登録しました。』を表示します。	

属性	意味	必須
warningMsg	plsSqlUpdate の結果を画面上に表示するかどうかを指定します。指定したメッセージをリソースから取得して表示します。表示させたくない場合は、warningMsg = "" をセットしてください。デフォルトは、『登録しました。』を表示します。	

④使用例

result.jsp 等の SQL 登録実行後の戻り画面に、上記タグを配置すれば、エラーメッセージが存在すればリンクとなって現れ、無ければなにも現れません。リンクのとび先は自動的に設定されます。なお、clear = "true" または、command="NEW" の場合に、エラーメッセージは、クリアされます。

```
[entry.jsp]
<% String forwardPage = "result.jsp"; %>
<jsp:forward page="<%= response.encodeRedirectURL( forwardPage ) %>" >
  <jsp:param name="command" value="REVIEW" />
  <jsp:param name="clear" value="false" />
</jsp:forward>
```

```
[result.jsp]
<mis:errorMessage command = "{@command}" clear = "{@clear}" />
```

40. mis:forward

①概要

形式	<mis:forward page="..." />
body	なし

②説明

ForwardTag は、フォワードを実行するタグです。フォワード時に、指定の引数をセットできます。各属性は、{@xxxx} 変数を使用できます。これは、ServletRequest から、xxxx をキーに値を取り出し、この変数に割り当てます。つまり、この xxxx をキーにリクエストすれば、この変数に値をセットすることができます。

③属性

属性	意味	必須
page	転送先 URL を指定します。	○
dbkeys	dbkeys セット。URI の引数にセットするキーを CSV 形式でセットします。ここの指定は、DBTableModel 上のデータを取り込みます。	

属性	意味	必須
keys	keys セット。URI の引数にセットするキーを CSV 形式でセットします。	
vals	vals セット。URI の引数にセットする値を CSV 形式でセットします。	
scope	DBTableModel を取得する場合のスコープを指定します。 "request", "page", "session", "applicaton" が指定できます。	

④使用例

41. mis:columnMarker

①概要

形式	<mis:columnMarker > … </mis:columnMarker >
body	あり

②説明

検索結果に対して様々な属性を付加するときに使用します。DBTableModel のカラムの属性を利用して、自由に HTML のタグを組み込むことが可能になります。各属性は、{@xxxx} 変数が使用できます。これは、ServletRequest から、xxxx をキーに値を取り出し、この変数に割り当てます。つまり、この xxxx をキーにリクエストすれば、この変数に値をセットすることができます

③属性

属性	意味	必須
column	マーク処理を行うカラム名を指定します。このカラム名の TableModel に対して、マークを処理します。	
columns	マーク処理を行うカラム名をカンマ区切りで複数指定します。この複数のカラム名の TableModel に対して、マークを処理します。カラム名は、カンマ区切りで複数指定することができます。その場合は、指定のカラムに対して、すべて同一の処理を行います。	
onMark	マークを処理する/処理しないの設定を指定します。マークを処理する場合は、"true"(または "1") 処理しない場合は、"true 以外"(または "0")をセットします。デフォルトは、"true"(マークを処理する)です。	

④使用例

商品 CD2(CDSYHN02)は赤字で表示します。
商品 CD3(CDSYHN03)は-----と表示します。

```

<mis:viewMarker command="{@command}">
  <mis:columnMarker column="CDSYHN02" onMark="true" >
    <font color='red'[VCDSYHN02]</font>
  </mis:columnMarker>
  <mis:columnMarker column="CDSYHN03" onMark="true" >
    <CENTER><-----</CENTER>
  </mis:columnMarker>
</mis:viewMarker>

```

42. mis:viewMarker

①概要

形式	<mis:viewMarker > ... </mis:viewMarker >
body	あり

②説明

検索結果に対して様々な属性を付加するときに使用します。DBTableModel オブジェクトの表示に HTML タグをマークするオブジェクトです。子タグとして、ColumnMarkerTag の Body タグをパースして、タグを作成します。

各属性は、{@xxxx} 変数が使用できます。

これは、ServletRequest から、xxxx をキーに値を取り出し、この変数に 割り当てます。つまり、この xxxx をキーにリクエストすれば、この変数に値をセットすることができます。

③属性

属性	意味	必須
command	処理コマンド。コマンドをセットします。コマンドは、HTML から (get/post) 指定されますので、setCommand()メソッドで 登録します。	○

④使用例

商品 CD2(CDSYHN02)は赤字で表示します。

商品 CD3(CDSYHN03)は-----と表示します。

```

<mis:viewMarker command="{@command}">
  <mis:columnMarker column="CDSYHN02" onMark="true" >
    <font color='red'[VCDSYHN02]</font>
  </mis:columnMarker>
  <mis:columnMarker column="CDSYHN03" onMark="true" >
    <CENTER><-----</CENTER>
  </mis:columnMarker>
</mis:viewMarker>

```


43. mis:file

①概要

形式	<mis:file action="..." fileURL="..." />...</mis:file>
body	あり

②説明

ファイル関連の取扱をおこなうタグです。各種アクション に応じた振る舞いを行います。結果については、false の場合は、body タグを表示して、終了します。[各種アクション] canRead 読み込めるかどうかを判定。canWrite 変更できるかどうか判定。createNewFile 空の新しいファイルを不可分(atomic)に生成。(そのファイルがまだ存在しない場合だけ) delete ファイルまたはディレクトリを削除。exists ファイルが存在するかどうか判定。isDirectory ファイルがディレクトリであるかどうか判定。isFile ファイルが普通のファイルかどうか判定。isHidden ファイルが隠しファイルかどうか判定。mkdir ディレクトリを生成。mkdirs ディレクトリを複数生成。

③属性

属性	意味	必須
fileURL	ファイルの URL を指定します。	
file1	ファイルを作成するときのファイル名を指定します。コマンドの左辺のファイル名です。	
file2	ファイルを作成するときのファイル名を指定します。コマンドの右辺のファイル名です。	
columns	カラムを CSV 形式で指定します。カラムを CSV 形式で指定します。	
action	アクションを指定します。 CanRead: 読み込めるかどうかを判定。 CanWrite: 変更できるかどうか判定。 CreateNewFile: 空の新しいファイルを不可分(atomic)に生成。 (そのファイルがまだ存在しない場合だけ) delete: ファイルまたはディレクトリを削除。 Exists: ファイルが存在するかどうか判定。 IsDirectory: ファイルがディレクトリであるかどうか判定。 IsFile: ファイルが普通のファイルかどうか判定。 IsHidden: ファイルが隠しファイルかどうか判定。 Mkdir: ディレクトリを生成。 Mkdirs: ディレクトリを複数生成。	○
tableId	生成した DBTableModel オブジェクトを session に登録する時の ID。デフォルトは、MISSystem.DB_TABLE_MODEL_KEY です。	

④使用例

例 1 ファイルの存在チェック→存在しなければエラーメッセージを表示。

```
<mis:file action="exists" fileURL="N:/CIR/"
  file1="{@USER.LKISB}/{@USER.LDNO1KAI}.cir/001.sht">
  <mis:message msg="RKE_0157" comment="回路図が存在しません。" />
</mis:file>
```

例 2 N:/Filetemp/にユーザーディレクトリが存在しなければ作成。→失敗した場合エラーメッセージを表示。

```
<mis:file action="mkdir" fileURL="N:/Filetemp/{@USER.ID}" >
  <mis:message comment="エラーが発生しました。システム管理者に連絡してください。" />
</mis:file>
```

44. mis:fileQuery

①概要

形式	<pre><mis:fileQuery select="..." from="..." orderBy="..." desc="true/false" multi="true/false" > <mis:fileWhere name="..." value="..." ... /> <mis:fileWhere name="..." value="..." ... /> ... </mis:fileQuery></pre>
body	あり

②説明

ファイル検索をおこなうタグです。指定のSELECT条件の項目をディレクトリより所得します。並び替えについては、このタグで指定しますが、ファイルの選別は、BODY 部に記述する fileWhere タグで指定します。(複数指定可能)

検索結果については、DBTableModel に登録します。

[select 属性]	検索するファイルの属性を以下のカラム名で指定します。
file_level	ディレクトリを展開する場合のレベル。
file_canRead	読み込めるかどうかを判定。
file_canWrite	変更できるかどうか判定。
file_canonicalPath	この抽象パス名の正規のパス名文字列を返します。
file_path	この抽象パス名をパス名文字列に変換します。
file_name	この抽象パス名が示すファイルまたはディレクトリの名前を返します。
file_parent	この抽象パス名の親のパス名文字列を返します。
file_isDirectory	ディレクトリであるかどうか判定。
file_isFile	普通のファイルかどうか判定。
file_isHidden	隠しファイルかどうか判定。

file_lastModified 最後に変更された時刻を返します。
 file_length ファイルの長さを返します。
 注意:select 属性に指定できる項目は、別名定義できません。

[from 属性] 検索を開始するファイルまたはディレクトリの名前
 [orderBy 属性] ソートを行うキーとなるカラム名称をカンマ区切りで並べます。
 [desc 属性] ソート時のソート順(昇順、降順)を指定します。
 true 降順ソート
 false 昇順ソート(デフォルト)
 [multi 属性] ディレクトリを下位展開するかどうか？
 true 下位展開する
 false 下位展開しない(デフォルト)
 [tableId 属性] session に登録する時の ID
 [scope 属性] session に登録する時のスコープ "request","page","session","applicaton"
 [maxRowCount 属性] 検索時の最大検索件数
 [displayMsg 属性] 検索終了時に表示する メッセージリソースの ID
 [overflowMsg 属性] オーバーフロー時に表示する メッセージリソースの ID
 [command 属性] 実行を制御するコマンド "NEW" と "RENEW" 時のみ実行します。(デフォルト NEW)

③属性

属性	意味	必須
select	検索するファイル属性をカンマ指定します。level ファイル階層のレベル canRead アプリケーションが読み込めるかどうかを判定します。 canWrite アプリケーションが変更できるかどうかを判定します。 canonicalPath 正規のパス名文字列を返します。 name ファイルまたはディレクトリの名前を返します。 parent 親のパス名文字列を返します。 path パス名文字列に変換します。 directory ファイルがディレクトリであるかどうかを判定します。 isFile ファイルが普通のファイルかどうかを判定します。 isHidden ファイルが隠しファイルかどうかを判定します。 lastModified ファイルが最後に変更された時刻を返します。 length ファイルの長さを返します。	
form	ファイルの検索元となるディレクトリを指定します。	
orderBy	検索した結果を表示する表示順をファイル属性名で指定します。 複数指定する場合は、カンマ区切りで並べてください。	
desc	表示順を逆転するかどうかを指定します。 デフォルトは、false (昇順) です。	
multi	多段階展開するか、1レベル展開するか指定します。デフォルトは、false (1レベル) です。	
tableId	(通常は使いません)結果を DBTableModel に書き込んで、session に登録するときのキーを指定します。 デフォルトは、MISSystem.DB_TABLE_MODEL_KEY です。	
level	多段階展開するレベルを指定します。	

属性	意味	必須
command	処理コマンド。コマンドをセットします。コマンドは、HTML から (get/post)指定されますので、setCommand()メソッドで 登録します。	
maxRowCount	データベースの最大検索件数をセットします。 データベース自体の検索は、指定された SQL の全件を検索しますが、DBTableModel のデータとして登録する最大件数をこの値に設定します。 サーバーのメモリ資源と応答時間の確保のためです。	
displayMsg	検索結果を画面上に表示するかどうかを指定します。 ここでは、検索結果の件数や登録された件数をまず出力し、その次に、ここで指定したメッセージをリソースから取得して 表示します。 表示させたくない場合は、displayMsg = "" をセットしてください。 デフォルトは、検索件数を表示します。	
overflowMsg	オーバーフロー時のメッセージを画面上に表示するかどうかを指定します。 表示させたくない場合は、overflowMsg = "" をセットしてください。	

④使用例

例 1 一般的な属性でファイルの検索を行います。

```
<mis:fileQuery
    select = "file_level,file_isDirectory,file_name,file_lastModified,file_length"
    from   = "d:/webapps/dbdef/jsp/"
    orderBy = "file_name"
    multi  = "true"
    command = "{@command}" >
    <mis:fileWhere name="file_name" endWith=".jsp" />
</mis:fileQuery>
```

例 2 最終変更日で逆順ソートします。対象は、2002/10/01 以降に変更されたファイル。

```
<mis:fileQuery
    select = "file_level,file_isDirectory,file_canonicalPath,file_name,file_lastModified,file_length"
    from   = "d:/webapps/dbdef/jsp/"
    orderBy = "file_lastModified"
    desc   = "true"
    multi  = "true"
    command = "{@command}" >
    <mis:fileWhere name="file_lastModified" gt="20021001000000" />
</mis:fileQuery>
```

45. mis:fileOption

①概要

形式	<mis:select name="..."> <mis:fileOption from="..."/> </mis:select>
body	あり

②説明

ファイルリストの作成をおこなうタグです。本タグは Select タグの BODY 部に指定します。並び替えについては、このタグで指定しますが、ファイルの選別は、BODY 部に記述する fileWhere タグで指定します。(複数指定可能)

③属性

属性	意味	必須
select	Option に表示する項目を指定します。	
from	ファイルの検索元となるディレクトリを指定します。	
value	Option の初期値で選ばれる値を指定します。	
orderBy	検索した結果を表示する表示順をファイル属性名で指定します。複数指定する場合は、カンマ区切りで並べてください。	
desc	表示順を逆転するかどうかを指定します。デフォルトは、false (昇順) です。	
fileType	表示するファイルの種別を指定します。複数指定する場合は、カンマ区切りで並べてください。	

④使用例

46. mis:fileWhere

①概要

形式	<mis:fileWhere name="..." action="..." value="..."/>
body	あり

②説明

fileQuery で検索するとき使用する検索条件を指定します。

③属性

属性	意味	必須
name	FileWhereTag のファイルフィルタの name 属性を設定します。	
value	FileWhereTag のファイルフィルタの value 属性を設定します。	
action	FileWhereTag のファイルフィルタの action 属性を設定します。	
ignoreCase	大文字、小文字を区別するかを指定します。	

④使用例

```
<mis:fileOptionselect="NAME"from="./jsp">
  <mis:fileWherename="file_name"action="endWith"value=".txt"/>
</mis:fileOption>
```

例 1 検索条件が入力された時

条件を満足して全てのファイルを列挙します。

例 2 検索条件が入力されなかった時

該当ディレクトリ内の全てのファイルを列挙します。

47. mis:img

①概要

形式	<mis:img src="..." alt="..."/>
body	なし

②説明

GIF等のイメージファイルを表示するタグです。

③属性

属性	意味	必須
name	名前を指定します。	
src	表示させる画像の URL を指定します。	○
alt	像が表示されない場合、代わりに表示させるテキストを指定します。	○
longdesc	longdesc を指定します。	
width	横幅を指定します。	
height	縦幅を指定します。	
usemap	クライアントサイドクlickカブルを指定します。	
ismap	サーバーサイドクlickカブルを指定します。	

属性	意味	必須
clazz	class セット。Java の言語使用上、class で作成できないため、代用として clazz を使用している。html で作成される属性は、class で作成される。	
id	id セット	
title	title セット	
style	style セット	
lang	lang セット	
dir	dir セット	

④使用例

```
<mis:imgsrc="{@URL}" alt="sample"/>
```

src:表示させる画像の URL (必須)

alt:画像が表示されない場合、代わりに表示させるテキスト (必須)

48. mis:sql

①概要

形式	<mis:sql command="..."> ... </mis:sql>
body	あり

②説明

SQLTag は、画面上にデータベースの検索結果を表示するタグです。例えば、社員番号のキーに対して、社員名称を表示させたい場合に使用します。

command="NEW"の場合と、SQL 文が完全に成立した場合に、SQL 文が実行されます。SQL 文の {@xxxx}に、値が設定されていない場合は、SQL 文が完全に成立したとはみなされませんので、データベースへのアクセスは実行されません。SQL 文が実行されなかった場合は、defaultVal にセットされた値を代わりに表示します。初期値は、スペースです。検索時のカラムが複数ある場合は、スペースで連結して返します。行が複数ある場合は、各行(複数カラム)を連結した後に、全行を連結します。各属性は、{@xxxx}変数が使用できます。これは、ServletRequest から、xxxx をキーに値を取り出し、この変数に割り当てます。つまり、この xxxx をキーにリクエストすれば、この変数に値をセットすることができます。

③属性

属性	意味	必須
command	処理コマンド。コマンドをセットします。コマンドは、HTML から (get/post)指定されますので、setCommand()メソッドで登録します。	
defaultVal	検索結果の値が null の場合に初期値をセットします。	

④使用例

49. mis:tbody

①概要

形式	<mis:tbody> … </mis:tbody>
body	あり

②説明

検索結果に対して表示フォーマットを変更するときに使用します。theadTag は、ボディー部のフォーマットを定義するタグです。

③属性

属性	意味	必須
rowspan	表示データを作成する場合のフォーマットの行数をセットします。	

④使用例

50. mis:tfoot

①概要

形式	<mis:tfoot>…</mis:tfoot>
body	あり

②説明

検索結果に対して表示フォーマットを変更するときに使用します。theadTag は、フッター部のフォーマットを定義するタグです。

③属性

属性	意味	必須
rowspan	表示データを作成する場合のフォーマットの行数をセットします。	

④使用例

51. mis:thead

①概要

形式	<mis:thead>...</mis:thead>
body	あり

②説明

検索結果に対して表示フォーマットを変更するときに使用します。theadTag は、ヘッダー部のフォーマットを定義するタグです。

③属性

属性	意味	必須
rowspan	表示データを作成する場合のフォーマットの行数をセットします。	

④使用例

52. mis:columnEditor

①概要

形式	<mis:ColumnEditor>...</mis:ColumnEditor>
body	あり

②説明

検索結果に対して様々な属性を付加するときに使用します。DBTableModel オブジェクトの表示に HTML タグをマークするオブジェクトです。子タグとして、ColumnMarkerTag の Body タグをパースして、タグを作成します。

各属性は、{@xxxx}変数が使用できます。これは、ServletRequest から、xxxx をキーに値を取り出し、この変数に割り当てます。つまり、この xxxx をキーにリクエストすれば、この変数に値をセットすることができます。

③属性

属性	意味	必須
lang	処理コマンド。言語を設定します。	
column	処理コマンド。ネームをセットします。	
label	処理コマンド。ラベルをセットします。	
className	処理コマンド。カラムのクラスをセットします。	

属性	意味	必須
size	処理コマンド。カラムのサイズをセットします。	
writable	処理コマンド。カラムの書き込み可能をセットします。	
renderer ※1	処理コマンド。カラムの表示用レンダラーをセットします。	
editor ※2	処理コマンド。カラムの編集用エディターをセットします。	
codeSel	処理コマンド。パラメータをセットします。	
dbType ※3	処理コマンド。データベースタイプをセットします。	
defaultVal	処理コマンド。デフォルト値をセットします。	
parameter	処理コマンド。パラメータをセットします。	
must	必須入力を表す色に変えるかどうか指定します。デフォルトは、必須でない("false")です。(注意)必須チェックとは連動していません。	
optionAttributes	optionAttributes をセットします。JavaScript などの HTML 基本タグ以外の属性を、そのままタグとして使用します。そのときに、(')は、(")に置き換えます。	
addNoValue	コンボボックスの値なしの Option の有無を指定します。コンボボックスの場合、値なしの Option を Select に含めるかどうかを指定します。デフォルトは、含める("true")です。 <mis:columnEditorcolumn="CDBK"addNoValue="false"/>	

※1 renderer 属性に指定できる値は、第III部-『第4章 カラム属性』-『1. 表示種別』の中から選択できます。

※2 editor 属性に指定できる値は、第III部-『第4章 カラム属性』-『2. 入力種別』の中から選択できます。

※3 dbType 属性に指定できる値は、第III部-『第4章 カラム属性』-『3. データベースタイプ』の中から選択できます。

④使用例

例1 必須表示の場合

```
<mis:columnEditor column="NOSYN" must="true"/>
```

例2 DB からのコンボボックスの場合

```
<mis:columnEditor column="CDBK" editor="DBMENU">
  <jsp:text>
    select CDBK,CDBK || ':' || NMBK from DB02 order by 1
  </jsp:text>
</mis:columnEditor>*
```

53. mis:favoriteLink

①概要

形式	<mis:favoriteLink direct="true" target="_new" method="GET"/>
body	あり

②説明

お気に入りリンクを作成する簡易タグです。画面検索時の引数やユーザー情報を元にして、ダイレクトに再検索できるリンクを作成します。このリンクをお気に入り等にセーブしておき、次回検索時にも使用できるようにします。

③属性

属性	意味	必須
target	リンク先の文書を表示させるフレーム、またはウィンドウの名前を指定します。リンク先のフレーム名(ターゲット属性)を設定します。	
direct	直接アクセスできる形式のリンクを作成するかどうかを指定します。true(デフォルト)では、ユーザー:パスワードの入力なしで直接アクセスできる形式のリンクが作成されます。false の場合は、通常のリンクが作成されます。	
method	リンクの作成元となるメソッド(POST/GET/ALL)を指定します。ここで指定したメソッドでリクエストされた場合のみ、リンクを作成します。デフォルトは、GET です。(GET のみリンクを作成)これは、POST では、引数が付かないためです。(実際は付けることができますが・・・)ALL は、どちらの場合でもリンクを作成しますが、先に述べたように POST では引数がつきません。	
title		
msg	メッセージを指定します。メッセージ ID は、所定の language に基づく ResourceManager の getMessage(id)を呼び出し、その結果を msglbl に登録します。getMsglbl()で取り出せます。ラベル ID とメッセージ ID は同時には登録できません。	
lbl	ラベルを指定します。ラベルを変更するときに、lbl 属性を使います。ラベル ID は、所定の language に基づく ResourceManager の getLabel(id)を呼び出し、その結果を msglbl に登録します。getMsglbl()で取り出せます。ラベル ID とメッセージ ID は同時には登録できません。	

④使用例

直接お気に入りのリンクを作成します。

```
デフォルト属性を使用 (direct="true" target="_new" method="GET")
<mis:favoriteLink>FavoriteLink</mis:favoriteLink>
```

54. mis:writeChart

①概要

形式	<mis:writeChart writerClass="SVGBar" ... />
body	なし

②説明

DBTableModel オブジェクトをチャートで表示する、カスタムタグです。このオブジェクトに、データ (DBTableModel) と、コントローラ (DBTableWriter) を与えて、外部からコントロールすることで、各種形式でデータ (DBTableModel) を表示させることが可能です。WriteChartTag に対して、コマンドを与えることにより、内部のコントローラの実装に対応した、形式でデータを作成します。

各属性は、{@xxxx}変数が使用できます。これは、ServletRequest から、xxxx をキーに値を取り出し、この変数に割り当てます。つまり、この xxxx をキーにリクエストすれば、この変数に値をセットすることができます。

③属性

属性	意味	必須
writerClass ※1	実際に書き出すクラス名 (の略称) をセットします。	○
title	チャートのタイトルをセットします。	
width	チャートの横幅をセットします。タイトルや凡例も含んだ大きさです。データ領域は自動計算されます。	
height	チャートの縦幅をセットします。タイトルや凡例も含んだ大きさです。データ領域は自動計算されます。	
maxRange	チャートの縦軸の最大値をセットします。0 (デフォルト) 時には、データの最大値より自動的に最適な値を設定します。	
minRange	チャートの縦軸の最小値をセットします。チャートの縦軸の最小値をセットします。	
noteDisplay	チャートの凡例の表示可否をセットします。初期値は、表示する (true) です	
language	言語コードを指定します。この言語コードに基づいて、表示のラベルをリソースから作成します。	
tableId	session から所得する DBTableModel オブジェクトの ID。表示処理後に、(内部ポインタを書き換えた) DBTableModel オブジェクトを同じキーで、session に登録します。デフォルトは、MISSystem.DB_TABLE_MODEL_KEY です。	
scope	キャッシュする場合の範囲を指定します。 "request", "page", "session", "applicaton" が指定できます。	

※1 writerClass 属性に指定できる値は、第III部『第7章 Writer/Reader クラス』—『3. ChartDBTableWriter クラス』の中から選択できます。

④使用例

55. mis:comment

①概要

形式	<pre><mis:comment title="タイトル" version="バージョン" build="ビルド番号" date="変更日付"[必須] author="変更者"[必須] > 変更内容 </mis:comment></pre>
body	あり

②説明

JSP アプリケーションに対してバージョン管理できるようにコメントを記述するためのタグです。属性には、バージョン、ビルド番号、日付、変更者、タイトルです。変更内容は、BODY 部に記述します。

③属性

属性	意味	必須
title	コメントのタイトルを設定します。	
version	コメントのバージョンを設定します。	
build	コメントのビルド番号を設定します。	
Date	コメントの変更日付を設定します。	
author	コメントの変更者を設定します。	

④使用例

56. mis:statusBar

①概要

形式	<p>① メッセージの書き込み処理 (Admin ツールより登録する)</p> <pre><mis:statusBar action="SET" timer="500" brink="true"> <jsp:text>{@MESSAGE}</jsp:text> </mis:statusBar></pre> <p>① メッセージの表示処理 (custom/result_info.jsp に、設定しておきます。)</p> <pre><mis:statusBar action="GET"/></pre>
body	あり

②説明

管理者が全てのユーザーにメッセージを同報できるタグオブジェクトです。STATUS_BAR_OBJECT_KEY に、保存されるメッセージオブジェクトから表示属性を設定/取得します。全てのユーザーに対して、同時にステータスバーに、必要なメッセージを書き出します。メッセージ、タイマー、ブリンクか繰り返しか、指定できます。各属性は、{@xxxx}変数が使用できます。これは、ServletRequest から、xxxx をキーに値を取り出し、この変数に割り当てます。つまり、この xxxx をキーにリクエストすれば、この変数に値をセットすることができます。

③属性

属性	意味	必須
action	アクションを指定します。アクションは、メッセージを書き込み (SET)読み出し (GET)消去 (REMOVE)の中から指定します。これらは、大文字で、かつリクエスト変数は使えません。	
timer	メッセージを表示する場合の、タイマーを[ms]で指定します。通常は、デフォルトの500ms で良いでしょう。デフォルトは、500 ms です	
brink	表示形式『点滅(true)か、繰り返し(false)』を指定します。ステータスバーに出す場合に、メッセージが点滅するか、繰り返し流れにするかを指定します。デフォルトは、繰り返し(false)になっています。	
image	イメージファイルの表示可否を指定します。このタグでイメージファイルを出力したい場合に"true"にセッティングし、イメージファイルを指定すれば、そのファイルを表示します。デフォルトは、"false"です。	

属性	意味	必須
src	イメージファイルのソース URI を指定します。このタグをイメージファイルを出力したい箇所にセッティングし、イメージファイルを指定すれば、そのファイルを表示します。なお、イメージファイルを使用する場合は、image 属性に"true"をセットしてください。デフォルトは、"./image/hint.gif"です。	

④使用例

書き込み(action="SET")と、読み出し(action="GET")でタグを配置する場所が異なります。actionとして、SET(書き込み)、GET(読み出し)、REMOVE(消去)を指定できます。

57. mis:report

①概要

形式	<mis:report ... />
body	なし

②説明

DBTableModel オブジェクトをレポート形式に返還するタグリブクラスです。このオブジェクトに、データ(DBTableModel)と、コントローラ(DBTableReport クラス)を与えて、外部からコントロールすることで、各種形式でデータ(DBTableModel)を表示させることが可能です。

各属性は、{@xxxx}変数が使用できます。これは、ServletRequest から、xxxx をキーに値を取り出し、この変数に割り当てます。つまり、この xxxx をキーにリクエストすれば、この変数に値をセットすることができます。

http://localhost/query.jsp?KEY1=VLA1&KEY2=VAL2
 のようなリクエストで、{@KEY}1とすれば、VAL1 がセットされます。

③属性

属性	意味	必須
headerKeys	固定部の{@KEY}の KEY 部分を指定します。カンマで複数指定できます。	
headerVals	固定部の KEY に対応する値を指定します。{@KEY}に置き換わります。カンマで複数指定で、リクエスト情報でも設定できます。	
footerKeys	繰り返し部の終了後に表示する key 部分を指定します。カンマで複数指定できます。	
footerVals	固定部の KEY に対応する値を指定します。{@KEY}に置き換わります。カンマで複数指定で、リクエスト情報でも設定できます。footer 関連の値をセットした場合は、pageEndCut は、強制的に"false"にされます。	

属性	意味	必須
pageEndCut	ボディー部(繰り返し部)がなくなったときに、それ以降を表示するかどうかを指定します。true では、それ以降を出力しません。footer 関連の値をセットした場合は、pageEndCut は、強制的に"false"にされます。デフォルト"true"(なくなった時点で、出力しない。)です。	
fileURL	fileURL をセットします。	
filename	ファイルを作成するときの雛型ファイル名をセットします。	○
outFileURL	outFileURL をセットします。	
outFilename	ファイルを作成するときの出力ファイル名をセットします。	○
encode	ファイルを作成するときのファイルエンコーディング名をセットします。 "DEFAULT","JISAutoDetect","JIS","EUC_JP","MS932","SJIS","Windows-31J","Shift_JIS"	
reportClass	実際に書き出すクラス名(の略称)をセットします。デフォルトは、"HTMLPrint"です。	
language	言語コードを指定します。この言語コードに基づいて、表示のラベルをリソースから作成します。	
tableId	session から所得する DBTableModel オブジェクトの ID。表示処理後に、(内部ポインタを書き換えた)DBTableModel オブジェクトを同じキーで、session に登録します。デフォルトは、MISSystem.DB_TABLE_MODEL_KEY です。	
scope	キャッシュする場合の範囲を指定します。 "request","page","session","applicaton"が指定できます。	
debug	デバッグ情報を出力する/しないを指定します。	

④使用例

58. mis:value

①概要

形式	<misvalue command="SET" key="..." value="..." />
body	あり

②説明

ValueTag は JSP 上からキー、値を設定することにより、リクエスト情報にキー、値を記録させ、キーにより取り出し可能なカスタムタグです。通常のリクエスト情報と同じ扱いができます。(優先順位は、Value タグが上位) 設定した値は、{@xxxx}形式で取り出すことができます。また、command="GET"で直接画面に値を書き出すことも可能です。

③属性

属性	意味	必須
command	処理コマンド。コマンドをセットします。コマンドは、HTML から (GET/SET/REMOVE/SQL/SETTBL/GETTBL/KEYTBL/CLEAR)指定されますので、setCommand()メソッドで登録します。	
action	処理アクション。アクションをセットします。アクションは、HTML から (UPPER/LOWER/MESSAGE/APPEND)指定されますので、setAction()メソッドで登録します。	
key	リクエスト情報に登録するキーをセットします。	
value	リクエスト情報に登録する値をセットします。	
defaultVal	初期値を設定します。value 値が NULL の場合に、この初期値をセットします。	
nullSet	value が NULL の時に、設定するかどうかを指定します。true の場合は、null のときでもセットします。false の場合は、null のときは、既存の値を置き換えません。デフォルトは、null のときでもセットするです。("true")	
dbid	(通常は使いません)Query オブジェクトを作成する時の DB 接続 ID を指定します。	
parameter	アクションの処理パラメータを設定します。	
scope	キャッシュする場合の範囲を指定します。"request","session"が指定できます。デフォルトは、"request"です。	
debug	デバッグ情報を出力する/しないを指定します。	

④使用例

例 1 <mis:value command="SET" key="ABC" value="123" />

例 2 <mis:value command="SQL"><jsp:text>SELECT1TESTFROMDUAL</jsp:text></mis:value>

例 3 <mis:value command="SET" key="DEF" value={@NOCOMMAND} defaultValue="0000" />

例 4 <mis:value command="GET" key="NOKEY" default value="NODATA"/>
 例 5 <mis:value command="SET" key="{@ABC}4" value="TEST1234"/>
 例 6 <mis:value command="GET" key="ABC"/>
 例 7 <mis:value command="SET" action="LOWER" key="LOWERTEST" value="ABCDEF"/>
 例 8 <mis:value command="GET" key="LOWERTEST"/>
 例 9 <mis:value command="GET" action="UPPER" key="LOWERTEST"/>
 例 10 <mis:value command="REMOVE" key="ABC"/>

59. mis:schedule

①概要

形式	<mis:schedule command="..." timerTask="..." />
body	なし

②説明

Context (アプリケーション) 共有のタイマー機能を用いて、スケジュールを、設定します。スケジュールすべきクラスは、MISTimerTask を拡張 (extends) しておく必要があります。

③属性

属性	意味	必須
command	処理コマンド。コマンドをセットします。コマンドは、HTML から (get/post)指定されますので、setCommand()メソッドで登録します。	
timerTask	処理を実行する MISTimerTask のクラス名を指定します。	○
name	タイマータスクの名称 (ユニークキー) を設定します。	○
comment	タイマータスクの説明を設定します。	
singleton	Name 属性が同一のタスクを2重登録できないようにします。同一種類のタスクを複数同時立上げたくない場合に、利用します。一般に、タイマータスクそのものが、繰り返し実行されますので、初期値は、2重登録禁止にしています。	
delay	タスクが実行される前のミリ秒単位の遅延を設定します。(デフォルト 0)	
period	連続するタスク実行のミリ秒単位の時間を設定します。(デフォルト 60000ms)	

属性	意味	必須
fixedDelay	固定遅延実行(true)/固定頻度実行(false)(デフォルト true)を設定します。固定遅延実行では、前の実行の実際の実行時間を基準にしてそれぞれの実行がスケジュールされます。何らかの理由で実行が遅延した場合そのあとの実行も遅延されます。最終的に、実行の頻度は通常、指定した期間の対応する頻度よりも若干遅くなります固定頻度実行では、最初の実行のスケジュールされた実行時間を基準にしてそれぞれの実行がスケジュールされます。何らかの理由で実行が遅延した場合、「遅れを取り戻す」ために2つ以上の実行が連続して行われます。最終的に実行の頻度は、指定した期間の対応する頻度と同じになります	
commonforward		

④使用例

60. mis:ldapQuery

①概要

形式	<mis:ldapQuery ... />
body	なし

②説明

LDAP にアクセスするための、ldapQuery タグです。このタグで検索した結果は、DBTableModel にセットされるため、JDBCQuery と同様に、view タグで表示させることが可能です。

下記の項目については、src/resource/SystemResource.properties に、あらかじめ設定しておきます。ことで、タグごとに指定する必要がなくなります。

LDAP_INITIAL_CONTEXT_FACTORY
 LDAP_PROVIDER_URL
 LDAP_ENTRYDN
 LDAP_SEARCH_BASE
 LDAP_SEARCH_SCOPE このタグリブでは、検索のみサポートしています。

③属性

属性	意味	必須
initctx	使用する初期コンテキストファクトリを指定するための、環境プロパティの名前を保持する定数です。初期値は、SystemResource.properties の INITIAL_CONTEXT_FACTORY 属性です。	

属性	意味	必須
providerURL	使用するサービスプロバイダの構成情報を指定するための、環境プロパティの名前を保持する定数です。プロトコルとサーバーとポートを指定します。例) 『ldap://ldap.muratec.co.jp:389』	
entrydn	属性の取得元のオブジェクトの名前 例)『cn=ldapAdministrator,o=muratec,c=JP』	
searchbase	検索するコンテキストまたはオブジェクトの名前。例) 『ou=employeeuser,o=muratec,c=JP』	
filter	検索に使用するフィルタ式。null は不可。例) 『employeeNumber=87019』	
searchScope	検索範囲を OBJECT_SCOPE、ONELEVEL_SCOPE、SUBTREE_SCOPE のどれか 1 つに設定します。指定文字列は、『OBJECT』『ONELEVEL』『SUBTREE』のどれかです。	
timeLimit	これらの SearchControls の時間制限をミリ秒単位で設定します。値が 0 の場合、無制限に待つことを意味します。	
derefLinkFlag	検索中のリンクへの間接参照を有効または無効にします。	
maxRowCount	レジストリの最大検索件数をセットします。DBTableModel のデータとして登録する最大件数をこの値に設定します。サーバーのメモリ資源と応答時間の確保のためです。0 は、無制限です。(デフォルトは、無制限です。)	
returningObjFlag	結果の一部としてオブジェクトを返すことを有効または無効にします。無効にした場合、オブジェクトの名前およびクラスだけが返されます。有効にした場合、オブジェクトが返されます。	
attributes	検索の一部として返される属性を指定します。null はすべての属性が返されることを示します。は属性が何も返されないことを示します。このメソッドからは、空の配列をセットすることはできません。	
orderBy	検索した結果を表示する表示順をファイル属性名で指定します。 現仕様では、複数のキーを指定することはできません。	
desc	表示順を逆転するかどうかを指定します。デフォルトは、false(昇順)です。	
tableId	(通常は使いません)結果を DBTableModel に書き込んで、session に登録するときのキーを指定します。デフォルトは、MISSystem.DB_TABLE_MODEL_KEY です。	
command	処理コマンド。コマンドをセットします。コマンドは、HTML から (get/post)指定されますので、setCommand()メソッドで登録します。	
displayMsg	検索結果を画面上に表示するかどうかを指定します。ここでは、検索結果の件数や登録された件数をまず出力し、その次に、ここで指定したメッセージをリソースから取得して表示します。表示させたくない場合は、displayMsg=""をセットしてください。デフォルトは、検索件数を表示します。	

属性	意味	必須
overflowMsg	オーバーフロー時のメッセージを画面上に表示するかどうかを指定します。表示させたくない場合は、overflowMsg=""をセットしてください。	
debug	デバッグ情報を出力する/しないを指定します。	
scope	キャッシュする場合の範囲を指定します。 "request","page","session","applicaton"が指定できます。	

④使用例

```
<mis:ldapQuery debug="false"
  command      = "NEW"           NEW,RENEW が使用可能です。
  initctx      = "com.sun.jndi.ldap.LdapCtxFactory"  初期コンテキストファクトリ
  providerURL  = "ldap://ldap.muratec.co.jp:389"    サービスプロバイダの構成情報
  entrydn      = "cn=ldapAdministrator,o=muratec,c=JP"  属性の取得元のオブジェクトの名前
  searchbase   = "ou=employeeuser,o=muratec,c=JP"
                                     検索するコンテキストまたはオブジェクトのベース名
  filter       = "ou=22"          検索に使用するフィルタ式。null は不可
  attributes   = "officeName,businessName,cn,employeeNumber"
                                     属性を識別する属性 ID をカンマ区切りで指定
  searchScope  = "SUBTREE"
                                     検索範囲を設定(『OBJECT』『ONELEVEL』『SUBTREE』のどれか)
  maxRowCount = "0"              検索時の最大検索数(0 は、無制限:デフォルト)
/>
```

SystemResource.properties に、必要なLDAP 定数を定義しておいた場合は、下記のタグで検索できます。

例では、filter にあいまい検索(*)と AND 条件(&(条件 A)(条件 B))を指定した例を挙げておきます。

filter の形式と解釈は、RFC2254 に準拠しています。

(RFC2254 で定義されている演算子のすべてをすべての属性に適用できるわけではありません。)

```
<mis:ldapQuery debug="false"
  command      = "NEW"
  filter       = "&(cnphoneticlangjp=ハセガワ*)(businessName=本社)"
  attributes   = "officeName,businessName,cn,employeeNumber"
/>
```

61. mis:regQuery

①概要

形式	<mis:regQuery basekey="..." />
body	なし

②説明

Windows レジストリにアクセスするための、regQuery タグです。このタグで検索した結果は、DBTableModel にセットされるため、JDBCQuery と同様に、view タグで表示させることが可能です。

③属性

属性	意味	必須
baseKey	検索ベースキーを設定します。キーは、HKEY_CURRENT_USER¥Software 以下をベースとします。	○
key	検索キーを設定します。	
value	検索バリューを設定します。	
maxRowCount	レジストリの最大検索件数をセットします。DBTableModel のデータとして登録する最大件数をこの値に設定します。サーバーのメモリ資源と応答時間の確保のためです。0 は、無制限です。(デフォルトは、無制限です。)	
orderBy	検索した結果を表示する表示順をファイル属性名で指定します。 現仕様では、複数のキーを指定することはできません。	
maxLevel	検索時の最大展開レベルを指定します。0を指定すると、無制限に階層を展開します。デフォルトは、1レベルです。	
like	キーおよびバリューについて、like 検索を行うかどうかを指定します。like 検索とは、キーの一部の文字を含む場合にマッチしたとして、値を取り出します。ここでの設定は、キーもバリューも同時に適用されます。また、大文字小文字の区別も行いません。	
tableId	(通常は使いません)結果を DBTableModel に書き込んで、session に登録するときのキーを指定します。デフォルトは、MISSystem.DB_TABLE_MODEL_KEY です。	
command	処理コマンド。コマンドをセットします。コマンドは、HTML から (get/post) 指定されますので、setCommand()メソッドで登録します。	

属性	意味	必須
displayMsg	検索結果を画面上に表示するかどうかを指定します。ここでは、検索結果の件数や登録された件数をまず出力し、その次に、ここで指定したメッセージをリソースから取得して表示します。表示させたくない場合は、displayMsg=""をセットしてください。 デフォルトは、検索件数を表示します。	
overflowMsg	オーバーフロー時のメッセージを画面上に表示するかどうかを指定します。表示させたくない場合は、overflowMsg=""をセットしてください。	
debug	デバッグ情報を出力する/しないを指定します。	
scope	キャッシュする場合の範囲を指定します "request","page","session","applicaton"が指定できます。	

④使用例

```

<mis:regQuery      debug="false"
  command          = "NEW"                NEW,RENEW が使用できます。
  baseKey          = "/microsoft/windowsnt" 検索時のベースとなるレジストリキー名
  key              = "driver"             検索したいレジストリキー名(デフォルトで全件)
  value           = "Ne"                 検索したいレジストリ値(デフォルトで全件)
  maxRowCount     = "0"                 最大検索件数(0で無制限)
  maxLevel        = "0"                 最大検索階層レベル(0で無制限)
  like            = "true"              trueで曖昧検索/falseは一致検索
/>

```

like は、key/value を設定したときのみ有効です。また、key/value を両方同時に設定した場合は、like 属性は両方に同時に適用されます。



警告！

このタグを使用するには、java.util.prefs.WindowsPreference クラスを変更し、正規のクラスローダーより先に読み込ませる必要があります。それは、endorsed を利用します。但し、現時点では、全クラスが対象になってしまうため、標準的に利用することはできません。(セキュリティ及び、不正な書込みを防ぐため)

62. mis:submit

①概要

形式	<mis:submit value="..." msg="..." />
body	なし

②説明

サブミットボタンを表示するタグです。value 属性 value 属性に、forward したい JSP ファイル名を記述します。JSP ファイル名は、標準で、insert,copy,modify,delete などと指定します。実際には、各 JSP 画面 (insert.jsp,copy.jsp,modify.jsp,delete.jsp) にリクエストが転送されます。

この JSP ファイル名は、同一画面 ID 内のフォルダに属している必要があります。

この ID は、JSP 画面そのものですので、大文字小文字は区別されます。

command 属性 command 属性を指定しない場合は、この value 属性値が、コマンドになります。先の例では、それぞれ、INSERT,COPY,MODIFY,DELETE というコマンドになります。コマンドは、大文字です。

msg 属性/lbl 属性 msg 属性/lbl 属性は、ボタンのラベルを表示します。過去のボタンには、ショートカット文字列もラベルとして登録する必要がありましたが、ここでは、自動付与されますので、純粋なラベルを指定してください。

accesskey 属性 accesskey 属性には、ショートカット記号を指定できます。これは、先のラベルに自動で追加されます。

gameId 属性 gameId 属性は、別の画面に forward する場合に使用します。実際は、forward ではなく、sendRedirect されます。

このタグは、必ず、WriteCheckTag で囲まれている必要があります。

③属性

属性	意味	必須
value	forward したい JSP ファイル名を記述します。JSP ファイル名は、標準で、insert,copy,modify,delete などと指定します。実際には、各 JSP 画面 (insert.jsp,copy.jsp,modify.jsp,delete.jsp) にリクエストが転送されます。この JSP ファイル名は、同一画面 ID 内のフォルダに属している必要があります。この ID は、JSP 画面そのものですので、大文字小文字は区別されます。	○
command	処理コマンド。command 属性を指定しない場合は、この value 属性値が、コマンドになります。value 属性に、insert,copy,modify,delete などと指定されていた場合は、それぞれ、INSERT,COPY,MODIFY,DELETE というコマンドになります。コマンドは、大文字です。	
gameId	gameId 属性を登録します。gameId 属性は、別の画面に forward する場合に使用します。実際は、forward ではなく、sendRedirect されます。	
id	id セット	

属性	意味	必須
lang	lang セット	
dir	dir セット	
title	title セット	
style	style セット	
readonly	readonly セット。readonly="readonly"以外の値はセットできないように制限をかけます。(大文字/小文字も判別します)	
disabled	disabled セット。disabled="disabled"以外の値はセットできないように制限をかけます。(大文字/小文字も判別します)	
tabindex	tabindex セット	
accesskey	accesskey セット	
clazz	class セット。Java の言語使用上、class で作成できないため、代用として clazz を使用している。html で作成される属性は、class で作成される。	
msg	メッセージを指定します。メッセージ ID は、所定の language に基づく ResourceManager の getMessage(id)を呼び出し、その結果を msglbl に登録します。getMsglbl()で取り出せます。ラベル ID とメッセージ ID は同時には登録できません。	
lbl	ラベルを指定します。ラベルを変更するときに、lbl 属性を使います。ラベル ID は、所定の language に基づく ResourceManager の getLabel(id)を呼び出し、その結果を msglbl に登録します。getMsglbl()で取り出せます。ラベル ID とメッセージ ID は同時には登録できません。	
language	言語コードを指定します。この言語コードに基づいて、表示のラベルをリソースから作成します。	
optionAttributes	optionAttributes セット。JavaScript などの HTML 基本タグ以外の属性を、そのままタグとして使用します。そのときに、() は、(") に置き換えます。	

④使用例

```

<mis:writeCheck>
  <mis:submit value="copy" msg="MSG0035" accesskey="C"/>
  <mis:submit value="modify" msg="MSG0036" accesskey="M"/>
  <mis:submit value="delete" msg="MSG0037" accesskey="D"/>
  <mis:submit value="dbset" msg="XXMSG01" accesskey="X"
    command="NEW" gamenId="../TEST01" />
  <jsp:directive.includefile="../common/Excel_direct.jsp"/><br/>
</ mis:writeCheck>

```

63. mis:responseHeader

①概要

形式	<mis:responseHeader … />
body	なし

②説明

HttpServletResponse オブジェクトに対して、キャッシュコントロールするヘッダー情報をセットするタグです。Cache-Control、Content-Type、Location、Refresh、status、redirect をセットできます。

③属性

属性	意味	必須
cacheControl	レスポンスヘッダ に、Cache-Control の値を設定します。	
contentType	レスポンスヘッダ に、content-Type の値を設定します。	
refresh	レスポンスヘッダ に、refresh の値を設定します。	
refreshURL	レスポンスヘッダ に、refresh の値を設定するときに、指定の URL をロードします。	
redirect	指定された URL へ一時的なリダイレクトレスポンスを送信します。	
status	ステータスコードを設定します。	
location	レスポンスヘッダ に、location の値を設定します。	

④使用例

64. mis:bodyHidden

①概要

形式	<ul style="list-style-type: none"> • <mis:hidden name="..." value="..." /> • <mis:hidden name="..." /> <jsp:text>...</jsp:text> </mis:hidden>
body	あり

②説明

隠しフィールドを作成する簡易タグです。

③属性

属性	意味	必須
name	名前を指定します。name 属性と names 属性は同時には登録できません。	
value	値を指定します。	

④使用例

例 1 次画面に PN という名前で{@PN}に入っている値を渡します。

```
<input:hidden name="PN" value="{@PN}" />
```

例 2 次画面で使用する PLSQL と引数を渡します。

```
<input:hidden name="name" value="UNIQ,LKIS,KBLOYA,KBLKO,KBNGK,HJO,KBLKNM" />
<input:hidden name="SQL" value="{ call RKP0011B.RK0011B(?,?,?,?)}" />
```

↓

```
<mis:plsSqlUpdate
  command = "{@command}"
  names   = "{@names}"
  dbType  = "RK0011ARG"
  queryType = "JDBCPLSQL" >
```

```
<jsp:text>{@SQL}</jsp:text>
```

```
</mis:plsSqlUpdate>
```

例 3 次画面に SQL という名前で Body に記述した値を渡します。

```
<mis:hidden name="SQL"
  <jsp:text>
    SELECT AAA
    FROM RK01
    WHERE UNIQ = ?
```

```
</jsp:text>  
</mis:hidden>
```

65. commonforward

①概要

形式	<mis:commonforward />
body	なし

②説明

フォワードキャッシュによりページ転送するタグです。

③属性

なし

④使用例

例 1 フォワードキャッシュによりページ転送します。
<mis:commonforward />

66. backGamen

①概要

形式	<mis:backGamen />
body	なし

②説明

戻るボタン/戻るリンクを表示するタグです。

③属性

属性	意味	必須
type	ボタンのタイプ(button,link)を指定します。 button と設定すると、戻るボタンに、link と設定すると、戻るリンクになります。デフォルトは、戻るリンク(link)です。ここでは、button,link のみを許可します。	
command	戻る時に指定する command を設定できます。通常は、RENEW で戻ります。(デフォルト RENEW なので設定不要です。)	
gamenId	戻り先の画面をセットします。通常は、自動的に、BACK_GAMENID の値がセットされますが、先祖に戻る(画面A⇒画面B⇒画面C のときに、画面Aに戻る)場合や、別の画面に、進む場合に、直接指定します。(デフォルトは、来た画面:BACK_GAMENID ですので、設定不要です。)	
keys	リンク先に渡すキーを指定します。戻る時に、検索時のキャッシュに指定した引数以外に指定したり、別の値に置き換えたりする場合のキーを設定できます。カンマ区切りで複数指定できます。vals 属性には、キーに対応する値を、設定してください。	
vals	names 属性に対応する値を CVS 形式で複数指定します。キーに設定した値を、カンマ区切り文字で複数して出来ます。指定順序は、キーと同じにしておいて下さい。	
href	リンク先の URL を指定します。	
target	リンク先の表示ターゲットを指定します。リンク先の文書を表示させるフレーム、またはウィンドウの名前を指定します。	
id	id セット。	
lang	lang セット。	
dir	dir セット。	
title	title セット。	
style	style セット。	
tabindex	tabindex セット。	

属性	意味	必須
accesskey	accesskey セット。ボタンのショートカットキーを表示します。デフォルトでは、R に設定されています。	
clazz	class セット。Java の言語使用上、class で作成できないため、代用として clazz を使用している。html で作成される属性は、class で作成される。	
msg	メッセージを指定します。メッセージ ID は、所定の language に基づく ResourceManager の getMessage(id) を呼び出し、その結果を msglbl に登録します。getMsglbl() で取り出せます。ラベル ID とメッセージ ID は同時には登録できません。	
lbl	ラベルを指定します。ラベルを変更するときに、lbl 属性を使います。ラベル ID は、所定の language に基づく ResourceManager の getLabel(id) を呼び出し、その結果を msglbl に登録します。getMsglbl() で取り出せます。ラベル ID とメッセージ ID は同時には登録できません。	
language	言語コードを指定します。この言語コードに基づいて、表示のラベルをリソースから作成します。	

④使用例

例 1 通常は何も設定しません。

```
<mis:backGamen />
```

例 2 msg 属性に、リレフに表示するメッセージを指定できます。

```
<mis:backGamen msg="MSG0049" />
```

例 3 ボタンで表示させることができます。

```
<mis:backGamen type="button" />
```

例 4 ボタンで表示させて、msg 属性でボタンのラベルを指定できます。

この例では、ショートカットキーも指定しています。

```
<mis:backGamen type="button" msg="MSG0049" accesskey="R" />
```


第 III 部

Taglib属性リファレンス

『信用とは人に好かれること、約束を守ること、人を儲けさせること』

本田宗一郎（ホンダ技研創業者）

ここでは、第二部－第2章－3. Taglib属性一覧で説明した、Taglibの属性について、内容を特性ごとに説明します。

構成は次のとおりです。

第4章 カラム属性

カラム属性について説明します。

第5章 ViewFormクラス

view タグの viewFormType 属性に設定することができる ViewForm クラスについて説明します。

第6章 Queryクラス

query タグ、plsqliUpdate タグ、queryOption タグの queryType 属性に設定することができる Query クラスについて説明します。

第7章 Writer/Readerクラス

Writer/Readerクラスについて説明します。

Web
Web
アプリケーション

第4章 カラム属性

この章では、カラム属性について説明します。

1. 表示種別

画面に表示するときの型を指定します。

DBCColumnResource ファイルのレンダラー部や、columnEditor タグの Renderer 属性に指定することのできる値です。

項目	意味
LABEL	テキスト表示。DBCCell の具象クラスで、カラムのデータをそのまま文字列として表示する場合に使用するクラスです。カラムの表示に必要な属性は、DBCColumn オブジェクトより取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。
MENU	メニュー表示。DBCCell の具象クラスで、カラムのデータをコードリソースに対応したラベルで表示する場合に使用するクラスです。カラムの表示に必要な属性は、DBCColumn オブジェクトより取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。
NUMBER	カンマ編集。DBCCell の具象クラスで、カラムのデータを数字表示する場合に使用するクラスです。カラムの表示に必要な属性は、DBCColumn オブジェクトより取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。
YMD	YYYY/MM/DD。DBCCell の具象クラスで、カラムのデータを日付(年/月/日)表示する場合に使用するクラスです。カラムの表示に必要な属性は、DBCColumn オブジェクトより取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。
YM	YYYY/MM。DBCCell の具象クラスで、カラムのデータを日付(年/月)表示する場合に使用するクラスです。カラムの表示に必要な属性は、DBCColumn オブジェクトより取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。
PN	XXX-XXXXX-XXX 表示。DBCCell の具象クラスで、カラムのデータを品番情報として表示する場合に使用するクラスです。これは、特殊な表示で、11桁の文字列を3-5-3表示します。カラムの表示に必要な属性は、DBCColumn オブジェクトより取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。
COLUMN	DBCCell の具象クラスで、カラムのデータを表示する場合に使用するクラスです。カラムの表示に必要な属性は、DBCColumn オブジェクトより取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。

項目	意味
FORM	DBCellRenderer_FORM は、DBCell の具象クラスで、パラメータで指定された FORM を表示するクラスです。元の Value を、\$1 として、使用することが可能です。カラムの表示に必要な属性は、DBCColumn オブジェクト より取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。
HMS	DBCell の具象クラスで、カラムのデータを時:分:秒に分けて表示する場合に使用するクラスです。カラムの表示に必要な属性は、DBCColumn オブジェクト より取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。
HTML	DBCell の具象クラスで、HTML タグを含むデータを表示する場合に使用するクラスです。クロスサイトスクリプティング問題に対応するフィールドに対して 定義することにより、エスケープ処理を行います。カラムの表示に必要な属性は、DBCColumn オブジェクト より取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。
MONEY	DBCell の具象クラスで、カラムのデータを金額表示する場合に使用するクラスです。カラムの表示に必要な属性は、DBCColumn オブジェクト より取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。
PASSWD	DBCellRenderer_PASSWD は、DBCell の具象クラスで、カラムのデータをパスワード情報として表示する場合に使用するクラスです。パスワード表示とは、すべての表示を、***** で表示します。カラムの表示に必要な属性は、DBCColumn オブジェクト より取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。
PRE	DBCell の具象クラスで、カラムのデータをそのまま PRE 表示する場合に使用するクラスです。カラムの表示に必要な属性は、DBCColumn オブジェクト より取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。
QUERY	DBCell の具象クラスで、パラメータで指定された SQL 文を実行し、その結果を表示するクラスです。元の Value を、\$1 として、使用することが可能です。カラムの表示に必要な属性は、DBCColumn オブジェクト より取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。
YMDH	DBCell の具象クラスで、カラムのデータを日時(年/月/日 時:分:秒)表示する場合に使用するクラスです。カラムの表示に必要な属性は、DBCColumn オブジェクト より取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。

2. 入力種別

入力時の型を指定します。

DBCColumnResorce ファイルのエディター部や、columnEditorタグの editor 属性に指定することができる値です。

項目	意味
TEXT	テキスト表示。DBCCell の具象クラスで、カラムのデータをテキストフィールドで編集する場合に使用するクラスです。カラムの表示に必要な属性は、DBCColumn オブジェクト より取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。
MENU	メニュー表示。DBCCell の具象クラスで、カラムのデータをコードリソースに対応したプルダウンメニューで編集する場合に使用するクラスです。カラムの表示に必要な属性は、DBCColumn オブジェクト より取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。
TEXTAREA	テキストエリア表示。DBCCell の具象クラスで、カラムのデータをテキストエリアで編集する場合に使用するクラスです。カラムの表示に必要な属性は、DBCColumn オブジェクト より取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。
COLUMN	DBCCell の具象クラスで、カラムのデータを編集する場合に使用するクラスです。カラムの表示に必要な属性は、DBCColumn オブジェクト より取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。
HTML	DBCCell の具象クラスで、HTML タグを含むデータを編集する場合に使用するクラスです。カラムの表示に必要な属性は、DBCColumn オブジェクト より取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。
NUMBER	DBCCell の具象クラスで、カラムのデータを数字編集する場合に使用するクラスです。カラムの表示に必要な属性は、DBCColumn オブジェクト より取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。
PASSWD	DBCCell の具象クラスで、パスワード情報として編集する場合に使用するクラスです。パスワード編集とは、すべての入力値を、***** に置き換えて表示します。カラムの表示に必要な属性は、DBCColumn オブジェクト より取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。
SEQMENU	DBCCell の具象クラスで、DBCCellEditor_MENU を継承しています。このメニューは、正方向にしか選べません。指定した値より小さな値は、メニューとして表示しないようにします。これにより、シーケンスにステータスを順に挙げていくような、メニューを作成することができます。(逆に戻れないメニュー)カラムの表示に必要な属性は、DBCColumn オブジェクト より取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。

項目	意味
YM	DBCellEditor_YM は、DBCell の具象クラスで、カラムのデータを日付(年/月)編集する場合に使用するクラスです。カラムの表示に必要な属性は、DBCColumn オブジェクトより取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。
YMD	DBCell の具象クラスで、カラムのデータを日付(年/月/日)編集する場合に使用するクラスです。カラムの表示に必要な属性は、DBCColumn オブジェクトより取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。
YMDH	DBCell の具象クラスで、カラムのデータを日時(年/月/日 時:分:秒)編集する場合に使用するクラスです。カラムの表示に必要な属性は、DBCColumn オブジェクトより取り出します。このクラスは、DBCColumn オブジェクト毎に1つ作成されます。

3. データベースタイプ

カラムの物理属性を指定します。

DBCColumnResorce ファイルの DBType 部や、columnEditor タグの dbtype 属性に指定することができる値です。

DBType	意味
K	全角。文字列のカラム属性を規定するオブジェクトです。DBType_K は、全角のみの Char または Varchar2 属性に対応するクラスです。
R	小数(XXXX.YY の場合は、6.2 と表す)。数字型文字列のカラム属性を規定するオブジェクトです。DBType_R は、半角数字の NUMBER 属性に対応するクラスで '0' ~ '9', '-' , '.' のみ 構成されている数字型文字列カラムです。さらに、カンマ','が含まれていても OK とします。ただし、データからは取り除きます。
S9	半角数字('0' ~ '9', '-'). 数字型文字列のカラム属性を規定するオブジェクトです。DBType_S9 は、半角数字の NUMBER 属性に対応するクラスで '0' ~ '9', '-' のみ 構成されている数字型文字列カラムです。さらに、カンマ','が含まれていても OK とします。ただし、データからは取り除きます。
X	半角文字(半角カナ含まず)。文字列のカラム属性を規定するオブジェクトです。DBType_X は、半角の Char または Varchar2 属性に対応するクラスで 一般的な制限のない文字列カラムです。
XH	半角文字(半角カナ含む)。文字列のカラム属性を規定するオブジェクトです。DBType_XH は、半角文字+半角カタカナに対応するクラスで 大文字小文字の一般的な制限のない文字列カラムです。
X9	数字型文字列('0' ~ '9', '-'). 数字型文字列のカラム属性を規定するオブジェクトです。DBType_X9 は、半角数字の NUMBER 属性に対応するクラスで '0' ~ '9', '-' のみ 構成されている数字型文字列カラムです。さらに、カンマ','が含まれていても OK とします。ただし、データからは取り除きます。

第4章 カラム属性

DBType	意味
XK	半角/全角混在。文字列のカラム属性を規定するオブジェクトです。DBType_XK は、半角/全角混在の Char または Varchar2 属性に対応するクラスで 一般的な制限のない文字列カラムです。
XL	半角小文字。半角小文字の文字列のカラム属性を規定するオブジェクトです。DBType_XL は、半角小文字の英数字の CHAR または VARCHAR2 属性に対応するクラスです。半角カタカナは対象外です。
XU	半角大文字。半角大文字の文字列のカラム属性を規定するオブジェクトです。DBType_XU は、半角大文字の英数字の CHAR または VARCHAR2 属性に対応するクラスです。半角カタカナは対象外です。
YM01	YYYYMM01 または、00000000 , 99999999。文字列の開始日付属性のカラムを規定するオブジェクトです。DBType_YM01 は、半角の Char または Varchar2 属性に対応するクラスで YYYYMM01 または、00000000 , 99999999 を許可します。また、入力データとして、YYYYMM 形式 は、許可しません。あらかじめ、valueSet(String) でデータを変換しておきます。必要があります。ただし、日付としての整合性チェックは行いません。
MD5	MessageDigest により、MD5 でハッシュした文字を規定するオブジェクトです。DBType_MD5 は、パスワード情報など、重要な情報のハッシュコードを求めて、直接値の比較を行わない 用にすることができます。
PN	品番情報のカラム属性を規定するオブジェクトです。DBType_PN は、3-5-3品番情報を判定します。また、それぞれのフィールドにおいて、使用可能文字(例えば、Rev 文字列の制限)などを考慮しています。
XHU	文字列のカラム属性を規定するオブジェクトです。DBType_XHU は、半角文字+半角カタカナに対応するクラスで、大文字のみに制限された文字列カラムです。ホスト送信用なので、半角カタカナ小文字は使用できません。
YM31	文字列の終了日付属性のカラムを規定するオブジェクトです。DBType_YM31 は、半角の Char または Varchar2 属性に対応するクラスで YYYYMM31 または、00000000 , 99999999 を許可します。また、入力データとして、YYYYMM 形式 は、許可しません。あらかじめ、valueSet(String) でデータを変換しておきます。必要があります。ただし、日付としての整合性チェックは行いません。
YMD	文字列の日付属性(年/月/日)のカラムを規定するオブジェクトです。DBType_YMD は、半角の Char または Varchar2 属性に対応するクラスで YYYYMMDD に対応している必要があります。ただし、日付の整合性チェックは行いませんが、valueAdd(String value)による 日付の加算時には正式な日付で加算されます。
YMDH	文字列の日付属性(年/月/日 時:分:秒)のカラムを規定するオブジェクトです。DBType_YMDH は、半角の Char または Varchar2 属性に対応するクラスで YYYYMMDDHHMMSS に対応している必要があります。ただし、日付の整合性チェックは行いませんが、valueAdd(String value)での 日付の加算時には、正式な日付データにて加算します。

第5章 ViewFormクラス

この章では、view タグの viewFormType 属性に設定することができる ViewForm クラスについて説明します。このクラスは、mis.pdm.hayabusa.html 以下の ****ViewForm クラスの **** 部分に該当します。

項目	意味
Calendar	HTMLTextFieldViewForm を継承した カレンダー表示クラスです。1ヶ月分のカレンダー形式で、検索結果を表示します。AbstractViewForm により、setter/getter メソッドのデフォルト実装を提供しています。各 HTML のタグに必要な setter/getter メソッドのみ、追加定義しています。AbstractViewForm を継承しているため、ロケールに応じたラベルを出力させることができます。
CustomTable	HTMLTableViewForm を継承した カスタムテーブル作成クラスです。ヘッダー情報、フッター情報、ボディー情報を指定することで、自由にテーブル表示のレイアウトが可能です。AbstractViewForm により、setter/getter メソッドのデフォルト実装を提供しています。各 HTML のタグに必要な setter/getter メソッドのみ、追加定義しています。AbstractViewForm を継承しているため、ロケールに応じたラベルを出力させることができます。
Dynamic	HTMLTableViewForm を継承した 動的カラム一覧表示クラスです。DBTableModel から 各フィールド情報を動的にカラムを作成して表示します。AbstractViewForm により、setter/getter メソッドのデフォルト実装を提供しています。各 HTML のタグに必要な setter/getter メソッドのみ、追加定義しています。AbstractViewForm を継承しているため、ロケールに応じたラベルを出力させることができます。
Format Table	HTMLTableViewForm を継承した フォーマットテーブル作成クラスです。ヘッダー部分のフォーマット情報に応じたテーブルを自動作成します。AbstractViewForm により、setter/getter メソッドのデフォルト実装を提供しています。各 HTML のタグに必要な setter/getter メソッドのみ、追加定義しています。AbstractViewForm を継承しているため、ロケールに応じたラベルを出力させることができます。
Format TextField	HTMLTextFieldViewForm を継承した テキストフィールド作成クラスです。フォーマットを外部から指定することにより、自由にレイアウトを作成できます。AbstractViewForm により、setter/getter メソッドのデフォルト実装を提供しています。各 HTML のタグに必要な setter/getter メソッドのみ、追加定義しています。AbstractViewForm を継承しているため、ロケールに応じたラベルを出力させることができます。

項目	意味
Table	AbstractViewForm を継承した テーブル作成クラスです。DBTableModel から HTML 環境で取り扱うための、テーブルタグを自動生成します。AbstractViewForm により、setter/getter メソッドのデフォルト実装を提供しています。各 HTML のタグに必要な setter/getter メソッドのみ、追加定義しています。AbstractViewForm を継承しているため、ロケールに応じたラベルを出力させることができます。
TextField	AbstractViewForm を継承した テキストフィールド作成クラスです。DBTableModel から HTML 環境で取り扱うための、テキストフィールドタグを自動生成します。AbstractViewForm により、setter/getter メソッドのデフォルト実装を提供しています。各 HTML のタグに必要な setter/getter メソッドのみ、追加定義しています。AbstractViewForm を継承しているため、ロケールに応じたラベルを出力させることができます。
TreeBOM	HTMLTableViewForm を継承した ツリーテーブル表示クラスです。DBTableModel のレベルから、JavaScript のツリー階層を作成し、ツリー階層を持ったテーブル表示を行います。AbstractViewForm により、setter/getter メソッドのデフォルト実装を提供しています。各 HTML のタグに必要な setter/getter メソッドのみ、追加定義しています。AbstractViewForm を継承しているため、ロケールに応じたラベルを出力させることができます。

第6章 Queryクラス

この章では、query タグ、plsqlUpdate タグ、queryOption タグの queryType 属性に設定することができる Query クラスについて説明します。queryType 属性のデフォルトクラスには、JDBC が使用されます。

項目	意味
JDBC	java.sql.Statement を用いて、データベース検索処理を行います。引数は無しです。(与えられた SQL 文を実行します。) 内部変数の受け渡しのデフォルト実装は、AbstractQuery クラスを継承しているため、ここでは、execute() メソッドを実装しています。このクラスでは、ステートメント文を execute() する事により、データベースを 検索した結果を DBTableModel に割り当てます。
JDBCCallable	java.sql.CallableStatement を用いて、データベース検索処理を行います。引数は、従来の PL/SQL の実行が可能ないように、第一引数はエラーコード、第二引数は、 エラーメッセージを返してきます。第三引数以降は、自由に指定できます。内部変数の受け渡しのデフォルト実装は、AbstractQuery クラスを継承しているため、ここでは、execute() メソッドを実装しています。このクラスでは、ステートメント文を execute() することにより、データベースを 検索した結果を DBTableModel に割り当てます。
JDBCErrMsg	java.sql.CallableStatement を用いて、データベース検索処理を行います。引数を配列指定で渡すことができ、エラー時には、DBErrMsg オブジェクトにエラー情報を 格納して返すことが可能です。内部変数の受け渡しのデフォルト実装は、AbstractQuery クラスを継承しているため、ここでは、execute() メソッドを実装しています。このクラスでは、ステートメント文を execute() することにより、データベースを 検索した結果を DBTableModel に割り当てます。
JDBCKeyEntry	java.sql.CallableStatement を用いて、データベース検索処理を行います。引数に、キーと値をセットで配列指定で渡すことができ、エラー時には、DBErrMsg オブジェクトにエラー情報を格納して返すことが可能です。内部変数の受け渡しのデフォルト実装は、AbstractQuery クラスを継承しているため、ここでは、execute() メソッドを実装しています。このクラスでは、ステートメント文を execute() することにより、データベースを 検索した結果を DBTableModel に割り当てます。

項目	意味
JDBCPLSQL	<p>java.sql.CallableStatement を用いて、データベース検索処理を行います。引数に、SYSARG_ARRAY と、USERARG_ARRAY を配列指定で渡すことができ、エラー時には、DBErrMsg オブジェクトにエラー情報を格納して返すことが可能です。内部変数の受け渡しのデフォルト実装は、AbstractQuery クラスを継承しているため、ここでは、execute() メソッドを実装しています。このクラスでは、ステートメント文を execute() することにより、データベースを 検索した結果を DBTableModel に割り当てます。</p>
JDBCPrepared	<p>java.sql.PreparedStatement を用いて、データベース検索処理を行います。引数に、指定した値を配列で渡します。内部変数の受け渡しのデフォルト実装は、AbstractQuery クラスを継承しているため、ここでは、execute() メソッドを実装しています。このクラスでは、ステートメント文を execute() することにより、データベースを 検索した結果を DBTableModel に割り当てます。</p>
JDBCUpdate	<p>java.sql.CallableStatement を用いて、データベース登録処理を行います。引数は、そのまま配列に格納して処理を行います。エラー時の処理や、検索結果の 取り出しはできません。内部変数の受け渡しのデフォルト実装は、AbstractQuery クラスを継承しているため、ここでは、execute() メソッドを実装しています。このクラスでは、ステートメント文を execute() することにより、データベースを 検索した結果を DBTableModel に割り当てます。</p>

第7章 Writer/Readerクラス

この章では、Writer/Readerクラスについて説明します。

1. DBTableWriter

DBTableWriter クラスは、writeTable タグの writerClass 属性に設定できるクラスです。

項目	意味
Chart	DBTableModel インターフェース のオブジェクトを PrintWriter を用いて出力するための、共通インターフェースです。
CSV	DefaultDBTableWriter を継承していますので、ラベル、名前、データの出力部のみ オーバーライドして、可変長カンマ区切り文字ファイルの出力機能を実現しています。
Default	ラベル、名前、データの出力部のみオーバーライドすれば、各種出力フォーマットに合わせた サブクラスを実現することが可能です。このクラスは、可変長タブ区切り文字ファイルの出力機能を実現しています。
Fixed	DefaultDBTableWriter を継承していますので、ラベル、名前、サイズ、データの 出力部のみオーバーライドして、固定長文字ファイルの出力機能を実現しています。なお、固定長出力されるのは、データ部のみで、ラベル、名前、サイズは、separator で指定された区切り記号で連結されて出力されます。
PDFXML	現在サポートされていません。
Properties	PropertiesDBTableWriter は、DBTableWriter インターフェース の実装クラスです。DefaultDBTableWriter を継承していますので、ラベル、名前、データの出力部のみ オーバーライドして、プロパティファイルの出力機能を実現しています。
XML	DefaultDBTableWriter を継承していますので、ラベル、名前、データの出力部のみ オーバーライドして、XML 形式ファイルの出力機能を実現しています。出力の XML 形式は、ORACLE XDK での出力ファイルと同じ形式ですので、直接データベースに 登録することができます。
T	TDBTableWriter は、DBTableWriter インターフェース の実装クラスです。DefaultDBTableWriter を継承して、データの出力部のみオーバーライドして、文字列カラム(クラス名 VARCHAR2 のカラム)に、=T("値")という文字列で出力する ファイルの出力機能を実現しています。
Renderrer	RenderrerDBTableWriter は、DBTableWriter インターフェース の実装クラスです。DefaultRenderrerDBTableWriter を継承して、データの出力部のみオーバーライドして、テーブルデータではなく、表示文字列(レンダラー情報)を出力します。

2. DBTableReader クラス

DBTableReader クラスは、readTable タグの readerClass 属性に設定できるクラスです。

項目	意味
Default	DefaultDBTableReader は、DBTableReader インターフェース のデフォルト実装クラスです。名前、データの入力部のみオーバーライドすれば、各種入力フォーマットに合わせたサブクラスを実現することが可能です。

3. ChartDBTableWriter クラス

ChartDBTableWriter クラスは、writeChart タグの writerClass 属性に設定できるクラスです。

項目	意味
SVGBar	SVGBarChartDBTableWriter は、SVGDBTableWriter インターフェース の実装クラスです。AbstractSVGDBTableWriter を継承していますので、writeData, writeChartArea, を、オーバーライドして、棒グラフを実現しています。入力データは、汎用的な、DBTableModel です。
SVGCircle	SVGCircleChartDBTableWriter は、SVGDBTableWriter インターフェース の実装クラスです。AbstractSVGDBTableWriter を継承していますので、writeData, writeChartArea, を、オーバーライドして、円グラフを実現しています。入力データは、汎用的な、DBTableModel です。
SVGLine	SVGLineChartDBTableWriter は、SVGDBTableWriter インターフェース の実装クラスです。AbstractSVGDBTableWriter を継承していますので、writeData, writeChartArea, を、オーバーライドして、折れ線グラフを実現しています。入力データは、汎用的な、DBTableModel です。

索引

A

and9, 18, 80
appear9, 18, 75
authenticate.jsp 5

B

backGamen12, 23, 126
bodyHidden12, 22, 124
button8, 15, 43

C

Calendar 135
calendar.css 5
Chart 139
ChartDBTableWriter クラス 140
check.jsp 5
column7, 14, 26
COLUMN130, 132
columnCheck9, 18, 84
columnEditor 11, 20, 106
columnEditor 130, 132, 133
columnMarker 10, 19, 95
columnSet8, 16, 54
comment11, 20, 110
common 4, 5
commonforward12, 22, 125
CSV 139
custom 4, 6
custom.css 6
CustomTable 135

D

DBCColumnResorce 133
DBTableReader クラス 140
DBTableWriter クラス 139
debug4, 6, 8, 17, 62

Default 139, 140
default.css 5
default.js 5
dir.jsp 6
Dynamic 135

E

encode.jsp 5
entry 8, 16, 53
entryQuery 10, 19, 90
equals 9, 17, 63
error.jsp 5
errorMessage 10, 19, 93
errorMessage.jsp 5
Excel_direct.jsp 5
Excel_insert.jsp 5
Excel_update.jsp 5

F

favoriteLink11, 20, 108
file 10, 19, 97
fileOption10, 19, 101
fileQuery 10, 19, 98
fileWhere10, 20, 101
Fixed 139
FORM 131
FormatTable 135
FormatTextField 135
forward 10, 19, 94
frame 9, 18, 86
fw_menu.js 6

G

GmisTree.js 6

H

headermenu.jsp 6

headermenu2.jsp..... 6
 help..... 10, 18, 89
 helpMenu.jsp..... 6
 hidden.....8, 17, 61
 HMS..... 131
 HTML.....131, 132
 HTML.jsp..... 5
 htмлend.jsp..... 5

I

img.....10, 20, 102
 init.jsp..... 6
 input.....7, 14, 29

J

JDBC..... 137
 JDBCCallable..... 137
 JDBCErrMsg..... 137
 JDBCKeyEntry..... 137
 JDBCPLSQL..... 138
 JDBCPrepared..... 138
 JDBCUpdate..... 138
 jsp_init.jsp..... 5

K

K..... 133

L

LABEL..... 130
 ldapQuery.....11, 21, 116
 link.....8, 16, 58
 log.jsp..... 5
 logout.html..... 6

M

mailReceiver.....9, 17, 74
 mailSender.....9, 17, 73
 MD5..... 134
 menu..... 4, 6
 MENU.....130, 132
 menu.jsp..... 6

menu1.jsp..... 6
 menu2.jsp..... 6
 message..... 9, 17, 64
 MONEY..... 131

N

NUMBER..... 130, 132

O

option..... 8, 14, 37
 orderby.....8, 15
 orderBy.....41

P

PASSWD..... 131, 132
 PDFXML..... 139
 plsqlUpdate.....9, 17, 70, 137
 PN..... 130, 134
 popupmenu.jsp..... 6
 PRE..... 131
 prev_next.jsp..... 5
 printConfig.jsp..... 6
 printContext.jsp..... 6
 printHeader.jsp..... 6
 printPageContext.jsp..... 6
 printRequesMethodt.jsp..... 6
 printRequest.jsp..... 6
 printSettion.jsp..... 6
 Properties..... 139

Q

query.....8, 15, 47, 137
 QUERY..... 131
 query_info.jsp..... 6
 queryOption.....8, 14, 40, 137
 Query クラス..... 137

R

R..... 133
 readTable.....9, 18, 76, 140
 regQuery.....11, 22, 119

Renderer..... 139
report..... 11, 21, 112
requestCache..... 10, 19, 92
DBColumnResource..... 130
responseHeader..... 12, 22, 123, 124
result_info.jsp..... 6

S

S9..... 133
schedule..... 11, 21, 115
select..... 7, 14, 34
SEQMENU..... 132
session_init.jsp..... 5
shell..... 10, 19, 92
space..... 9, 17, 68
sql..... 10, 20, 104
start_jsp_init.jsp..... 5
statusBar..... 11, 21, 111
statusBar.js..... 5
stopFile.html..... 6
submit..... 11, 22, 23, 121
submitClearQuery.jsp..... 5
submitFieldQuery.jsp..... 5
submitQuery.jsp..... 5
submitSelectQuery.jsp..... 5
submitSet.jsp..... 5
SVGBar..... 140
SVGCircle..... 140
SVGLine..... 140

T

T..... 139
Table..... 136
tbody..... 10, 20, 105
text..... 10, 18, 88
TEXT..... 132
textarea..... 8, 15, 45
TEXTAREA..... 132
TextField..... 136
tfoot..... 11, 20, 105

thead..... 11, 20, 106
TreeBOM..... 136
treeBOM.js..... 5

U

update..... 8, 16, 56
upload..... 8, 16, 57
userInfo..... 9, 18, 85

V

value..... 11, 21, 114
view..... 8, 15, 50, 135
ViewForm クラス..... 135
viewLink..... 9, 18, 82
viewMarker..... 10, 19, 96

W

where..... 9, 18, 78
writeChart..... 11, 20, 109, 140
writeCheck..... 9, 17, 75
writeTable..... 9, 17, 66, 139

X

X..... 133
X9..... 133
XH..... 133
XHU..... 134
XK..... 134
XL..... 134
XML..... 139
XU..... 134

Y

YM..... 130, 133
YM01..... 134
YM31..... 134
YMD..... 130, 133, 134
YMDH..... 131, 133, 134

