

Webアプリケーション開発者ガイド(実践編)

リリース 1.0
初版:2003 年 9 月



*Muratec Information
Systems.LTD.*

Webアプリケーション開発者ガイド(実践編)、リリース 1.0

原本部品番号:W1B0201-01

原本名: Hayabusa Web Application Developer's Guide (Practice)、Release 1

原本著者: 長谷川 和彦

デザイン: 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	画面作成の基礎	1
第1章	画面作成の基礎	2
	1. 画面の名称	2
	2. 標準画面パターン	3
	3. 標準画面パターンの使用方法	4
	4. 標準JSP	5
	5. 仕様書作成用部品	6
	6. 共通ルール	7
	7. 画面作成ルール	8
第 II 部	標準画面パターン	13
第2章	メニュー画面	14
	1. メニュー画面	14
第3章	TypeS: 条件検索	15
	1. 検索画面	15
	2. 検索結果	18
第4章	Type1: 一覧検索一覧登録	21
	1. 検索結果	21
	2. 複写処理	23
	3. 変更処理	25
	4. 削除処理	27
	5. DB 登録処理	28
第5章	Type2: 一覧検索明細登録	29
	1. 検索結果	29
	2. 複写処理	30
第6章	Type3: 検索エントリー	31
	1. エラー画面	31
	2. 明細エントリー処理	32
第7章	Type4: その他	34
第 III 部	例題と演習課題	35
第8章	実習環境設定	36
第9章	TypeS: 条件検索	38
	課題1 SQL 検索	38
	課題2 リソースファイル作成	38
	課題3 リンク、マーカー作成	39
	課題4 PL/SQL 検索	39
第10章	Type1: 一覧検索一覧登録	40

課題5 PL/SQL 登録	40
課題6 EXCEL 出力/EXCEL 取込	40
課題7 カスタムボタン作成	40
第11章 Type3: 検索エントリー	41
課題8 PL/SQL 登録	41

はじめに

このマニュアルでは、Webアプリケーションの画面開発に関する機能について説明します。また、タイプ別画面サンプルを利用した演習を通して、開発技術を習得します。

《対象読者》

このマニュアルは、新しいアプリケーションを開発したり、既存のアプリケーションを実行できるように変換するプログラマを対象にしています。このマニュアルは、システム・アナリスト、プロジェクト管理者およびデータベース・アプリケーション開発に関心がある方にも役立ちます。

このマニュアルは、アプリケーション・プログラミングの実践的な知識があり、Structured Query Language (SQL) を使用してリレーショナル・データベース・システムの情報にアクセスできることを前提としています。また、このマニュアルの特定の項では、オブジェクト指向プログラミングの基本的な概念を理解していることも前提としています。

《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

画面作成の基礎

『これを知るをこれを知るとなし、
これを知らざるを知らずとなせ。これ知るなり。』
孔子

(知っていること、知らないことが
明確にできることが、知るという行ためである。)

ここでは、Webアプリケーションで画面を開発するための基本事項について説明します。
構成は、次のとおりです。

第 1 章 画面作成の基礎

Web アプリケーションを開発するにあたり、基本となる画面の名称や画面作成パターンを説明します。

Web
Web
アプリケーション

第1章 画面作成の基礎

この章では、Web アプリケーションを開発するにあたり、基本となる画面の名称や画面作成パターンを説明します。

1. 画面の名称

JSP 画面は、MENU フレーム、CONTENTS フレーム(QUERY フレーム、RESULT フレーム)に分かれています。

①MENU フレーム

メニューを表示するエリアです。メニューの項目をクリックすると、右側のコンテンツフレームに、選択したメニューの内容が表示されます。

②CONTENTS フレーム

メニューフレームで選択した項目の内容を表示するエリアです。各メニューごとに、独自にフレームを分割することができます。通常は、QUERY フレームと RESULT フレームに分割して使用します。

③QUERY フレーム

検索条件や表示順を指定するエリアです。
この条件により、RESULT フレームが表示されます。

④RESULT フレーム

QUERY フレームの検索条件をもとに、結果を表示するエリアです。
登録処理のボタンやメニューを選択すると、同じフレームが登録画面に変わります。



2. 標準画面パターン

MENU フレームはリソースファイルより標準のメニュー画面が自動作成されるので、特に開発の必要はありません。

CONTENTS フレームを開発する際には、標準画面パターンの雛型をベースに開発を行います。

雛形は、検索/登録方法により5パターンを用意しています。

ここでは、その5つの標準画面パターンそれぞれについて概要を説明します。

各標準画面パターンの開発方法の詳細については、『第 II 部 標準画面パターン』を参照ください。

①TypeS: 条件検索

検索結果を表示するのみのパターンの画面です。データの登録(追加、複写、変更、削除などの処理)はできません。

②Type1: 一覧検索一覧登録

検索結果を一覧で表示し、一覧形式のままデータを登録するパターンの画面です。複数行を一回の処理で登録できるため、最も効率的な登録方法で、Excel(CSV)形式でのファイル登録など、応用が可能です。エラーチェックは、登録行全てのチェックを一回の登録作業で行います。このパターンは基本となる検索/登録画面といえます。まずは、この形式で設計を考えてください。

③Type2: 一覧検索明細登録

検索結果を一覧で表示し、そこで選択したデータのみを表示させて、1件ずつ登録するパターンの画面です。マスター登録など、登録項目が多い場合に利用します。

但し、本当にそれだけ多くの項目を登録する必要があるのか、設計を十分に検討してから利用してください。この形式を安易に使用することは、推奨しません。

④Type3: 検索エントリー

検索結果に対して、直接エントリー登録するパターンの画面です。Type2の明細登録に近いのですが、絞り込み登録の必要がない、エントリーデータ(例えば、伝票IDをバーコードリーダーで読み取るなど)の場合は、アクションを減らすことが可能になります。検索後直接エントリーするため、基本的には、『あれば更新なければ追加』となります。

⑤Type4: その他

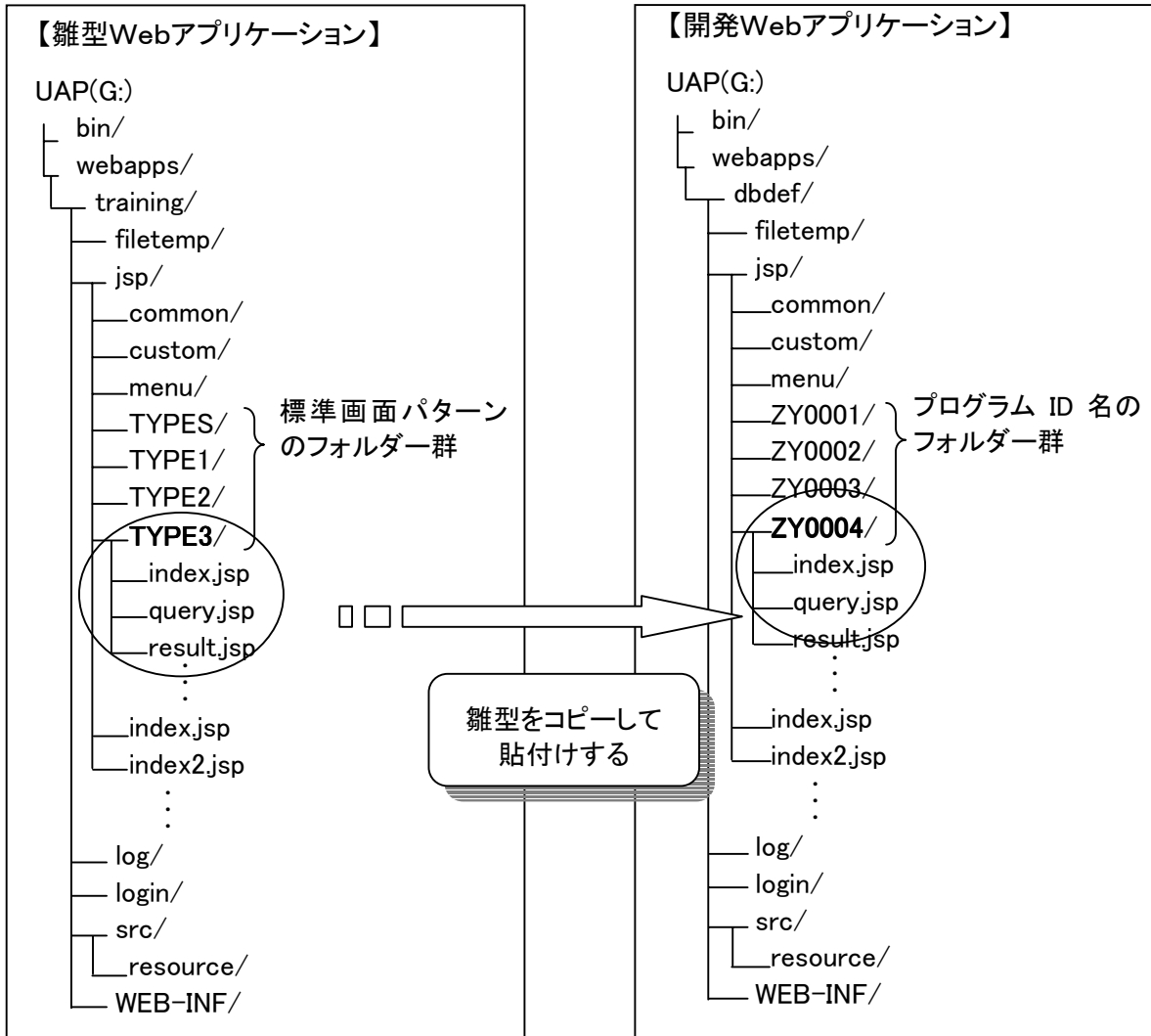
上記以外のパターンで開発した画面を、Type4 とします。

3. 標準画面パターン の使用方法

Web アプリケーションには、ひとつのメニュー単位にプログラム ID 名のフォルダが用意されています。開発者は、使用する標準画面パターンの雛型フォルダの JSP ファイルをコピーして、作成したいプログラム ID のフォルダ内に貼付けます。

これで準備が整いました。雛型 JSP をカスタマイズすれば、画面を作成することができます。

例) TYPE3雛型フォルダのjspファイルをプログラム ID ZY0004 のフォルダにコピーする。



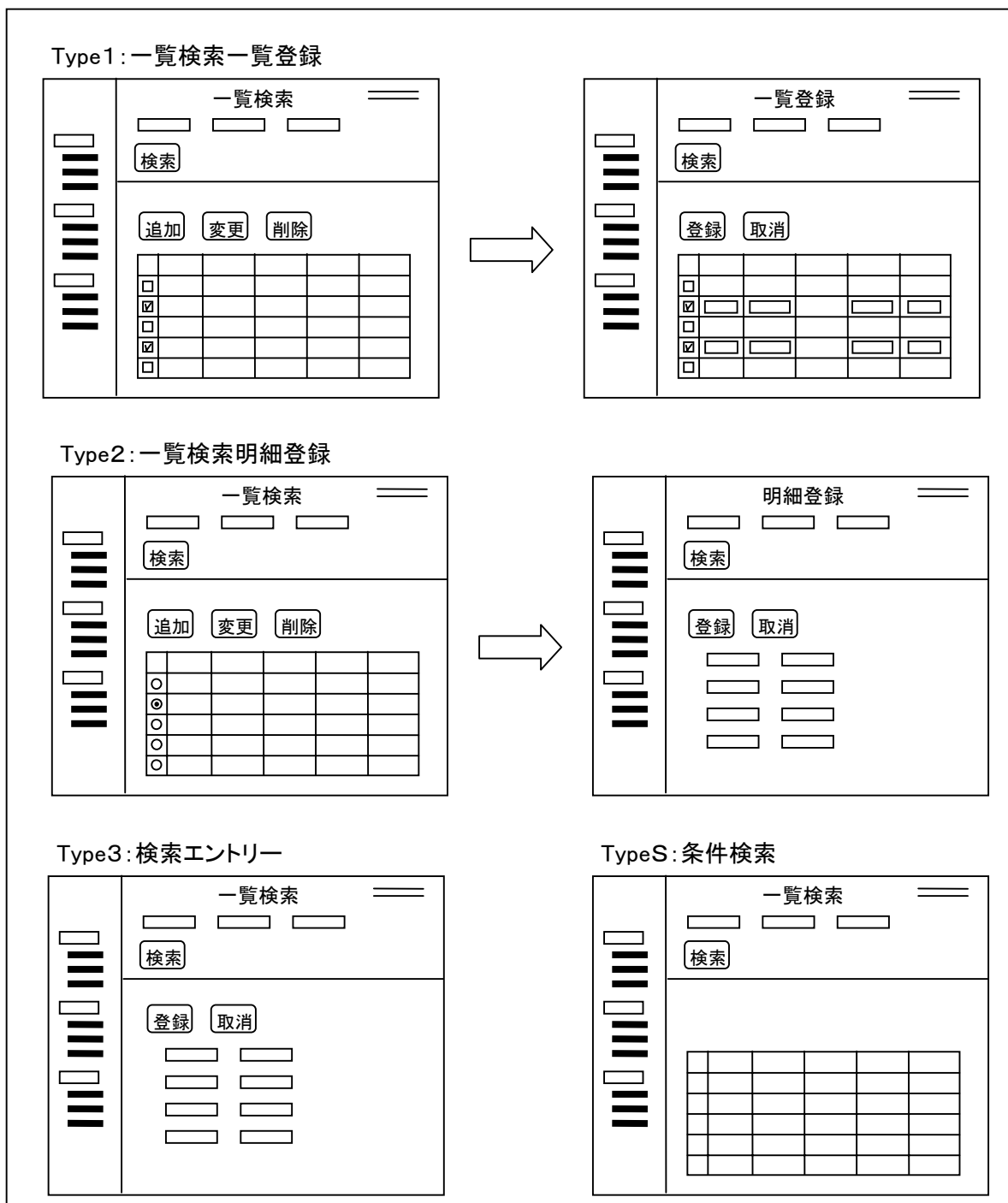
4. 標準JSP

標準画面パターン用のJSPには主に次のようなファイルがあります。
各jspファイルの詳細については、『第 II 部 標準画面パターン』を参照ください。

- index.jsp : トップ画面 (フレーム分割用)
- query.jsp : 検索入力画面
- result.jsp : 結果表示画面
- copy.jsp : 複写画面
- modify.jsp : 変更画面
- delete.jsp : 削除画面
- insert.jsp : 追加画面
- forward.jsp : 振分用 (書換え不要)
- dummy.html : result フレームに初期表示するダミー画面

5. 仕様書作成用部品

標準画面パターンを利用して仕様書を作成する場合に不可欠な 画面遷移用部品は次のとおりです。これを利用して、画面遷移を作成します。



6. 共通ルール

Web アプリケーションフレームワークを利用して、アプリケーションを開発するにあたり、以下の共通ルールを定めます。

1. 標準画面パターンの分類番号を設計仕様書に明記してください。
2. JSP 画面では、独自のスクリプトレット (Java のプログラム) の使用を禁止します。すべての画面処理は、フレームワーク側で用意したファイルか、タグリブを使用してください。
3. JSP 画面で使用する HTML は、XHTML 準拠とします。
4. JSP 画面上に直接メッセージやラベルを書かないでください。フレームワークのリソースを使用して、かならずタグリブ経由で、HTML タグを作成してください。
5. JSP 画面上では、色、文字フォント、文字サイズ、右寄せなどの情報は、すべて共通の CSS で設定するようにできています。デザインを変える場合は、custom フォルダの CSS ファイルを修正してください (基本的に、画面中に直接スタイルに関わる属性を設定しないこと)。
6. 入力チェックや DB 関連の項目については、デフォルトで DB 定義書よりリソースファイルを作成し、フレームワークで自動的にチェックします。どうしても画面固有の表示方法が必要な場合は、タグリブを用いて、表示形式を外部から書き換えるようにしてください。
7. このフレームワークは、将来的に拡張する方向で考えているため、個別に機能を増やすと拡張時の互換性が損なわれ、拡張メリットを受けられなくなる恐れがあるので、できる限り標準的な方法で開発してください。また、独自機能を作成した場合は、フレームワーク開発担当者に連絡してください (標準機能として取り込む可能性があるため)。
8. 将来の拡張性案
 - ・XML 化による画面情報のピックアップ ⇒ 保守仕様書の自動作成
 - ・XML 化による画面情報の自動作成 ⇒ 設計仕様書からの画面作成
 - ・画面を XHTML タグから XML + XSLT での運用への移行
 - ・携帯端末等の各種 Web バージョンへのコンテンツ自動変換
 - ・DB 定義からのリソース自動作成 / 自動登録
 - ・Java 登録系プログラムの自動作成
 - ・画面実行処理でのログ出力、自動解析、集計よりパフォーマンスの調整
 - ・画面の自動テスト / パフォーマンス自動測定
 - ・標準パターンからの画面サンプル自動作成

7. 画面作成ルール

標準的な画面の作成方法を説明します。すべての画面で共通して適用されるルールをピックアップしています。

< 画面例① >

1. メニューはリソースファイルより自動作成されます。

2. 画面名称、画面 ID、ログイン時刻、担当者 ID、担当者名の情報は、フレームワーク側で設定します。

3. 検索条件設定は、DB 定義のラベルリソースより自動作成されます。

4. 検索ボタン、BACK、NEXT ボタン等はフレームワーク側で設定します。

[query.jsp]

```
<?xml version="1.0" encoding="UTF-8" ?>
<jsp:root
  xmlns:jsp="http://java.sun.com/JSP/Page"
  xmlns:mis="mis-taglib.tld"
  version="1.2">
<jsp:directive.page language="java" contentType="text/html; charset=UTF-8"
/>
<jsp:directive.include file="../../custom/query_info.jsp" />
```

.....

2. QUERY 画面の
ヘッダー情報

```
<form method="GET" action="forward.jsp" target="RESULT">
<table summary="layout" >
<tr>
  <mis:column name="SYSTEM_ID" defaultVal="{@SYS.SYSTEM_ID}"
              must="true" />
  <mis:column name="LANG" />
  <mis:column name="KBSAKU" />
</tr>
<tr>
  <mis:select name="CLM" >
    <mis:option value="" lbl="" />
    <mis:queryOption value="{@CLM}" >
      <jsp:text>
        SELECT CLM,CLM FROM GE08 GROUP BY CLM ORDER BY 1
      </jsp:text>
    </mis:queryOption>
  </mis:select>
  <mis:column name="NAME_JA" />
  <mis:select name="IMAGE" lbl="IMAGE_VIEW" >
    <mis:option value="true" lbl="true" selected="selected" />
    <mis:option value="false" lbl="false" />
  </mis:select>
</tr>
<tr>
  <mis:column name="LABEL_NAME" />
  <mis:select name="ORDER_BY" msg="ORDER_BY">
    <mis:option value="SYSTEM_ID,CLM,LANG" lbl="CLM" selected="selected" />
    <mis:option value="SYSTEM_ID,LANG,NAME_JA" lbls="LANG,NAME_JA" />
    <mis:option value="SYSTEM_ID,LANG,LABEL_NAME" lbls="LANG,LABEL_NAME" />
  </mis:select>
  <mis:space />
</tr>
</table>
<jsp:directive.include file="../common/submitClearQuery.jsp" />
</form>

<jsp:directive.include file="../common/htmlend.jsp" />
</jsp:root>
```

3. 検索条件設定
(コラム情報)

4. QUERY 画面の検索ボタン

<画面例②>

5. 検索結果は、queryタグより自動的に表示されます。登録時の処理は、entry.jspで行います。

6. DBの検索は、表示形式とは無関係で同一の方法で行う。

7. テーブル形式の検索結果の表示は、ラベルをテーブル上に並べて表示します。

8. フィールド形式の検索結果の表示は、ラベルを並べて表示します。

[result.jsp]


```

<mis:query command="{@command}" debug="false">
  <jsp:text>
    select CLM,NAME_JA,LABEL_NAME,KBSAKU,SYSTEM_ID,LANG,FGJ
    from GE08
  </jsp:text>
  <mis:where>
    <mis:and value = "FGJ      in ('0','1')      />
    <mis:and value = "SYSTEM_ID like '{@SYSTEM_ID}%' />
    <mis:and value = "LANG     like '{@LANG}%'     />
    <mis:and value = "CLM      like '{@CLM}%'     />
    <mis:and value = "NAME_JA  like '{@NAME_JA}%'  />
    <mis:and value = "LABEL_NAME like '{@LABEL_NAME}%' />
    <mis:and value = "KBSAKU   =   '{@KBSAKU}'    />
  </mis:where>
  <mis:appear startKey = "order by" value = "{@ORDER_BY}"
    defaultVal = "SYSTEM_ID,CLM,LANG" />
</mis:query>
    
```

5、6

5. 検索結果の表示

.

6. 検索方法は、表示方法とは分離している。

[テーブル表示の場合]

```
<mis:view
  viewFormType = "HTMLTable"
  command      = "{@command}"
  writable     = "false"
  noDisplay    = "FGJ"
  checked      = "{@checked}"
  startNo     = "{@startNo}"
  pageSize    = "{@pageSize}" />
```

7. テーブル表示

[フィールド表示の場合]

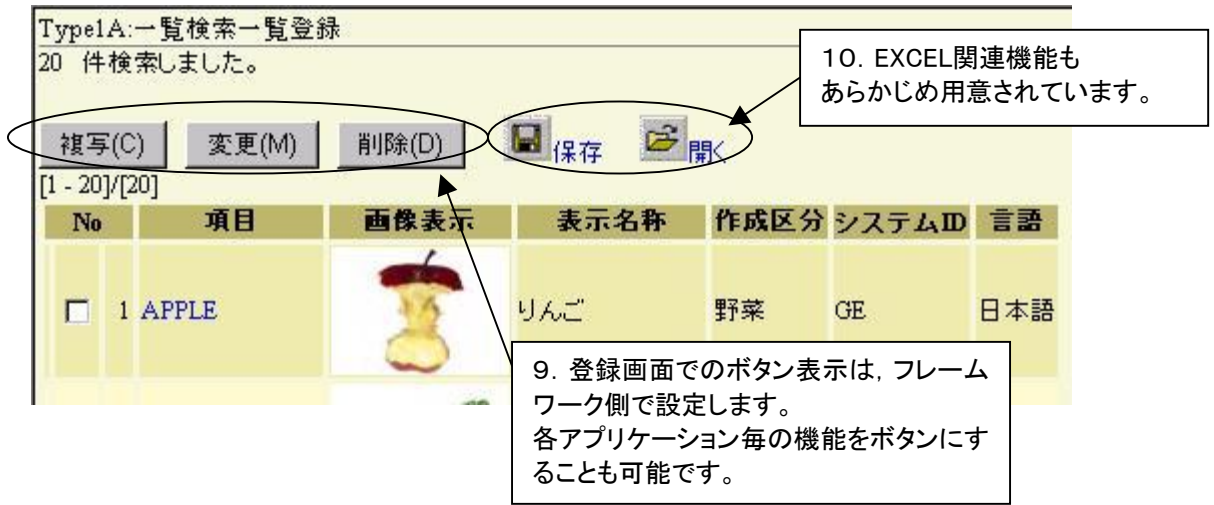
```
<mis:view
  viewFormType = "HTMLTextField"
  command      = "{@command}"
  writable     = "false"
  noDisplay    = "FGJ"
  checked      = "{@checked}"
  startNo     = "{@startNo}"
  pageSize    = "{@pageSize}" />
```

8. フィールド表示

[カスタマイズ表示の場合]

```
<mis:view
  viewFormType = "HTMLFormatTable"
  command      = "{@command}"
  noDisplay    = "FGJ"
  checked      = "{@checked}"
  startNo     = "{@startNo}"
  pageSize    = "{@pageSize}"
  selectedType = "radio"
  rowspan     = "3" >
  <tr>
    <td>[SYSTEM_ID]</td>
    <td>[LANG]</td>
    <td>[KBSAKU]</td>
    <td rowspan="3">[NAME_JA]</td>
  </tr>
  <tr><td colspan="3">[CLM]</td></tr>
  <tr><td colspan="3">[LABEL_NAME]</td></tr>
</mis:view>
```

<画面例③>



[result.jsp]

.....

```

<form onSubmit="return oneClick();" method="POST" action="forward.jsp"
target="RESULT">
<mis:writeCheck>
  <mis:submit value="copy"      lbl="COPY"      accesskey="C" />
  <mis:submit value="modify"    lbl="MODIFY"    accesskey="M" />
  <mis:submit value="delete"    lbl="DELETE"    accesskey="D" />
  <jsp:directive.include file="../common/Excel_direct.jsp" /><br />
</mis:writeCheck>
<mis:view
  viewFormType = "HTMLTable"
  command      = "{@command}"
  noDisplay    = "FGJ"
  checked      = "{@checked}"
  startNo     = "{@startNo}"
  pageSize    = "{@pageSize}" />
</form>
<jsp:directive.include file="../common/htmlend.jsp" />
</jsp:root>

```

9. 登録ボタン

10. EXCEL 関連機能

.....

第 II 部

標準画面パターン

『朝（あした）に道を聞かば、夕（ゆうべ）に死すとも可なり。』
孔子

（立派な教えを聞いたなら、その日の内に死んでも悔いはない。）

Webアプリケーションを開発するにあたり、メニュー画面及び標準画面パターン4+1通りを用意しています。これらのパターンは基本であり、このパターンの応用として、標準画面パターンに準拠する形で開発を行い、生産性の向上を図ります。パターンが少ないのは、各パーツの簡単な修正で応用パターンを作り出せるためです。ここでは、各標準画面パターンについて説明していきます。

構成は次のとおりです。

第2章 メニュー画面
メニュー画面について説明します。

第3章 TypeS:条件検索
検索のみの画面パターンについて説明します。

第4章 Type1:一覧検索一覧登録
基本となる検索/登録画面のパターンについて説明します。

第5章 Type2:一覧検索明細登録
検索結果を絞り込み、1件ずつ登録する画面のパターンについて説明します。

第6章 Type3:検索エントリー
検索結果に対して、直接エントリー登録を行う画面のパターンについて説明します。

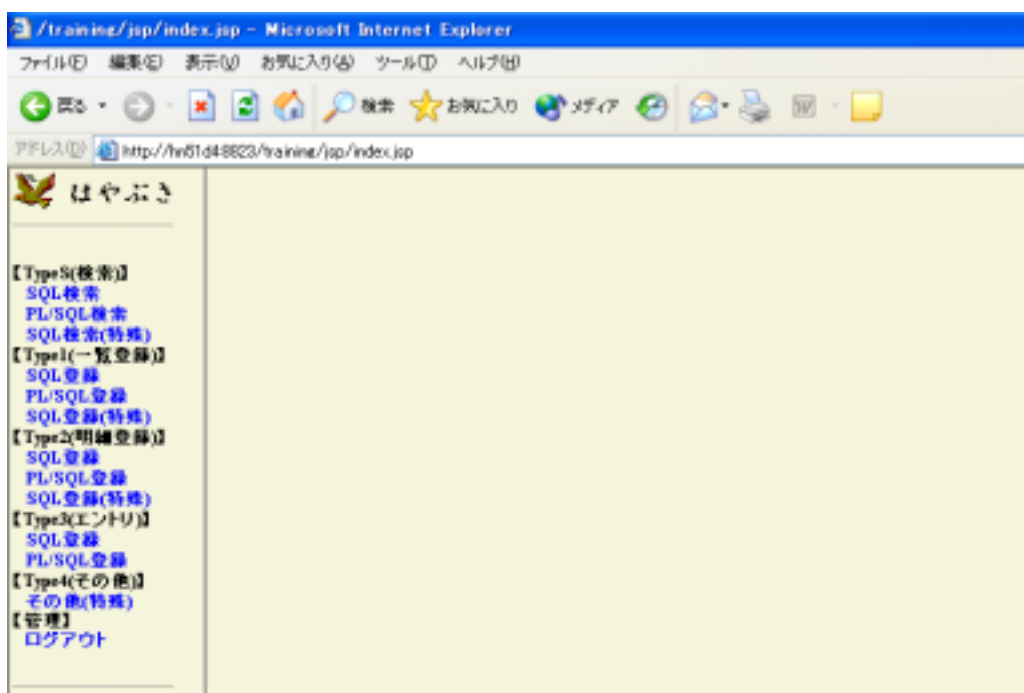
第7章 Type4:その他
上記以外のパターンの特殊な画面のパターンについて説明します。

Web
Web
アプリケーション

第2章 メニュー画面

この章では、メニュー画面について説明します。

1. メニュー画面

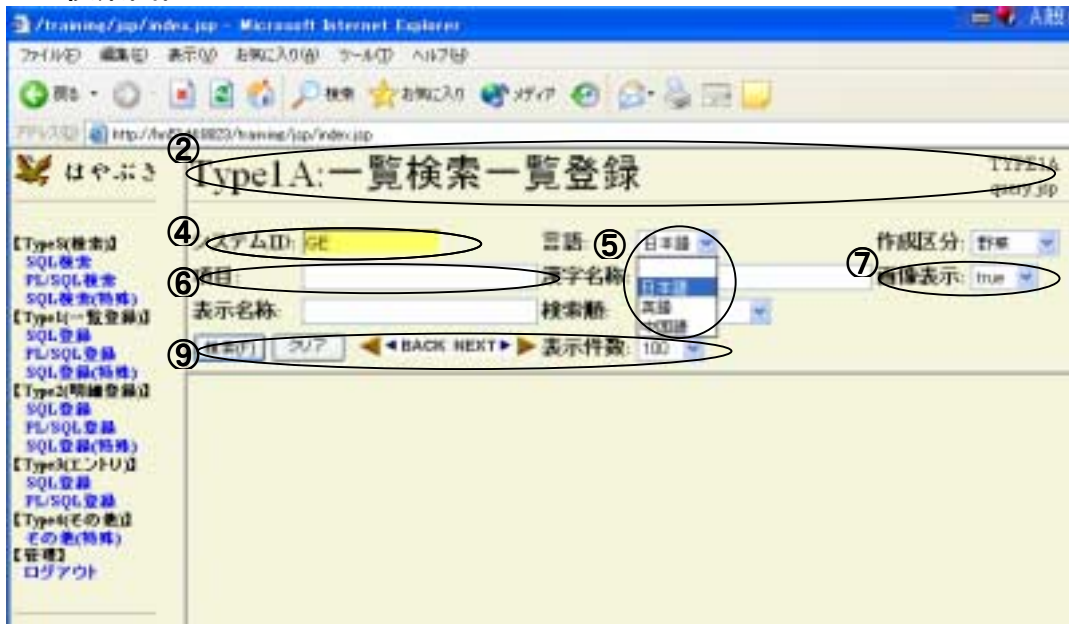


メニュー画面は、jsp/menu/menu.jsp により、GUIResourceより読み取った画面一覧より、ログインユーザーのロールに応じたアクセス制御で読み取り可能なメニューが表示されます。この、menu.jsp の種類に応じて、JavaScriptによるツリー階層メニューや、フォルダ階層のメニューを表示させることが可能です。

第3章 TypeS:条件検索

この章では、検索のみの画面パターンについて説明します。

1. 検索画面



[TYPESA/query.jsp]

- ①JSPの宣言部。画面上には現れません。エンジンでは、将来の拡張性を考慮し、JSPファイルは、すべてXML形式で記述しています。
- ②タイトル表示。GUIResource のLNAME属性(名前(長))を表示します。右端には、ログインユーザー名や時刻、GAMENID番号やJSPファイル名も出しています。実体は、custom/query_info.jsp というファイルをインクルードして共有しています。
- ③JSPファイル毎に修正履歴を記述します。このコメント内容は、DBに取り込み可能です。仕様書に記述されたバージョンと変更タイトル、内容等を記述します。
- ④カラム情報を記述します。Must="true"とすることで、画面上必須入力を表す色に変化します。実際の null チェックは、result.jsp ファイルで行っています。
- ⑤LANGは、CodeResource で、プルダウンの要素をあらかじめ定義しています。
- ⑥CLMは、データベースの内容から、プルダウンメニューを作成しています。
- ⑦IMAGEは、直接プルダウンメニューを作成しています。
- ⑧検索条件の指定です。lbls を使うと複数のラベル名を同時に記述できます。
- ⑨検索、クリアや、BACK,NEXT,pageSize などの共通機能を、インクルードして使用しています。

[TYPESA/query.jsp]

```

<?xml version="1.0" encoding="UTF-8" ?>
<jsp:root
  xmlns:jsp="http://java.sun.com/JSP/Page"
  xmlns:mis="mis-taglib.tld"
  version="1.2">
<jsp:directive.page language="java" contentType="text/html; charset=UTF-8" />
<jsp:directive.include file="../custom/query_info.jsp" />

<mis:comment
  title = "新規作成"
  version = "001"
  date = "2003/07/08"
  author = "K.HASEGAWA"
  text = "Type1 パターンの雛型を新規作成します。"
/>

<form method="GET" action="forward.jsp" target="RESULT">
<table summary="layout" >
<tr>
  <mis:column name="SYSTEM_ID" defaultVal="{@SYS.SYSTEM_ID}" must="true" />
  <mis:column name="LANG" />
  <mis:column name="KBSAKU" />
</tr>
<tr>
  <mis:select name="CLM" >
    <mis:option value="" lbl="" />
    <mis:queryOption value="{@CLM}" >
      <jsp:text>
        SELECT CLM,CLM FROM GE08 GROUP BY CLM ORDER BY 1
      </jsp:text>
    </mis:queryOption>
  </mis:select>
  <mis:column name="NAME_JA" />
  <mis:select name="IMAGE" lbl="IMAGE_VIEW" >
    <mis:option value="true" lbl="true" selected="selected" />
    <mis:option value="false" lbl="false" />
  </mis:select>
</tr>

```

①

②

③

④

⑤

⑥

⑦

```
<tr>
  <mis:column name="LABEL_NAME" />
  <mis:select name="ORDER_BY" msg="ORDER_BY">
    <mis:option value="SYSTEM_ID,CLM,LANG" lbl ="CLM" selected="selected" />
    <mis:option value="SYSTEM_ID,LANG,NAME_JA" lbls="LANG,NAME_JA" />
    <mis:option value="SYSTEM_ID,LANG,LABEL_NAME" lbls="LANG,LABEL_NAME" />
  </mis:select>
  <mis:space />
</tr>
</table>
<jsp:directive.include file="../../common/submitClearQuery.jsp" />
</form>

<jsp:directive.include file="../../common/htmlend.jsp" />
</jsp:root>
```

⑧

⑨

2. 検索結果

①

No	項目	④ 画像表示	表示名称	作成区分	システムID	言語
⑤ 1	APPLE	⑥ 	りんご	野菜	GE	日本語
2	ASPARAGUS		アスパラガス	野菜	GE	日本語
3	BANANA		バナナ	野菜	GE	日本語

⑦

[TYPESA/result.jsp]

- ①タイトル表示。GUIResource のLNAME属性(名前(長))を表示します。Query.jsp と比べ、簡易的な表示になります。右端には、GAMENIDとファイル名を表示しています。GAMENID には、お気に入りリンクとして、直接検索結果を得られるリンクが表示されます。実体は、custom/result_info.jsp というファイルをインクルードして共有しています。
- ②カラムチェックは、通常は画面には現れません。入力条件が DBColumnResource の指定範囲外の場合に、エラーメッセージを表示します。また、nullCheck で、引数が nullかどうかの判断も行います。(must 属性との自動連携はしていません。)
- ③検索条件を元に、実際にデータベースを検索します。検索結果は、DBTableModel というオブジェクトに一時保管されます。
- ④フレームワーク外ですが、columnEditor を用いて、結果データのラベルを入れ替えています。(equal タグは、単に引数が画像である場合に、true としています。)
- ⑤viewLink で、引数のパラメータに応じたリンクを動的に作成しています。[カラム名]で、DBTableModel 中のデータを取り出しています。
- ⑥viewMaker で、カラムにマーキングしています。この例の場合では、img タグに置き換え画像表示を実現しています。
- ⑦view を用いて、先ほど検索した DBTableModel を表示しています。この例では、viewFormType= “HTMLTable”を用いて、表形式で表示しています。


```

<?xml version="1.0" encoding="UTF-8" ?>
<jsp:root
  xmlns:jsp="http://java.sun.com/JSP/Page"
  xmlns:mis="mis-taglib.tld"
  version="1.2">
<jsp:directive.page language="java" contentType="text/html; charset=UTF-8"
/>
<jsp:directive.include file="../custom/result_info.jsp" />
<mis:comment
  title   = "新規作成"
  version = "001"
  date    = "2003/07/08"
  author  = "K.HASEGAWA"
  text    = "TypeS パターンの雛型を新規作成します。"
/>
<mis:columnCheck
  command = "{@command}"
  columns = "SYSTEM_ID,LANG,CLM,NAME_JA,LABEL_NAME"
  nullCheck = "SYSTEM_ID"
/>
<mis:query command="{@command}" >
  <jsp:text>
    select CLM,NAME_JA,LABEL_NAME,KBSAKU,SYSTEM_ID,LANG,FGJ
    from GE08
  </jsp:text>
  <mis:where>
    <mis:and value = "FGJ          in ('0','1')"          />
    <mis:and value = "SYSTEM_ID   like '{@SYSTEM_ID}%"    />
    <mis:and value = "LANG        like '{@LANG}%"         />
    <mis:and value = "CLM         like '{@CLM}%"          />
    <mis:and value = "NAME_JA     like '{@NAME_JA}%"       />
    <mis:and value = "LABEL_NAME  like '{@LABEL_NAME}%"   />
    <mis:and value = "KBSAKU     =    '{@KBSAKU}'"        />
  </mis:where>
  <mis:appear startKey = "order by" value = "{@ORDER_BY}"
    defaultVal = "SYSTEM_ID,CLM,LANG" />
</mis:query>

```

①

②

③

```

<mis:equals val1="{@IMAGE}" val2="true" >
  <mis:columnEditor column="NAME_JA" label="IMAGE_VIEW" />
</mis:equals>

<mis:viewLink command="{@command}">
  <mis:link column="CLM" gameId="TYPESA"
    href="index.jsp" target="CONTENTS"
    keys="command,SYSTEM_ID,LANG,CLM,IMAGE"
    value="NEW,{@SYSTEM_ID},[LANG],[CLM},{@IMAGE}" />
</mis:viewLink>

<mis:viewMarker command="{@command}">
  <mis:columnMarker column="NAME_JA" onMark="{@IMAGE}" >
    
  </mis:columnMarker>
</mis:viewMarker>

<mis:view
  viewFormType = "HTMLTable"
  command      = "{@command}"
  writable     = "false"
  noDisplay    = "FGJ"
  checked      = "{@checked}"
  startNo      = "{@startNo}"
  pageSize     = "{@pageSize}" />

<jsp:directive.include file="../../common/htmlend.jsp" />
</jsp:root>

```

④

⑤

⑥

⑦

[TYPESB/result.jsp]

検索方法は、SELECT 文以外に、PL/SQL でカーソルを返す方法も可能です。

```

<mis:query
  command      = "{@command}"
  names        = "SYSTEM_ID,LANG,CLM,NAME_JA,LABEL_NAME,KBSAKU"
  queryType    = "JDBCErrorMsg" >
  <jsp:text>
    { call TYPESB01.TYPESB01( ?,?,?,? ) }
  </jsp:text>
</mis:query>

```

③

第4章 Type1:一覧検索一覧登録

この章では、基本となる検索/登録画面のパターンについて説明します。

1. 検索結果



[TYPE1A/result.jsp]

- ①複製、変更、削除の各ボタンです。WriteCheck タグで囲まれた内容は、書き込み属性(GUIResource の RWMODE(rwrwrwr))で制御しています。ここでも、submit タグ
- ②EXCEL 出力のボタンです。共通処理の、ExcelOut が呼ばれます。
- ③view で行を選択するためのチェックボックスです。Type1:では、一覧検索・一覧登録なので、複数の行を同時にチェックを入れて処理することが可能です。

```

.
.
.
<form onSubmit="return oneClick();" method="POST" action="forward.jsp"
target="RESULT">
<mis:writeCheck>
  <mis:submit value="copy" lbl="COPY" accesskey="C" />
  <mis:submit value="modify" lbl="MODIFY" accesskey="M" />
  <mis:submit value="delete" lbl="DELETE" accesskey="D" />
  <jsp:directive.include file="../common/Excel_direct.jsp" /><br />
</mis:writeCheck>

<mis:view
  viewFormType = "HTMLTable"
  command = "{@command}"
  noDisplay = "FGJ"
  checked = "{@checked}"
  startNo = "{@startNo}"
  pageSize = "{@pageSize}" />
</form>
.
.
.
```

①

②

③

2. 複写処理



[TYPE1A/copy.jsp]

- ①message タグでメッセージを出します。メッセージ ID は、MessageResource に登録しておきます。comment に記述された値は使用されていません。
- ②登録、取消、クリアの各ボタンです。WriteCheck タグで囲まれた内容は、書き込み属性 (GUIResource の RWMODE(rwrwrwrw)で制御しています。ここでも、submit タグ
- ③EXCEL 取り込みのボタンです。先の EXCEL 出力で抜き出した値を修正後、このメニューで取り込むことが可能です。共通処理の、ExcelOut が呼ばれます。
- ④view が、HTMLTable の場合は、複数行を同時に編集可能です。
- ⑤次の entry.jsp に渡すための、カラム名列をカンマ区切りで作成しています。このカラム名の順番が、次の SQL 文の ? の順番に対応します。
- ⑥次の entry.jsp に渡すための、処理 SQL 文です。

```

        .
        .
        .
<mis:message msg="MSG0045" comment="複写処理を行います。" /> ①
<form onSubmit="return oneClick();" method="POST"
        action="forward.jsp" target="RESULT">
<mis:writeCheck>
    <mis:submit value="entry" lbl="ENTRY" accesskey="E" />
    <mis:submit value="reset" lbl="RESET" accesskey="R" />
    <mis:input type="reset" name="command" msg="MSG0011" td="false" />
    <jsp:directive.include file="../common/Excel_insert.jsp" /><br />
</mis:writeCheck>
<mis:view
    viewFormType = "HTMLTable"
    command      = "@command"
    noDisplay    = "FGJ"
    noWritable   = "FGJ"
    checked      = "@checked"
    startNo      = "@startNo"
    pageSize     = "@pageSize" />
<mis:hidden name="names"
    value="CLM,NAME_JA,LABEL_NAME,KBSAKU,SYSTEM_ID,LANG" />
<mis:hidden name="SQL" >
    <jsp:text>
        INSERT INTO GE08
            (CLM,NAME_JA,LABEL_NAME,KBSAKU,SYSTEM_ID,LANG,
            FGJ,DYSET,DYUPD,USRSET,USRUPD,PRGUPD)
        VALUES
            (?, ?, ?, ?, ?, ?,
            1, '@USER.YMDH', '@USER.YMDH', '@USER.ID', '@USER.ID', 'TYPE1A')
    </jsp:text>
</mis:hidden>
</form>
        .
        .
        .

```

②

③

④

⑤

⑥

3. 変更処理



[TYPE1A/modify.jsp]

- ①message タグで、メッセージを表示させます。
- ②view タグで、一覧表示形式でデータを表示します。
- ③次の entry.jsp に渡すための、カラム名列をカンマ区切りで作成しています。このカラム名の順番が、次の SQL 文の?の順番に対応します。
- ④次の entry.jsp に渡すための、処理 SQL 文です。

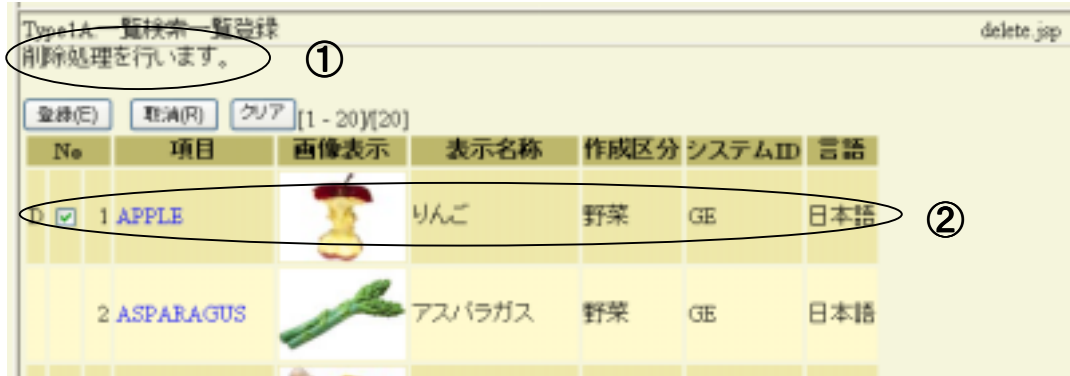
```

        .
        .
        .
<mis:message msg="MSG0046" comment="変更処理を行います。" /> ①
<form onSubmit="return oneClick();" method="POST" action="forward.jsp"
target="RESULT">
<mis:writeCheck>
    <mis:submit value="entry" lbl="ENTRY" accesskey="E" />
    <mis:submit value="reset" lbl="RESET" accesskey="R" />
    <mis:input type="reset" name="command" msg="MSG0011" td="false" />
    <jsp:directive.include file="../common/Excel_update.jsp" /><br />
</mis:writeCheck>

<mis:view
    viewFormType = "HTMLTable"
    command      = "{@command}"
    noDisplay    = "FGJ"
    noWritable   = "SYSTEM_ID,LANG,CLM"
    checked      = "{@checked}"
    startNo      = "{@startNo}"
    pageSize     = "{@pageSize}" /> ②

<mis:hidden name="names"
    value="NAME_JA,LABEL_NAME,KBSAKU,SYSTEM_ID,LANG,CLM" /> ③
<mis:hidden name="SQL" >
    <jsp:text>
        UPDATE GE08 SET
            NAME_JA      = ?,
            LABEL_NAME   = ?,
            KBSAKU       = ?,
            DYUPD        = '{@USER.YMDH}',
            USRUPD       = '{@USER.ID}',
            PRGUPD       = 'TYPE1A'
        WHERE SYSTEM_ID = ?
        AND   LANG      = ?
        AND   CLM       = ?
    </jsp:text>
</mis:hidden>
</form>
    
```


4. 削除処理



[TYPE1A/delete.jsp]

- ①message タグで、メッセージを表示させます。
- ②view タグで、一覧表示形式でデータを表示します。
- ③次の entry.jsp に渡すための、カラム名列をカンマ区切りで作成しています。このカラム名の順番が、次の SQL 文の？の順番に対応します。
- ④次の entry.jsp に渡すための処理 SQL 文です。

```

.
.
<mis:message msg="MSG0047" comment="削除処理を行います。" /> ——— ①
<form onSubmit="return oneClick();" method="POST" action="forward.jsp"
target="RESULT">
.
.
<mis:view
  viewFormType = "HTMLTable"
  command      = "{@command}"
  noDisplay    = "FGJ"
  startNo      = "{@startNo}"
  pageSize     = "{@pageSize}" /> } ②
<mis:hidden name="names" value="SYSTEM_ID,LANG,CLM" /> ——— ③
<mis:hidden name="SQL" >
  <jsp:text>
    DELETE FROM GE08
    WHERE SYSTEM_ID = ?
    AND   LANG      = ?
    AND   CLM       = ?
  </jsp:text>
</mis:hidden> } ④
</form>
.
.

```

5. DB 登録処理

DB登録処理は、画面はありません。すぐに forward されるためです。

[TYPE1A/entry.jsp]

- ①処理エラー時は、④まで処理が進みません。この form ボタンで元の画面に戻ります。
- ②カラムのチェックやリクエスト情報をDBTableModelに設定したりします。
- ③実際にデータベースに書き込むところです。names と SQL は、先の画面で引数として渡しています。
- ④正常にデータベース登録が終了すると、result.jsp に戻ります。(forward)ここで、command="RENEW"により、再検索(つまり先に検索した引数を用いてもう一度検索)します。

.....

```

<form method="POST" action="forward.jsp" target="RESULT">
<mis:input type="button" optionAttributes="onClick='history.back()' " ①
  msg="MSG0049" accesskey="R" td="false" />
</form>

<mis:columnCheck command="{@command}" />
<mis:entry command="{@command}" /> ②

<mis:update
  command = "{@command}"
  queryType = "JDBCPrepared"
  names = "{@names}" > ③
<jsp:text>{@SQL}</jsp:text>
</mis:update>

<mis:forward page="result.jsp" keys="command" vals="RENEW" /> ④
<jsp:directive.include file="../common/htmlend.jsp" />
</jsp:root>

```

[TYPE1B/entry.jsp]

INSERT/UPDATE 文の代わりに、PI/SQL で登録を行うことも可能です。

```

<mis:plsqlUpdate
  command = "{@command}"
  names = "SYSTEM_ID, LANG, CLM, NAME_JA, LABEL_NAME, KBSAKU, FGJ, USRSET" ③
  dbType = "GE08ARG"
  queryType = "JDBCPLSQL" >
  <jsp:text>{ call TYPE1B01.TYPE1B01( ?,?,? ) }</jsp:text>
</mis:plsqlUpdate>

```

第5章 Type2:一覧検索明細登録

この章では、検索結果を絞り込み、1件ずつ登録する画面のパターンについて説明します。

1. 検索結果



[TYPE2A/result.jsp]

①明細登録時には、1件だけ選ぶように、radio ボタンに設定します。

```

<form onSubmit="return oneClick();" method="POST" action="forward.jsp"
target="RESULT">
<mis:writeCheck>
  <mis:submit value="copy"    lbl="COPY"    accesskey="C" />
  <mis:submit value="modify"  lbl="MODIFY" accesskey="M" />
  <mis:submit value="delete"  lbl="DELETE" accesskey="D" />
  <jsp:directive.include file="../common/Excel_direct.jsp" /><br />
</mis:writeCheck>
<mis:view
  viewFormType = "HTMLTable"
  command      = "{@command}"
  noDisplay    = "FGJ"
  checked      = "{@checked}"
  startNo     = "{@startNo}"
  pageSize    = "{@pageSize}"
  selectedType = "radio" /> ①
</form>

```

2. 複写処理

[TYPE2A/copy.jsp]

①明細登録時には、HTMLTable ではなく、HTMLTextField の仲間を使用します。

```

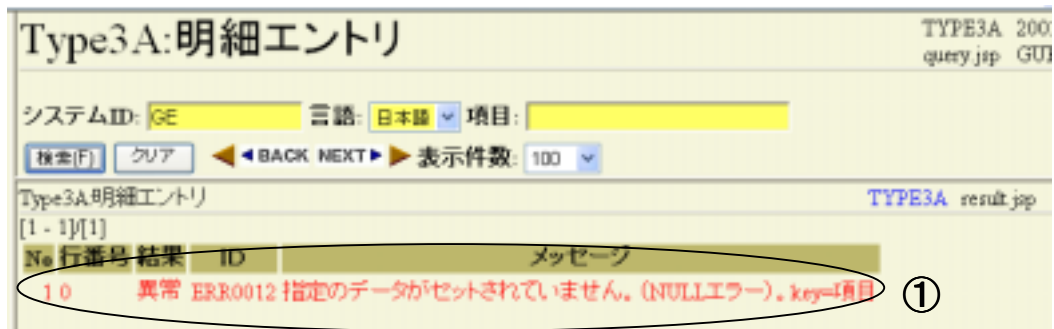
    .
    .
    .
<mis:view
    viewFormType = "HTMLTextField"
    command      = "{@command}"
    noDisplay    = "FGJ"
    noWritable   = "FGJ"
    checked      = "{@checked}"
    startNo      = "{@startNo}"
    pageSize     = "{@pageSize}"
    skip         = "true"
    />
    .
    .
    .
    
```

①

第6章 Type3:検索エントリー

この章では、検索結果に対して、直接エントリー登録を行う画面のパターンについて説明します。

1. エラー画面



[TYPE3A/result.jsp]

①columnCheck タグでエラー時の表示です。上記エラーは、must 項目である CLMが null だったケースです。

・
・
・

```
<mis:errorMessage command="{@command}" clear="{@clear}"
                    displayMsg="{@displayMsg}" />
```

```
<mis:columnCheck
  command = "{@command}"
  columns = "SYSTEM_ID,LANG,CLM"
  nullCheck = "SYSTEM_ID,LANG,CLM"
/>
```

} ①

・
・
・

2. 明細エントリー処理

The screenshot shows the 'Type3A:明細エントリー' (Type3A: Detail Entry) page. At the top right, it displays 'TYPE3A 200' and 'query.jsp GU'. The main form contains the following fields and controls:

- システムID: GE (text input)
- 言語: 日本語 (dropdown menu)
- 項目: APPLE (text input, circled with ①)
- 検索(F) (button)
- クリア (button)
- ◀ BACK NEXT ▶ (navigation buttons)
- 表示件数: 100 (dropdown menu)

Below the search controls, the page title is 'Type3A明細エントリー' and the URL is 'TYPE3A result.jsp'. The search result is displayed as:

1件検索しました。 (2A)

登録(E) (button) クリア (button) [1 - 1][1]

システムID: GE (text input) 言語: 日本語 (dropdown menu) 作成区分: 野菜 (dropdown menu)

項目: APPLE (text input)

漢字名称: リンゴ (text input)

表示名称: リンゴ (text input)

[TYPE3A/result.jsp]

①項目に既存の『APPLE』を入力した場合。

②A. 検索しましたと表示され、値が表示されています。

The screenshot shows the 'Type3A:明細エントリー' (Type3A: Detail Entry) page. At the top right, it displays 'TYPE3A 200' and 'query.jsp GU'. The main form contains the following fields and controls:

- システムID: GE (text input)
- 言語: 日本語 (dropdown menu)
- 項目: ORANGE (text input, circled with ①)
- 検索(F) (button)
- クリア (button)
- ◀ BACK NEXT ▶ (navigation buttons)
- 表示件数: 100 (dropdown menu)

Below the search controls, the page title is 'Type3A明細エントリー' and the URL is 'TYPE3A result.jsp'. The search result is displayed as:

1件検索しました。
追加処理を行います。 (2B)

登録(E) (button) クリア (button) [1 - 1][1]

システムID: GE (text input) 言語: 日本語 (dropdown menu) 作成区分: 野菜 (dropdown menu)

項目: ORANGE (text input)

漢字名称: (empty text input)

表示名称: (empty text input)

[TYPE3A/result.jsp]

①項目に新規の『ORANGE』を入力した場合。

②B. 追加処理をしますと表示され、初期値が設定されて表示されます。

[TYPE3A/result.jsp]

```

<mis:value command="SQL">
  <jsp:text>
    select count(*) CNT from GE08
    where FGJ      in ('0','1')
    and  SYSTEM_ID = '{@SYSTEM_ID}'
    and  LANG      = '{@LANG}'
    and  CLM       = '{@CLM}'
  </jsp:text>
</mis:value>

<mis:equals val1="{@CNT}" val2="0" >
  <mis:query command="NEW" >
    <jsp:text>
      INSERT INTO GE08
      (CLM,NAME_JA,LABEL_NAME,KBSAKU,
       SYSTEM_ID,LANG,FGJ,DYSET,
       DYUPD,USRSET,USRUPD,PRGUPD)
      VALUES
      ('{@CLM}','{@NAME_JA}','{@LABEL_NAME}','{@KBSAKU}',
       '{@SYSTEM_ID}','{@LANG}','1','{@USER.YMDH}',
       '{@USER.YMDH}','{@USER.ID}','{@USER.ID}','TYPE3A')
    </jsp:text>
  </mis:query>
</mis:equals>

<mis:equals val1="{@CNT}" val2="0" notEquals="true" >
  <mis:query command="NEW" >
    <jsp:text>
      UPDATE GE08 SET
        NAME_JA      = '{@NAME_JA}',
        LABEL_NAME   = '{@LABEL_NAME}',
        KBSAKU       = '{@KBSAKU}',
        DYUPD        = '{@USER.YMDH}',
        USRUPD       = '{@USER.ID}',
        PRGUPD       = 'TYPE1A'
      WHERE SYSTEM_ID = '{@SYSTEM_ID}'
      AND  LANG      = '{@LANG}'
      AND  CLM       = '{@CLM}'
    </jsp:text>
  </mis:query>
</mis:equals>

<mis:forward page="result.jsp" keys="command,SYSTEM_ID,LANG,CLM"
  vals="RENEW,{@SYSTEM_ID},{@LANG},{@CLM}" />
<jsp:directive.include file="../common/htmlend.jsp" />
</jsp:root>

```

①
マスターテーブルの
キー存在チェック

②A

②B

第7章 Type4:その他

この章では、特殊な画面パターンについて説明します。

画面の具体例は提示しません。

この Type は、基本的には先の3タイプに含まれないケースすべてを示しています。

画面設計時には、先の3パターン(+1検索パターン)でまず考え、どうしても当てはまらないケースのみ、タイプ4として分けて考えてください。

あくまで、画面設計時の目安(標準外のための工数見積りへの影響等)のための分類とお考え下さい。



コラム

その他の分類が、なぜ、TYPE4 になったか？

分類方法として、色々考えることができます。

たとえば、TYPEZ や TYPE9 をその他としておき、間を空き番にするなどです。

そうすれば、新しいタイプが発生したとき、順次増やしていくことが可能になります。

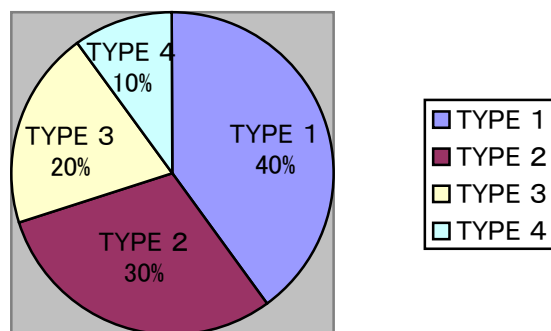
では、なぜ、そうせずに、あえて、TYPE4 として、連番にしたか？です。

簡単にいえば、『新しいタイプを増やすつもりがない』ということです。

確かに、細かく分類することにより、再利用性は増すかもしれませんが、逆に、めったに使わない標準タイプが増えることにより、維持、管理工数が増大します。

ほとんどのケースで、タイプ1~3で収まるなら、あえてそれ以上増やす必要がないというのが、その他の分類を TYPE4 にした理由です。

タイプ別使用比率(例)



第 III 部

例題と演習課題

『何かを学ぶためには、自分で体験する以上にいい方法はない。』
Albert Einstein (アルベルト・アインシュタイン)

ここでは、例題の解説と演習課題を提示します。
構成は次のとおりです。

第8章 実習環境設定
Java, Tomcat の確認を行い、実習環境を作成します。

第9章 TypeS: 条件検索
TypeS: 条件検索を例に、SQL 検索、PL/SQL 検索および、リンク、マーカー作成を実習します。

第10章 Type1: 一覧検索一覧登録
Type1: 一覧検索一覧登録を例に、PL/SQL 登録、EXCEL 出力/EXCEL 取込および、カスタムボタン作成を実習します。

第11章 Type3: 検索エントリー
Type3: 検索エントリーを例に、PL/SQL 登録を実習します。

Web
Web
アプリケーション

第8章 実習環境設定

ここでは、Java, Tomcat の確認を行い、実習環境を作成します。

1. Java, Tomcat 確認

通常、H:ドライブにアプリケーションをインストールします。

H:¥java¥jdk14 フォルダと、H:¥java¥tomcat フォルダを確認してください。

また、これらへのパスは、初期化ファイル(g:¥bin¥init.bat) で設定されています。

初期化ファイルの設定内容を確認してください。

2. training システム確認

Web アプリケーションは、G:¥webapps フォルダ以下に配置します。

training フォルダを確認してください。

3. 各自システムホルダ作成

各自の[名前]を使用して、アプリケーションのフォルダを、webapps 以下に作成してください。

ここでは、例として、training フォルダを丸ごとコピーして[名前]というフォルダを作成します。

G:¥webapps

----- training

----- [名前]

----- filetemp

----- help

----- jsp

----- TYPE** を削除する。

----- log

----- src

----- SystemResource.properties を確認する。

----- GUIResource_ja.properties を書き換える。

----- jccall.bat を確認する。

----- WEB-INF

----- web.xml を確認する。

----- classes

----- resource 以下のフォルダを確認する。

----- lib

----- hayabusa3.jar を確認する。

4. リソース情報設定

- jspフォルダの TYPE** (TYPE で始まるフォルダ) は、削除します。
- SystemResource.properties で、各種設定を行います。
今回は、training がすでに動作しているため、データベース設定は不要です。
SYSTEM_ID = GE の個所を探し、各自のシステム ID に書き換えてください。
- GUIResource_ja.properties より、先ほど削除した TYPE** のメニューを削除します。代わりに、TEST01, TEST02, TEST03 のコメントをはずします。
修正後、src の下の jccall.bat をクリックして、リソースファイルのコンパイルを行います。
正常にコンパイルができれば、WEB-INF\classes¥resource 以下のファイルの更新日時がコンパイルした時刻になっているはずです。
- WEB-INF\web.xml ファイルで、認証やフィルター等、必要な設定を行います。
今回は、training がすでに動作しているため、変更不要です。

5. 動作チェック

[http://サーバー名:8823/\[名前\]/jsp/index.jsp](http://サーバー名:8823/[名前]/jsp/index.jsp) にアクセスして、初期画面が見えるか、確認してください。

第9章 TypeS:条件検索

ここでは、TypeS:条件検索を例に、SQL 検索、PL/SQL 検索および、リンク、マーカー作成を実習します。

検索画面としては、ユーザー定義画面を作成していただきます。
ユーザー定義テーブル(GE10)は、すでに作成済みですので、DB 定義書を参考に、課題をクリアしてください。

GE10

ユーザー定義		
○システム ID	SYSTEM_ID	X(10)
○ユーザーID	USERID	X(10)
パスワード	PASSWD	X(32)
ロケール(言語)	LANG	X(2)
名称	NAME	X(30)
ロール	ROLE	X(20)
ユニークキー	UNIQ	S(9)
状態フラグ	FGJ	X(1)
登録日時	DYSET	X(14)
更新日時	DYUPD	X(14)
登録者	USRSET	X(10)
更新者	USRUPD	X(10)
更新PRG	PRGUPD	X(10)

課題1 SQL 検索

TEST01 フォルダを使用します。
条件:システム IDとユーザーIDと名称のあいまい条件から、一覧表示で検索する。
並び替え順は、
システム ID+ユーザーID
システム ID+名称
システム ID+パスワード+ユーザーID
システム ID は、必須入力(must="true")としてください。



ヒント

training の TYPE1A などの、query.jsp が使用できます。検索条件のキーを columnTag を用いて記述してみてください。

課題2 リソースファイル作成

ユーザーID、パスワード、名称、ロールが、データベースのカラム名のままとなっています。リソースファイルに必要な情報を設定します。
カラムリソース設定時に、パスワードのレンダラー、エディターには、PASSWD を使用してみましょう。

```
DBCColumnResource.properties
LabelResource_ja.properties
```

修正後、jccall.bat でコンパイルが必要です。

課題3 リンク、マーカー作成

ユーザーIDにリンクを作成します。飛ばす画面は、TEST03 です。
 まだ、作成していないので、とりあえず、自分自身(TEST01)に飛ばすように
 しておいてください。

?

ヒント

training の TYPE1A などの、result.jsp の viewLink が参考になります。
`<mis:viewLink command="{@command}">`
`<mis:link column="USERID" gameId="TEST01" href="index.jsp"`
`target="CONTENTS"`
`keys="command,SYSTEM_ID,USERID"`
`value="NEW,{@SYSTEM_ID},{@USERID}" />`
`</mis:viewLink>`

検索項目をひとつ(checkFlag)増やして、そこにマーカーを作成します。
 名称が未設定(null)の場合に、警告を意味するマークをつけます。

```
select .....
                                          ,case when name is null then 'NG' else 'OK' end checkFlag
from GE10
.....
```

OK.gif と、NG.gif がデフォルトで jsp/image フォルダに存在しています。

?

ヒント

training の TYPE1A などの、result.jsp の viewMarker が参考になります。
`<mis:viewMarker command="{@command}">`
`<mis:columnMarker column="CHECKFLAG" >`
``
`</mis:columnMarker>`
`</mis:viewMarker>`

課題4 PL/SQL 検索

検索系の PL/SQL を作成して、カーソルを返してください。

?

ヒント

training の TYPESB が参考になります。

第10章 Type1:一覧検索一覧登録

ここでは、Type1:一覧検索一覧登録を例に、PL/SQL 登録、EXCEL 出力/EXCEL 取込および、カスタムボタン作成を実習します。

フォルダは、TEST02 として、先ほどの検索画面 (TEST01) をコピーし、作り変えます。

課題5 PL/SQL 登録

result.jsp 画面に、追加、変更、削除のボタンを用意します。
また、それぞれに対応した、copy.jsp , modify.jsp , delete.jsp ファイルを用意します。

登録用 PL/SQL を作成して、動作確認を行います。

課題6 EXCEL 出力/EXCEL 取込

追加、変更、削除ボタンの横に、EXCEL 関連のボタンを配置します。
result.jsp には、EXCEL 出力と、EXCEL 編集を、copy.jsp には、EXCEL 取り込みを配置します。
(modify.jsp と delete.jsp には、EXCEL 関連機能は設けません。)

?

ヒント

EXCEL 出力と編集がセットになったファイルは、common フォルダの、Excel_direct.jsp です。これを、インクルードして使用します。

```
<jsp:directive.include file="../../common/Excel_direct.jsp" /><br />
```

EXCEL 取り込みは、Excel_insert.jsp です。同様にインクルードします。

```
<jsp:directive.include file="../../common/Excel_insert.jsp" /><br />
```

?

ヒント

これらの EXCEL 関連機能は、Web サーバーとクライアント間で、ディスクのマウントを行う必要があります。ドライブ割り当てを行うのではなく、ネットワークドライブのままでも指定できます。

共有テンポラリフォルダの設定は、SystemResource.properties の FILE_URL で指定します。

例: FILE_URL=¥¥¥¥サーバー名/uap/webapps/[名前]/filetemp/

課題7 カスタムボタン作成

パスワードを自動設定するボタンを作成してください。
(パスワードが NULL の場合は、ユーザーID を設定する。)
処理する JSP は、passwd.jsp ファイルとしてください。
ボタンのラベルは、MessageResource_ja.properties に作成してください。

第 11 章 Type3: 検索エントリー

ここでは、Type3: 検索エントリーを例に、PL/SQL 登録を実習します。
TEST03 というフォルダを作成します。
コピー元は、training の TYPE3B が利用できます。

課題8 PL/SQL 登録

システム ID、ユーザー ID を入力して、あれば更新用に検索して、
情報を表示してください。
データがなければ、デフォルト値を表示して、新規登録してください。
なお、課題3で作成したリンクを、この画面 (TEST03) にはりなおしてください。

演習課題は、これで終了です。

お疲れ様でした。



索引

C

columnCheck 19, 28, 31
comment 16, 19, 23, 24, 26, 27
copy.jsp 5

D

delete.jsp 5
dummy.html 5

E

Excel_direct.jsp 22, 29
Excel_update.jsp 26

F

forward.jsp 5

H

HTMLTextField 30

I

index.jsp 5
insert.jsp 5

J

JDBCPrepared 28
JSP 5

M

modify.jsp 5
must 16, 18, 31

Q

query 15, 16, 19, 28, 33
query.jsp 5
query_info.jsp 15, 16
QUERY フレーム 2

R

result.jsp 5

result_info.jsp 18, 19
RESULT フレーム 2

S

selectedType 29

T

Type 1 3, 21
Type 2 3, 29
Type 3 3, 31
Type 4 3, 34
Type S 3, 15

V

viewFormType 18, 20, 22, 24, 26, 27, 29, 30

い

一覧検索一覧登録 3
一覧検索明細登録 3

け

検索エントリー 3

こ

コンテンツフレーム 2

し

条件検索 3

そ

その他 3

ひ

標準画面パターン 3, 4

め

メニューフレーム 2

