

Oracle® Application Server 10g

Multimedia Tag Library for JSP ユーザーズ・ガイド
およびリファレンス

10g (9.0.4)

部品番号 : B12335-01

2004 年 3 月

Oracle Application Server 10g Multimedia Tag Library for JSP ユーザーズ・ガイドおよびリファレンス, 10g (9.0.4)

部品番号 : B12335-01

原本名 : Oracle Application Server 10g Multimedia Tag Library for JSP User's Guide and Reference, 10g (9.0.4)

原本部品番号 : B10445-01

原本著者 : Sue Pelski

原本協力者 : Fengting Chen, Dong Lin, Susan Mavris, Susan Shepard, Richard Wang, Manjari Yalavarthy

Copyright © 2003 Oracle Corporation. All rights reserved.

制限付権利の説明

このプログラム（ソフトウェアおよびドキュメントを含む）には、オラクル社およびその関連会社に所有権のある情報が含まれています。このプログラムの使用または開示は、オラクル社およびその関連会社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と工業所有権に関する法律により保護されています。

独立して作成された他のソフトウェアとの互換性を得るために必要な場合、もしくは法律によって規定される場合を除き、このプログラムのリバース・エンジニアリング、逆アセンブル、逆コンパイル等は禁止されています。

このドキュメントの情報は、予告なしに変更される場合があります。オラクル社およびその関連会社は、このドキュメントに誤りが無いことの保証は致し兼ねます。これらのプログラムのライセンス契約で許諾されている場合を除き、プログラムを形式、手段（電子的または機械的）、目的に関係なく、複製または転用することはできません。

このプログラムが米国政府機関、もしくは米国政府機関に代わってこのプログラムをライセンスまたは使用する者に提供される場合は、次の注意が適用されます。

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation, and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

このプログラムは、核、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への用途を目的としておりません。このプログラムをかかるとの目的で使用する場合、上述のアプリケーションを安全に使用するために、適切な安全装置、バックアップ、冗長性 (redundancy)、その他の対策を講じることは使用者の責任となります。万一かかるプログラムの使用に起因して損害が発生いたしましても、オラクル社およびその関連会社は一切責任を負いかねます。

Oracle は Oracle Corporation およびその関連会社の登録商標です。その他の名称は、Oracle Corporation または各社が所有する商標または登録商標です。

目次

はじめに	v
対象読者	vi
このマニュアルの構成	vi
関連ドキュメント	vii
表記規則	viii
1 Multimedia Tag Library の概要	
Oracle <i>interMedia</i> の概要	1-2
メディアの取得の概念	1-3
メディアのアップロードの概念	1-3
Java の概念	1-4
J2EE	1-4
Java サーブレット	1-4
JavaServer Pages	1-4
JSP タグ・ライブラリ	1-5
Oracle <i>interMedia</i> Java Classes for Servlets and JSP または Multimedia Tag Library の選択	1-5
2 マルチメディア JSP タグを使用したサンプル・プログラム	
メディアの取得例	2-2
メディアのアップロード例	2-6

3 メディアの取得に関する参照情報

前提条件	3-3
参照情報	3-3
メディア取得タグの一般的な書式	3-4
custom-retrieval-attributes	3-5
database-connection-attributes	3-6
media-access-attributes	3-8
media-cache-control-attributes	3-10
table-and-column-attributes	3-12
media-render-attributes	3-14
embedAudio	3-15
embedImage	3-17
embedVideo	3-20
mediaUrl	3-25
メディア取得 URL の書式	3-27

4 メディアのアップロードに関する参照情報

前提条件	4-2
参照情報	4-2
storeMedia	4-3
uploadFile	4-6
uploadFormData	4-8

A インストールおよび構成に関する情報

前提条件	A-2
Multimedia Tag Library のインストール	A-2
Multimedia Tag Library の構成	A-3
TLD ファイルの仮想パスの指定	A-3
メディア配信コンポーネントの指定	A-3
メディア・データへのアクセスの許可	A-4
キャッシュ制御属性の設定	A-5
サンプル構成ファイル	A-5

B エラー・メッセージ

索引

例

2-1	PhotoAlbum.jsp の内容	2-2
2-2	PhotoAlbumInsertPhoto.jsp の内容	2-6

はじめに

このマニュアルでは、Oracle Application Server 10g Multimedia Tag Library for JSP の使用方法について説明します。

対象読者

このマニュアルは、初級から中級レベルの JSP ベース・マルチメディア Web アプリケーションを開発する、熟練したアプリケーション開発者を対象としています。また、マルチメディア・アプリケーション・プロトタイプを速やかに作成する必要があるアプリケーション開発者や、Java アプリケーションの記述に関していくらか経験のある JavaServer Pages の作成者も対象としています。

このマニュアルの構成

このマニュアルは次の章および付録で構成されています。

第 1 章「Multimedia Tag Library の概要」

メディアの取得およびアップロードの概要について説明します。

第 2 章「マルチメディア JSP タグを使用したサンプル・プログラム」

(Oracle *interMedia* のフォト・アルバム・サンプル・アプリケーションに基づいた) サンプル・プログラムについて説明します。

第 3 章「メディアの取得に関する参照情報」

メディア取得タグに関する参照情報について説明します。

第 4 章「メディアのアップロードに関する参照情報」

メディア・アップロード・タグに関する参照情報について説明します。

付録 A「インストールおよび構成に関する情報」

Oracle Application Server 10g Multimedia Tag Library for JSP のインストールおよび構成方法について説明します。

付録 B「エラー・メッセージ」

Oracle Application Server 10g Multimedia Tag Library for JSP により発生する可能性があるエラーについて説明します。

関連ドキュメント

このマニュアルは単独で使用するドキュメントではありません。『Oracle *interMedia* リファレンス』、『Oracle *interMedia* ユーザーズ・ガイド』および『Oracle *interMedia* Java Classes リファレンス』を補足するドキュメントです。Java インタフェースを使用して Oracle *interMedia* (「*interMedia*」) オブジェクトに対して正常に操作を実行するには、次のマニュアルも必要です。

- 『Oracle9i JDBC 開発者ガイドおよびリファレンス』
- 『Oracle Application Server Containers for J2EE ユーザーズ・ガイド』
- 『Oracle Application Server Containers for J2EE Support for JavaServer Pages 開発者ガイド』
- 『Oracle Application Server Containers for J2EE JSP タグ・ライブラリおよびユーティリティ・リファレンス』
- 『PL/SQL ユーザーズ・ガイドおよびリファレンス』

開発環境における *interMedia* の使用方法の詳細は、Oracle ドキュメント・セットに含まれる次のドキュメントを参照してください。

- 『Oracle9i アプリケーション開発者ガイド - 基礎編』
- 『Oracle9i データベース概要』

このマニュアルのリリース後に追加された情報は、次の OTN-J Web サイトの Oracle Application Server のオンライン・リリース・ノートのセクションを参照してください。

<http://otn.oracle.co.jp/document/index.html>

Java の詳細は、次の URL にある Sun 社提供の API ドキュメントを参照してください。

<http://java.sun.com/docs>

このマニュアルのハードコピーはありません。

リリース・ノート、インストール関連ドキュメント、ホワイト・ペーパーまたはその他の関連ドキュメントは、OTN-J (Oracle Technology Network Japan) から、無償でダウンロードできます。OTN-J を使用するには、オンラインでの登録が必要です。登録は、次の Web サイトから無償で行えます。

<http://otn.oracle.co.jp/membership/>

OTN-J のユーザー名とパスワードを取得済であれば、次の OTN-J Web サイトのドキュメントのセクションに直接接続できます。

<http://otn.oracle.co.jp/document/>

表記規則

例の中では、特に記載されていないかぎり、各行の最後で暗黙的な改行が行われています。入力行の最後では [Return] キーを押す必要があります。

このマニュアルでは、次の表記規則も使用します。

規則	意味
.	例の中で使用されている垂直の省略記号は、その例に直接関係のない情報が省略されていることを示します。
太字のテキスト	太字のテキストは、本文で定義されている用語を示します。
固定幅テキスト	固定幅テキストは例に使用します。本文で、ファイル名やディレクトリ名、例の抜粋にも使用します。
<>	山カッコで囲まれている名前は、ユーザーが指定する名前を示します。エラー・メッセージ・テキストの中で山カッコで囲まれている名前は、システムにより指定される名前を示します。
{ }	構文または例の中で中カッコで囲まれている 2 つ以上の項目は、どちらかが必須であることを示します。
[]	大カッコで囲まれている句は、その中から 1 つを選択するか、または何も選択しないオプションの句を示します。Java コードの中で使用されている大カッコは、メソッドにより一連の値が戻されることを示します。

Multimedia Tag Library の概要

Oracle *interMedia* (「*interMedia*」) で提供されるカスタム JavaServer Pages (JSP) タグ・ライブラリによって、ユーザーは JavaServer Pages でマルチメディア HTML タグを容易に生成し、マルチメディア・データを *interMedia* オブジェクトにアップロードできます。

この章では、*interMedia* および Java の概念を簡潔に簡単に説明し、Oracle Application Server 10g Multimedia Tag Library for JSP (「Multimedia Tag Library」) が JSP ベースのマルチメディア Web アプリケーションにどのように組み込まれているかを示します。

Oracle *interMedia* の概要

Oracle *interMedia* を使用すると、Oracle Database ソフトウェアにおいて、イメージ、オーディオ、ビデオおよびその他のメディア・データを格納、取得、管理および操作し、これらのデータをその他の企業情報と統合できます。

具体的には、Oracle *interMedia* では、メディアの格納、取得、管理、および Oracle Database で管理され、バイナリ・ラージ・オブジェクト、ファイル・ベースのラージ・オブジェクト、メディア・データが含まれる URL および特殊サーバーのいずれかに格納されているメディア・データの操作がサポートされています。Oracle *interMedia* のメディアの格納および取得サービスは、Web コンテンツの管理機能を拡張するものです。

アプリケーションでは、リレーショナル・インタフェースおよびオブジェクト・インタフェースのどちらを介しても Oracle *interMedia* にアクセスできます。Java、C++ または従来の 3GL で記述されたデータベース・アプリケーションは、今日のクラス・ライブラリ・インタフェースまたは PL/SQL および Oracle Call Interface を介して *interMedia* と対話できます。

Oracle *interMedia* では、メディア・データを記述するため、Java クラスまたは C++ クラスに類似したオブジェクト・タイプが使用されます。これらの *interMedia* オブジェクト・タイプは、ORDAudio、ORDDoc、ORDImage および ORDVideo と呼ばれます。*interMedia* オブジェクトには、共通のメディア・データ格納モデルがあります。

Oracle *interMedia* では、ユーザーが *interMedia* オブジェクトを使用した Java アプリケーションを記述できるように、Java Classes も提供されます。Oracle *interMedia* では、データベース表へのメディア情報の格納、この表からのメディア情報の取得および操作が可能です。Oracle *interMedia* Java Classes では、ユーザー独自の Java アプリケーションを記述して、データベースに格納されたメディア・データを使用、操作および変更できます。Oracle *interMedia* Java Classes により、アプリケーションでは結果セットからオブジェクトを取得し、オブジェクトのコンテンツを操作できます。

Oracle *interMedia* Java Classes for Servlets and JavaServer Pages (JSP) により、データベースに対するメディア・データの取得およびアップロードが容易になります。

Oracle *interMedia* および *interMedia* オブジェクトの詳細は、『Oracle *interMedia* ユーザーズ・ガイド』および『Oracle *interMedia* リファレンス』を参照してください。Java アプリケーションでの *interMedia* オブジェクトの使用の詳細は、『Oracle *interMedia* Java Classes リファレンス』を参照してください。

メディアの取得の概念

Oracle *interMedia* Java Classes for Servlets and JSP では、`OrdHttpServletResponse` クラスにより、Oracle データベースからメディア・データが取得され、このデータが Java サーブレットまたは JSP ページからブラウザまたはその他の HTTP クライアントに配信されます。

Multimedia Tag Library により提供されるメディア取得タグを使用することで、JSP 開発者は、完全な HTML マルチメディア・タグを生成するか、マルチメディア取得 URL を作成して HTML マルチメディア・タグのカスタマイズ用途に含めることができます。メディア取得タグには、`embedAudio`、`embedImage`、`embedVideo` および `mediaURL` があります。

メディアのアップロードの概念

Oracle *interMedia* Java Classes for Servlets and JSP では、アップロードするメディア・ファイルの処理を容易にするため、`OrdHttpUploadFile` クラスが使用されます。このクラスでは単純な Application Programming Interface (API) が用意されており、アプリケーションにおいてメディア・データをデータベースにロードする際にコールされます。

HTML フォームを使用してファイルをアップロードする場合、マルチパート / フォームデータ形式を使用した POST リクエスト内のフォーム・データおよびアップロード・ファイルがエンコードされます。`OrdHttpUploadFormData` クラスを使用すると、POST データが解析され、通常のフォーム・フィールドのコンテンツおよびアップロード・ファイルのコンテンツが Java サーブレットまたは JSP ページからすぐにアクセスできるようになるため、こうしたリクエストの処理が容易になります。

Multimedia Tag Library で提供されるメディア・アップロード・タグにより、メディア・データをデータベースにアップロードするマルチメディア・アプリケーションの開発が容易になります。メディア・アップロード・タグには、`storeMedia`、`uploadFile` および `uploadFormData` があります。

Java の概念

Java は、Sun 社が開発したオブジェクト指向のプログラミング言語です。Java プログラムは、Java 仮想マシン (JVM) が組み込まれたコンピュータ・プラットフォーム上で実行できます。JVM は、コードをコンパイルする特殊なソフトウェア環境です。JVM では、ユーザー入力またはシステム入力の収集、情報の表示およびメモリーの割当てなどのタスクを実行するために Java プログラムで使用されるソフトウェアが提供されます。Java は、小規模の Web アプリケーションまたは大規模のネットワーク・アプリケーションの開発に使用できます。

J2EE

Java2 Enterprise Edition (J2EE) は、Sun 社が開発した統合開発環境であり、旧 Java2 Standard Edition (J2SE) プラットフォームの機能を拡張したものです。J2EE は、数ある機能の中で、特に Java サーブレット、JSP ページおよび XML をサポートしています。

Java サーブレット

Java サーブレットは、Web 開発者による Web サーバー機能の拡張および強化を可能にする Java プログラムです。Java サーブレットは、Java クラスおよびストリームを含む Java 言語の最新機能を使用したモジュール式プログラミングに役立ちます。Java サーブレットは、HTTP に固有のコールのコレクションの他、最新の Java API にもアクセスできます。Java サーブレットは、JVM が組み込まれたプラットフォームおよびサーブレットをサポートする Web サーバー上でサポートされています。

Java サーブレットの多くは、クライアントからリクエストを受け取り、レスポンスで動的 HTML テキストを生成し、Web ブラウザで表示されるようにこのテキストをクライアントに戻します。Java サーブレットは、XML テキストを生成してデータをカプセル化し、クライアントまたはアプリケーションの他の部分にデータを送信することもできます。Java サーブレットは、対話型アプリケーションの作成に多く使用されます。

JavaServer Pages

JavaServer Pages (JSP) テクノロジは、Java サーブレット・テクノロジの拡張機能です。これによって、ユーザーは Java コードの各部を HTML ページまたは別のタイプのドキュメント (XML ファイルなど) に含めることができます。

一般的な JSP ページでは、フォームは機能から独立しています。情報のフォームまたはレイアウトは JSP ページの作成者が決定しますが、データベース・アクセスなどの他の機能は、JavaBeans へのコールまたは他のメカニズムにより処理されます。

JSP タグ・ライブラリ

JSP タグ・ライブラリはすべて、JSP テクノロジーを拡張したものです。一般に、JSP タグ・ライブラリには、JSP ページで使用できるタグ・セット内でカプセル化された特定のモジュール機能が含まれています。

カスタム JSP タグ・ライブラリでは、具体的に Java コードを入力しなくても、Java 機能を組み込むことができます。各 JSP タグ・ライブラリでは特殊な二次用語が定義されているため、JSP ページでこれらの Java 機能を一層無理なく使用できます。たとえば、Oracle Application Server Containers for J2EE (OC4J) では、JSP Markup Language (JML) Tag Library が提供されます。JML を他の機能と併用することで、アプリケーション開発者は、Java コード・スクリプトの構文を直接使用しなくても、条件ロジックおよびループ構成を含む JSP ページを記述できます。

Oracle *interMedia* Java Classes for Servlets and JSP または Multimedia Tag Library の選択

Oracle *interMedia* Java Classes for Servlets and JSP または Multimedia Tag Library のどちらを使用するかは、開発するアプリケーションのタイプと、Java プログラミングをどの程度経験しているかによって判断します。

Java サーブレットおよび JSP ページには柔軟性があり、アプリケーション出力をカスタマイズできます。このカスタマイズを行うには、Java プログラミング言語やサーブレットおよび JSP テクノロジーについて、ある程度のスキルが必要です。したがって、サーブレットおよび JSP ページは、適度に熟練した Java プログラマーが複雑なアプリケーションを開発する場合に最も適しています。

Multimedia Tag Library に用意されているタグはすばやく簡単に使用できるため、ほとんどカスタマイズしなくても単純な Java アプリケーションを記述できます。JSP タグは、限られた Java コードを使用して一般的なアプリケーションをすばやく開発する必要がある、経験の浅い Java プログラマーに適しています。

マルチメディア JSP タグを使用した サンプル・プログラム

この章では、Oracle Application Server 10g Multimedia Tag Library for JSP (「Multimedia Tag Library」) を使用したユーザー定義 JSP ページのサンプル・コードを省略なしで示しています。この章で説明する内容は、SQL、PL/SQL、Java Database Connectivity (JDBC)、OC4J および JSP について予備知識があることを前提としています。関連ドキュメントのリストは、このマニュアルの「はじめに」を参照してください。

このコードは、OTN からサンプル・アプリケーションとしてダウンロード可能なファイル内のコードと一致しない場合があります。システム上でサンプル・コードを実行する場合は、OTN の Multimedia Tag Library ソフトウェアで提供されているファイルを使用するようにし、この章に示したコードはコンパイルおよび実行しないでください。

メディアの取得およびメディアのアップロードの例では、データベースへのアクセスに SQL タグ・ライブラリも使用しています。これらのタグの接頭辞は sql です。SQL タグ・ライブラリの詳細は、『Oracle Application Server Containers for J2EE JSP タグ・ライブラリおよびユーティリティ・リファレンス』のデータ・アクセス用 SQL タグに関する資料を参照してください。

メディアの取得およびメディアのアップロードの例では、どちらも photos というデータベース表を使用しています。photos 表の表定義は次のとおりです。

```
photos( id          NUMBER UNIQUE NOT NULL,
        description VARCHAR2(40) NOT NULL,
        location    VARCHAR2(40),
        image       ORDSYS.ORDIMAGE,
        thumb       ORDSYS.ORDIMAGE);
```

注意： この章では、Java および HTML コードのサンプルを示しています。サンプル・コードでは、数値が大カッコで囲まれています。これは、サンプルの直後の番号付きリストで、そのコードを詳細に説明していることを示します。

メディアの取得例

PhotoAlbum.jsp ファイルは、サンプル JSP アプリケーションのコンポーネントの 1 つであり、データベースからメディア・データを取得し、このデータをブラウザに配信するために、Multimedia Tag Library のタグが使用されています。ブラウザでは、メディアが単純なフォト・アルバム・アプリケーションに表示されます。例 2-1 では、mediaUrl および embedImage のタグを示しています。

PhotoAlbum.jsp ファイルでは、photos というデータベース表の内容 (description、location および thumb 列の内容を含む) を表示する HTML コードが生成されます。photos 表の thumb 列の内容は、フルサイズ・イメージにリンクしたサムネイル・イメージで表示されます。フルサイズ・イメージは、photos 表の image 列に格納されています。ユーザーは、ブラウザでサムネイル・イメージをクリックすると、フルサイズ・イメージを表示できます。

例 2-1 PhotoAlbum.jsp の内容

```
[1] <%@ page language="java" %>
[2] <%@ taglib prefix="ord" uri="/Web-inf/intermedia-taglib.tld" %>
<%@ taglib prefix="sql" uri="/web-inf/sqltaglib.tld" %>

<%
[3] public static final String escapeHtmlString(String input)
    {
        StringBuffer sb = new StringBuffer();

        for (int i = 0; i < input.length(); i++)
        {
            char ch = input.charAt(i);
            switch (ch)
            {
                case '<':
                    sb.append("&lt;");
                    break;
                case '>':
                    sb.append("&gt;");
                    break;
                case '&':
                    sb.append("&amp;");
                    break;
                case '"':
                    sb.append("&quot;");
                    break;
                case ' ':
                    sb.append("&nbsp;");
                    break;
            }
        }
    }
}
```

```
        default:
            sb.append(ch);
        }
    }

    return sb.toString();
}

%>

<%-- HTML header --%>
<HTML LANG="EN">
<HEAD>
<TITLE>interMedia JavaServer Pages Photo Album Demo</TITLE>
</HEAD>

<BODY>

<%-- Page heading --%>
[4] <TABLE BORDER="0" WIDTH="100%">
    <TR>
        <TD COLSPAN="2" BGCOLOR="#F7F7E7" ALIGN="CENTER"><FONT SIZE="+2">
            <I>inter</I>Media JavaServer Pages Photo Album Demo</FONT>
        </TD>
    </TR>
</TABLE>

<P>
<TABLE BORDER="1" CELLPADDING="3" CELLSPACING="0" WIDTH="100%"
    SUMMARY="Table of thumb nail images">
    <TR BGCOLOR="#336699">
        <TH id="description"><FONT COLOR="#FFFFFF">Description</FONT></TH>
        <TH id="location"><FONT COLOR="#FFFFFF">Location</FONT></TH>
        <TH id="image"><FONT COLOR="#FFFFFF">Image</FONT></TH>
    </TR>

    <% int rowCount = 0; %>
[5] <sql:dbOpen connId = "myConn" dataSource="jdbc/OracleDS" />
<sql:dbQuery connId = "myConn" queryId="myQuery" output="jdbc">
    SELECT id, description, location from photos order by description
</sql:dbQuery>
<sql:dbNextRow queryId="myQuery">
    <%
[6]     String id = myQuery.getString(1);
        String description = myQuery.getString(2);
        String location = myQuery.getString(3);
    %>
[7]     <TR>
```

```
<TD HEADERS="description">
  <%= escapeHtmlString(description) %>
</TD>
<%
  if ( location != null )
    out.print( "<TD HEADERS=\"location\">" +
      escapeHtmlString(location) + "</TD>" );
  else
    out.print( "<TD HEADERS=\"location\">&nbsp;</TD>" );
%>
<TD HEADERS="image">
[8]   <ord:mediaUrl dataSourceName="jdbc/OracleDS"
      table = "photos"
      column = "image"
      key = "<%=id%>"
      keyColumn = "id"
      id = "urlId">
[9]   <A HREF="<%= urlId.getUrl() %>">
[10]  <ord:embedImage dataSourceName="jdbc/OracleDS"
      table = "photos"
      column = "thumb"
      key = "<%= id %>"
      keyColumn = "id"
      alt = "<%=escapeHtmlString(description)%>"
      border="1" />

      </A>
    </ord:mediaUrl>
  </TD></TR>

[11] <% rowCount ++; %>
</sql:dbNextRow>
<sql:dbCloseQuery queryId="myQuery"/>
<sql:dbClose connId="myConn"/>

<TR>
  <TD SCOPE="col" COLSPAN="3" ALIGN="CENTER"><FONT COLOR="#336699"><B><I>
<%
  if (rowCount == 0)
  {
    out.println(" The photo album is empty");
  }
  else
  {
    out.println(" Select the thumb-nail to view the full-size image");
  }
%>
```

```
        </I></B></FONT></TD>
    </TR>
<%-- Finish the table --%>
</TABLE>
</P>

<P>
<TABLE WIDTH="100%">
  <TR BGCOLOR="#F7F7E7">
    <TD COLSPAN="3" ALIGN="CENTER">
      <A HREF="PhotoAlbumUploadForm.jsp">Upload new photo</A>
    </TD>
  </TR>
</TABLE>
</P>

</BODY>
</HTML>
```

PhotoAlbum.jsp ファイル内の Java、SQL および HTML 文により、次の操作が実行されます。

1. JSP ページで使用するスクリプト言語として、Java が宣言されます。(このコード行は JSP ディレクティブです。)
2. 接頭辞および URI 属性が指定されます。URI 属性の値は、タグ・ライブラリのタグ・ライブラリ・ディスクリプタ (TLD) ファイルの場所を示します。接頭辞属性 (ord) は、XML 名前空間識別子を指定します。この識別子は、JSP ページ内でライブラリのタグの各出現箇所の前に挿入する必要があります。(この 2 つのコード行は Multimedia Tag Library ディレクティブです。)
3. HTML 内のよく使用される特殊文字を解析するためのエスケープ・シーケンスを提供するメソッドが宣言されます。(これはメソッド宣言文と呼ばれます。)
4. データベース内の photos 表のエントリを表示するために、HTML 表が使用されます。(これは HTML プログラムです。)
5. データベース接続がオープンされ、photos 表で問合せが実行された後、取得済の結果セットにおいてループが実行されます。
6. 結果セットからデータが取得されます。
7. 表内のエントリの表示が開始されます。
8. データベース内の photos 表の image 列を示す urlId というスクリプト変数が作成されます。(このコード行では、マルチメディア JSP タグ mediaUrl を示しています。)
9. スクリプト変数に格納されている URL を示すリンクが指定されます。

10. photos 表の thumb 列を表示する HTML の タグが生成されます (HTML 出力の詳細は、[embedImage](#) を参照してください)。HTML の <A HREF> タグでは、リンク・アンカーとして JSP タグ `embedImage` が使用されます。(このコード行では、マルチメディア JSP タグ `embedImage` を示しています。)
11. ループが終了し、問合せおよびデータベース接続がクローズします。

メディアのアップロード例

PhotoAlbumInsertPhoto.jsp ファイルは、サンプル JSP アプリケーションのコンポーネントの 1 つであり、メディア・ファイルをデータベースにアップロードするために、Multimedia Tag Library のタグが使用されています。例 2-2 では、`uploadFormData`、`uploadFile` および `storeMedia` のタグを示しています。

例 2-2 PhotoAlbumInsertPhoto.jsp の内容

```
[1] <%@ page language="java" %>
<%@ taglib prefix="ord" uri="/Web-inf/intermedia-taglib.tld" %>
<%@ taglib prefix="sql" uri="/web-inf/sqltaglib.tld" %>

[2] <ord:uploadFormData formDataId = "fd">

[3] <ord:uploadFile
      parameter = "photo"
      fullFileName = "ffName"
      shortFileName = "sfName"
      length = "fLength" >

    <%
[4]   if (ffName == null || ffName.length() == 0)
      {
    %>
        <jsp:forward
page="PhotoAlbumUploadForm.jsp?error=Please+supply+a+file+name."/>
    <%
        return;
      }

      if (fLength.intValue() == 0)
      {
    %>
        <jsp:forward
page="PhotoAlbumUploadForm.jsp?error=Please+supply+a+valid+image+file."/>

    <%
```

```

        return;
    }

    String description = fd.getParameter("description");
    String location = fd.getParameter("location");
[5] if ( description == null || description.length() == 0 )
    {
        description = "Image from file: " + sfName + ".";
        if(description.length() > 40)
        {
            description = description.substring(0, 40);
        }
    }

    java.util.Vector otherValuesVector = new java.util.Vector();
    otherValuesVector.add(description);
    otherValuesVector.add(location);
%>

<%String id = "original"; %>
[6] <sql:dbOpen connId = "myConn" dataSource="jdbc/OracleDS"
        commitOnClose="true"/>

<sql:dbQuery connId = "myConn" queryId="myQuery" output="jdbc">
    SELECT photos_sequence.nextval from dual
</sql:dbQuery>

<sql:dbNextRow queryId="myQuery">
    <% id = myQuery.getString(1); %>
</sql:dbNextRow>

<sql:dbCloseQuery queryId="myQuery"/>

[7] <ord:storeMedia
    conn = "<%= (oracle.jdbc.driver.OracleConnection)myConn.getConnection() %%"
    table = "photos"
    key = "<%=id%%"
    keyColumn = "id"
    mediaColumns = "image"
    mediaParameters = "photo"
    otherColumns = "description, location"
    otherValues = "<%=otherValuesVector%%"
/>

[8] <sql:dbSetParam name = "myid" value = "<%=id%%" />
<sql:dbExecute connId = "myConn" bindParams="myid">
    {call generateThumbNail(?)}
```

```

        </sql:dbExecute>

        <sql:dbClose connId = "myConn" />

    </ord:uploadFile>
</ord:uploadFormData>

<%-- HTML header --%>
<HTML LANG="EN">
<HEAD>
<TITLE>interMedia JavaServer Pages Photo Album Demo</TITLE>
</HEAD>

[9] <META HTTP-EQUIV="REFRESH" CONTENT="2;URL=PhotoAlbum.jsp">

<BODY>

    <%-- Page heading --%>
    <TABLE BORDER="0" WIDTH="100%">
        <TR>
            <TD COLSPAN="2" BGCOLOR="#F7F7E7" ALIGN="CENTER"><FONT SIZE="+2">
                <I>inter</I>Media JavaServer Pages Photo Album Demo</FONT>
            </TD>
        </TR>
    </TABLE>

    <%-- Display header and instructions --%>
    <P>
    <FONT SIZE=3 COLOR="#336699">
    <B>Photo successfully uploaded into photo album</B>
    </FONT>
    <HR SIZE=1>
    </P>
    <P>
    Please click the link below or wait for the browser to refresh the page.
    </P>

    <%-- Output link to return to the main page --%>
    <P>
    <TABLE WIDTH="100%">
        <TR BGCOLOR="#F7F7E7">
            <TD COLSPAN="3" ALIGN="CENTER">
                <A HREF="PhotoAlbum.jsp">Return to photo album</A>
            </TD>
        </TR>
    </TABLE>

```



```
</TABLE>
</P>

<%-- Finish the page --%>
</BODY>
</HTML>
```

PhotoAlbumInsertPhoto.jsp ファイル内の Java、SQL および HTML 文により、次の操作が実行されます。

1. JSP ページで使用するスクリプト言語として、Java が宣言されます (JSP ディレクティブ)。TLD ファイルの場所と、必要な ord および sql 接頭辞属性が指定されます (タグ・ライブラリ・ディレクティブ)。
2. fd というスクリプト変数が作成されます。この変数は、oracle.ord.im.OrdHttpUploadFormData オブジェクトのインスタンスです。(このコード行では、マルチメディア JSP タグ uploadFormData を示しています。)
3. スクリプト変数 ffName、sfName および fLength が作成されます。これらの変数には、アップロードするメディアの完全なファイル名、短縮ファイル名およびファイルの長さがそれぞれ含まれます。(このコード行では、マルチメディア JSP タグ uploadFile を示しています。)
4. エラー・チェックが実行されます。
5. 説明が示されていない場合、デフォルトの説明が生成されます。
6. データベース接続がオープンし、データベース内の photos 表における次の一意の ID が取得されます。
7. メディア・データが photos 表の image 列に、また説明および場所の情報が photos 表の description および location 列にアップロードされます。(このコード行では、マルチメディア JSP タグ storeMedia を示しています。)
8. アップロードされた image 列から photos 表の thumb 列を移入するための PL/SQL プロシージャがコールされます。
9. 正常に実行されたことを示すメッセージが表示され、JSP ページがメイン・ページ (PhotoAlbum.jsp) に戻されます。

メディアの取得に関する参照情報

Oracle Application Server 10g Multimedia Tag Library for JSP (「Multimedia Tag Library」) に用意されたメディア取得タグを使用すると、完全な HTML マルチメディア・タグを生成したり、マルチメディア取得 URL を作成して HTML マルチメディア・タグのカスタマイズ用途に含める操作が簡単になります。メディア取得タグは次のとおりです。

- embedAudio
- embedImage
- embedVideo
- mediaUrl

マルチメディア・アプリケーションにおけるメディアの取得には、次のタスクが含まれます。

- メディア・データを取得またはレンダリングするための HTML タグの生成。生成された HTML タグには、メディア・データを取得する URL が含まれます。

次の例は、PhotoAlbum.jsp ファイルから抜粋したものです。この例は、マルチメディア JSP タグ embedImage を示しています。

```
<ord:embedImage dataSourceName="jdbc/OracleDS"
    table = "photos"
    column = "thumb"
    key = "<%=id%>"
    keyColumn = "id"
    alt = "<%=escapeHtmlString(description)%>"
    border="1" />
```

この例では、次の HTML タグが生成されます。このタグには、データベース内のイメージを取得する URL が含まれます。

```

```

データベースからメディア・データを取得し、このデータをブラウザに配信するために **Multimedia Tag Library** のタグが使用されるサンプル JSP アプリケーションの詳細は、[第 2 章](#) を参照してください。ブラウザでは、単純なフォト・アルバム・アプリケーションにメディアが表示されます。

- データベースからメディア・データを取得し、このデータをブラウザに配信するメディア配信コンポーネントの使用。前の例では、`OrdGetMediaServlet` が、生成された HTML `` タグの URL におけるメディア配信コンポーネントです。

より一般的な用途では、**Multimedia Tag Library** により、次のメディア・タイプについて、ある程度のカスタマイズ機能を持つ HTML タグを生成できます。

- HTML `` タグを使用してレンダリングされるイメージ。
- HTML `<EMBED>` および `<OBJECT>` タグを使用してレンダリングされるオーディオおよびビデオ・データ。この場合、メディア・データはデータベースからブラウザのプラグインまたはその他のメディア・プレーヤに直接配信されます。

注意： RealNetworks サーバーによりアクセスおよび配信される、Oracle データベースに格納されたオーディオおよびビデオのメディア・データの場合、より高いパフォーマンスを得るため、**Multimedia Tag Library** ではなく、RealNetworks ストリーミング・サーバー用の Oracle *interMedia* プラグインを使用することをお勧めします。

さらに、**Multimedia Tag Library** の柔軟性により、アプリケーションの要件に従ってメディア・データをレンダリングする HTML タグを構成しながら、タグ・ライブラリを使用してメディア取得 URL を作成し、メディアをブラウザに配信できます。

前提条件

なし。

参照情報

この項では、*interMedia* オブジェクト上で動作するメディア取得タグに関する参照情報を提供します。

メディア取得タグの一般的な書式

書式

```
<ord:tagName [ custom-retrieval-attributes ]  
             [ database-connection-attributes ]  
             [ media-access-attributes ]  
             [ media-cache-control-attributes ]  
             [ table-and-column-attributes ]  
             [ media-render-attributes ]>  
</ord:tagName>
```

説明

 などの完全な HTML メディア・タグを生成したり、メディア取得 URL を作成して HTML メディア・タグのカスタマイズ用途に含めることができます。メディア取得タグでは、接頭辞 `ord` の下に、`mediaUrl`、`embedImage`、`embedAudio` および `embedVideo` のタグが含まれます。

メディア取得タグには、共通属性のセットと、メディアをレンダリングするタグ固有の属性があります。この項では、共通属性 `custom-retrieval-attributes`、`database-connection-attributes`、`media-access-attributes`、`media-cache-control-attributes` および `table-and-column-attributes` と、これらのパラメータについて詳しく説明します。`media-render-attributes` はタグ固有であるため、各メディア取得タグの項で説明しています。

使用方法

メディア取得タグを使用する場合、次の情報を指定する必要があります。

- データベースからメディア・データを取得する際に使用するデータベース接続情報
- データベースからメディア・データを取得する際の場所を示した表、列およびキー情報

したがって、`database-connection-attributes` および `table-and-column-attributes` または `media-access-attributes` を指定する必要があります。

custom-retrieval-attributes

書式

```
custom-retrieval-attributes =  
    retrievalPath = "string | <%= jspExpression %>"
```

説明

メディア・データの配信について高度な制御を必要とするアプリケーションについて、アプリケーション固有のメディア取得コンポーネントの名前を指定するメカニズムが提供されます。この取得コンポーネントは、生成される URL に含まれます。

パラメータ

retrievalPath

カスタマイズしたメディア取得コンポーネントへのパス。メディア取得コンポーネントに対する入力パラメータの詳細は、この章の「[メディア取得 URL の書式](#)」を参照してください。

使用方法

なし。

例

この例では、`retrievalPath` 属性を使用して、メディア配信 JSP ページとして動作するカスタマイズ・バージョンの `OrdGetMediaJsp.jsp` ファイルを取り入れる方法を示しています。

```
<ord:embedVideo { database-connection-attributes }  
    [ media-cache-control-attributes ]  
    retrievalPath = "customGet.jsp"  
    { table-and-column-attributes }  
    { media-attributes } />
```

database-connection-attributes

書式

```
database-connection-attributes =  
    [ dataSourceName = "string | <%= jspExpression %>" ]  
    [ connCache = " <%= jspExpression %>" ]
```

説明

データベース接続を指定します。データベースからメディア・データを取得し、このデータをブラウザに配信するには、メディア配信コンポーネントがデータベース接続にアクセスできる必要があります。Multimedia Tag Library は、データベース接続情報を指定するための多くのメカニズムをサポートしています。アプリケーション開発者は、アプリケーションの要件および実装に基づいて適切なメカニズムを選択します。メカニズムには次のようなものがあります。

- データソース名 : OC4J では、JDBC データ・ソースのサポートが組み込まれています。データベースの *interMedia* オブジェクトからメディア・データを取得できるように、データ・ソースから取得された JDBC 接続は、データベース・オブジェクトおよびバイナリ・ラージ・オブジェクト (BLOB) へのアクセスをサポートしている必要があります。このため、システム固有の Oracle データ・ソースが必要です。

システム固有の Oracle データ・ソースは数多くあります。ただし、本番環境で実際に使用できるのは、`oracle.jdbc.pool.OracleConnectionCacheImpl` などの接続キャッシュ・メカニズムをサポートするデータ・ソースに限られます。各接続リクエストに対して新規 JDBC データベース接続を作成するような他のデータ・ソースを使用すると、パフォーマンスが大幅に低下します。特に、サムネイル・イメージのセットなど、多くのマルチメディア・データ項目が含まれるページの場合は、パフォーマンスの低下が顕著に現れます。

- Oracle JDBC 接続キャッシュ : JDBC 接続に関してステートレスであるアプリケーションの場合、JSP ページおよび JavaBeans では Oracle JDBC 接続キャッシュ機能を使用してデータベース接続を管理できます。このシナリオでは、JDBC 接続は各 HTTP リクエストの最初のキャッシュから取得され、リクエストの最後のキャッシュに戻されます。Multimedia Tag Library では、メディア・データを取得してブラウザに配信するために、メディア配信コンポーネントに `oracle.jdbc.pool.OracleConnectionCache` インタフェースを実装する接続キャッシュを使用するサポートが組み込まれています。
- OracleJSP ConnBeanCache JavaBean: この JavaBean は、`oracle.jdbc.pool.OracleConnectionCacheImpl` 接続キャッシュ・メカニズムを拡張したものです。したがって、この接続キャッシュ・メカニズムは Oracle JDBC 接続キャッシュの特殊な使用方法として暗黙的にサポートされています。

指定されたデータベース接続は、データベースからメディア・データを取得する際に、メディア配信コンポーネントにより使用されます。

パラメータ

dataSourceName

デフォルトの初期 Java Naming and Directory Interface (JNDI) コンテキストを使用して取得できる JDBC データ・ソースの名前。

connCache

次のいずれかの形式で表したデータベース接続情報。

- oracle.jdbc.pool.OracleConnectionCacheImpl などの oracle.jdbc.pool.OracleConnectionCache インタフェースを実装するクラスのインスタンス
- oracle.jsp.util.ConnBeanCache オブジェクトのインスタンス

使用方法

media-access-attributes を指定しない場合、database-connection-attributes および table-and-column-attributes を指定する必要があります。

例

この例では、connCache 属性を使用してデータベース接続を指定する方法を示しています。

```
<ord:embedVideo connCache = "<%= albumBean.getConnectionCache() %>"
    [ media-cache-control-attributes ]
    [ custom-retrieval-attributes ]
    { table-and-column-attributes }
    { media-attributes } />
```

media-access-attributes

書式

```
media-access-attributes =  
    mediaAccessUnit = "string | <%= jspExpression %>"  
    [ key = "string | <%= jspExpression %>" ]  
    [ rowid = "string | <%= jspExpression %>" ]
```

説明

取得タグの使用方法を単純にします。database-connection-attributes、table-and-column-attributes、media-cache-control-attributes および custom-retrieval-attributes 属性の情報は、事前定義の OrdJspTag.xml ファイルで指定できます。その後、取得タグ内でこれらの属性を media-access-attributes に置換できます。この属性では、OrdJspTag.xml ファイルで定義された情報を参照するために mediaAccessUnit 属性が使用されます。mediaAccessUnit 属性およびその他の属性 (media-cache-control-attributes など) を同時に使用した場合、タグ内で定義された属性は、OrdJspTag.xml ファイルで定義された属性よりも優先されます。(このタグ・ライブラリ構成ファイルの例は、[付録 A](#) を参照してください。)

パラメータ

mediaAccessUnit

OrdJspTag.xml ファイルで定義された access-unit 属性の名前を指定する文字列リテラルまたは式。

key

表内のメディア・データを検索する際に mediaAccessUnit 属性とともに機能する行情報。

rowid

表内のメディア・データを検索する際に mediaAccessUnit 属性とともに機能する行情報。

使用方法

database-connection-attributes および table-and-column-attributes を指定しない場合、media-access-attributes を指定する必要があります。

key 属性または rowid 属性のいずれかをタグで指定する必要があります。

例

この例では、2通りの方法で使った `mediaAccessUnit` 属性を示しています。1つ目は `key` 属性を使用し、2つ目は `rowid` 属性を使用しています。最初の6行のコードは、`access-unit` 属性により `photoUnit` が定義されている `OrdJspTag.xml` ファイルのセクションを表します。

```
<access-unit name="photoUnit"
             dataSourceName="myMediaDataDS"
             table="public_photos"
             keyColumn="ename"
             column="photo"
             expiration="3600+60"/>
```

```
<ord:embedImage mediaAccessUnit="photoUnit">
    key = "SMITH"
    { media-render-attributes }/>
```

または

```
<ord:embedImage mediaAccessUnit="photoUnit">
    rowid = "<%= empBean.getRowid() %>"
    { media-render-attributes }/>
```

media-cache-control-attributes

書式

```
media-cache-control-attributes =  
    [ expiration = "string | <%= jspExpression %>" ]  
    [ cache = "yes | no | no-remote | <%= jspExpression %>" ]
```

説明

Web キャッシュ制御用の **Surrogate-Control** レスポンス・ヘッダーの生成に役立ちます。このレスポンス・ヘッダーにより、元の Web サーバーではサロゲートによるレスポンス・エンティティの処理方法を指定できます。タグ内の **expiration** 属性の値は、**Surrogate-Control** 制御の **max-age** ディレクティブに変換されます。タグ内の **cache** 属性の値は、**Surrogate-Control** の **no-store** または **no-store-remote** ディレクティブに変換されます。

パラメータ

expiration

Surrogate-Control ヘッダーの **max-age** ディレクティブと同じ書式 (**expiration_time** [**+removal_time**]) の値。 **expiration_time** は、メディア・オブジェクトが有効とみなされる期間を秒数で指定します。この期間以降、キャッシュ実装では、キャッシュ・オブジェクトを無効とみなす必要があります。 **removal_time** は、有効期間がすぎてからオブジェクトがキャッシュから削除されるまでの遅延です。

cache

可能な値は次のとおりです。

- **no:Surrogate-Control** レスポンス・ヘッダー内の **no-store** ディレクティブ。
- **no-remote:Surrogate-Control** レスポンス・ヘッダー内の **no-store-remote** ディレクティブ。
- **yes:** Web キャッシュが使用されます。 **expiration** 属性が指定されていない場合、**Surrogate-Control** レスポンス・ヘッダー内の **max-age** ディレクティブは **infinity** に設定されます。

使用方法

なし。

例

例 1: この例は、`expiration` 属性の使用方を示しています。

```
<ord:embedVideo { database-connection-attributes }  
    expiration = "600"  
    [ custom-retrieval-attributes ]  
    { table-and-column-attributes }  
    { media-attributes } />
```

例 2: この例は、`cache` 属性の使用方を示しています。

```
<ord:embedVideo { database-connection-attributes }  
    cache = "no"  
    [ custom-retrieval-attributes ]  
    { table-and-column-attributes }  
    { media-attributes } />
```

table-and-column-attributes

書式

```
table-and-column-attributes =  
    table = "string | <%= jspExpression %>"  
    column = "string | <%= jspExpression %>"  
    [ key = "string | <%= jspExpression %>" ]  
    [ keyColumn = "string | <%= jspExpression %>" ]  
    [ rowid = "string | <%= jspExpression %>" ]
```

説明

データベース内のメディア・データの格納場所を指定します。データベースでメディア・データを識別するには、次の3通りの方法があります。

- 主キーを使用し、**key** 属性でキー値を指定します。
- 任意の列を使用し、**keyColumn** 属性で列名を指定し、**key** 属性で列値を指定します。
- ROWID を使用し、**rowid** 属性を指定します。

パラメータ

table

メディア・データが含まれる表の名前を指定した文字列リテラルまたは式。

column

メディア・データが含まれる列の名前を指定した文字列リテラルまたは式。

key

メディアのフェッチに使用するキー値を指定した文字列リテラルまたは式。**keyColumn** 属性が指定されていない場合、表の主キーが使用されます。表に主キーがない場合、**keyColumn** 属性によりキー列名を指定する必要があります。

keyColumn

メディアへのアクセスに使用する列を指定した文字列リテラルまたは式。

rowid

指定した表内のメディアの ROWID を示す文字列リテラルまたは式。この属性の指定は、**key="rowid-value" keyColumn="rowid"** を指定するのと同じです。

使用方法

`media-access-attributes` を指定しない場合、`table-and-column-attributes` および `database-connection-attributes` を指定する必要があります。

属性の `key`、`keyColumn` および `rowid` は、次のいずれかの組合せで指定する必要があります。

- `key`
- `key` および `keyColumn`
- `rowid`

例

例 1: この例は、`table`、`column` および `key` 属性の使用法を示しています。

```
<ord:embedVideo { database-connection-attributes }
  [ media-cache-control-attributes ]
  [ custom-retrieval-attributes ]
  table = "emp" column = "photo"
  key = "<%= empBean.getEmpno() %>"
  { media-attributes } />
```

例 2: この例は、`table`、`column`、`key` および `keyColumn` 属性の使用法を示しています。

```
<ord:embedVideo { database-connection-attributes }
  [ media-cache-control-attributes ]
  [ custom-retrieval-attributes ]
  table = "emp" column = "photo"
  key = "SMITH" keyColumn = "ename"
  { media-attributes } />
```

例 3: この例は、`table`、`column` および `rowid` 属性の使用法を示しています。

```
<ord:embedVideo { database-connection-attributes }
  [ media-cache-control-attributes ]
  [ custom-retrieval-attributes ]
  table = "emp" column = "photo"
  rowid = "<%= empBean.getRowid() %>"
  { media-attributes } />
```

media-render-attributes

以降の項で、メディア取得タグがそれぞれの固有の `media-render-attributes` により定義されています。

interMedia では、イメージ、オーディオおよびビデオ・データをサポートしています。したがって、*interMedia* マルチメディア取得タグでは、これらのメディア・タイプのみをサポートしています。テキストベース・タイプまたはアプリケーション固有のタイプなど、`OrdDoc` オブジェクト・タイプに格納されている場合があるその他のメディア・タイプはサポートしていません。

embedAudio

書式

```
<ord:embedAudio [ database-connection-attributes ]
    [ table-and-column-attributes ]
    [ custom-retrieval-attributes ]
    [ media-access-attributes ]
    [ media-cache-control-attributes ]

    [ height = "number | <%= jspExpression %>" ]
    [ width = "number | <%= jspExpression %>" ]
    [ alt = "string | <%= jspExpression %>" ]
    [ helperApp = "mediaPlayer | realPlayer |
        quicktimePlayer | <%= jspExpression %>" ]
    [ showControls = "true | false | <%= jspExpression %>" ]
    [ autoStart = "true | false | <%= jspExpression %>" ]
    [ loop = "true | false | <%= jspExpression %>" ]
    [ standby = "string | <%= jspExpression %>" ]
[ audio = "<%= jspExpression %>" ]/>
```

説明

オーディオ・オブジェクトをページに埋め込むための HTML `<OBJECT>` タグおよび `<EMBED>` タグを作成します。このタグにより、クライアントのブラウザで使用されるオーディオ・プレーヤを指定できます。さらに、共通のオーディオ属性セットが定義されます。サポートされているオーディオ形式の詳細は、『Oracle *interMedia* リファレンス』および『Oracle *interMedia* ユーザーズ・ガイド』を参照してください。

パラメータ

height

表示ウィンドウの高さ。表示ウィンドウのデフォルトの高さよりも優先されます。

width

表示ウィンドウの幅。表示ウィンドウのデフォルトの幅よりも優先されます。

alt

メディアを取得または表示できない場合に表示される簡潔な代替テキスト。

helperApp

ブラウザでメディアの再生に使用されるメディア・プレーヤの名前。これが指定されていない場合、ブラウザのデフォルト・プレーヤが起動されます。現在、Multimedia Tag Library では、Windows Media Player、RealPlayer および QuickTime Player という 3 つの主要メディア・プレーヤをサポートしています。ただし、Netscape ブラウザでは、メディアの MIME タイプに基づいてメディア・プレーヤ・プラグインが起動されます。したがって、Netscape ブラウザで起動されるメディア・プレーヤは、生成された HTML タグには制御されません。

showControls

プレーヤの制御コンポーネントを表示するかどうかを決定する属性。

autoStart

オーディオが取得された場合に自動的に再生するかどうかを決定する属性。

loop

オーディオを繰り返して再生するかどうかを決定する属性。

standby

オーディオの取得時に表示する内容を指定する属性。

audio

oracle.ord.im.OrdAudio オブジェクトのインスタンス。Multimedia Tag Library では、oracle.ord.im.OrdAudio オブジェクトからオーディオ情報を取得する必要があるため、JSP ページでオーディオ・オブジェクトが含まれる行がすでにデータベースからフェッチされている場合、Multimedia Tag Library を使用するには、audio 属性を使用する方法が最も効率的です。

使用方法

なし。

例

[「embedVideo」](#) の例を参照してください。

embedImage

書式

```
<ord:embedImage [ database-connection-attributes ]
                 [ table-and-column-attributes [
                 [ custom-retrieval-attributes ]
                 [ media-access-attributes ]
                 [ media-cache-control-attributes ]

                 [ height = "number | <%= jspExpression %>" ]
                 [ width = "number | <%= jspExpression %>" ]
                 [ border = "number | <%= jspExpression %>" ]
                 [ align = "left | right | top | bottom | middle |
                 <%= jspExpression %>" ]
                 [ alt = "string | <%= jspExpression %>" ]
                 [ longdesc = "string | <%= jspExpression %>" ]
                 [ image = "<%= jspExpression %>" ]/>
```

説明

イメージをページに埋め込むための HTML タグを作成します。**height**、**width**、**border** および **align** などの共通の タグ属性セットも定義されます。生成された タグには、**height** および **width** 属性が必ず含まれます。これらの属性がこのタグで指定されていない場合、このタグで指定されたイメージ・オブジェクトまたはデータベースからフェッチされたイメージ・オブジェクトのいずれかから取得されます。

パラメータ

height

表示ウィンドウの高さ。表示ウィンドウのデフォルトの高さよりも優先されます。

width

表示ウィンドウの幅。表示ウィンドウのデフォルトの幅よりも優先されます。

border

表示されるイメージの枠。

align

表示されるイメージの位置合せ。値には、**bottom**、**middle**、**top**、**left** または **right** があります。

alt

イメージを取得または表示できない場合に表示される簡潔な代替テキスト。

longdesc

イメージの詳細な（長い）説明へのリンク。alt 属性で指定された簡潔な説明を補足するものです。

image

oracle.ord.im.OrdImage オブジェクトのインスタンス。Multimedia Tag Library では、oracle.ord.im.OrdImage オブジェクトからイメージ情報を取得する必要があるため、JSP ページでイメージ・オブジェクトが含まれる行がすでにデータベースからフェッチされている場合、Multimedia Tag Library を使用するには、image 属性を使用する方法が最も効率的です。

使用方法

なし。

例

例 1: この例は、embedImage タグでの border、alt および align 属性の使用方法を示しています。

```
<ord:embedImage dataSourceName = "empDB"
  table = "emp" column = "photo"
  key = "<%= empBean.getEmpno() %>"
  alt = "Employee photo"
  border = "1"
  align = "middle" />
```

生成される HTML タグは、次のようになります。

```

```

例 2: この例は、イメージの実際の高さと幅よりも優先される height および width 属性の使用方法を示しています。

```
<ord:embedImage dataSourceName = "empDB"
  table = "emp" column = "photo"
  key = "SMITH" keyColumn = "ename"
  height = "100" width = "100" />
```

生成される HTML タグは、次のようになります。

```

```

例 3: この例は、`image` 属性の使用方を示しています。

```
<ord:embedImage dataSourceName = "empDB"  
table = "emp" column = "photo"  
rowid = "<%= empBean.getRowId() %>"  
image = "<%= empBean.getPhoto() %>"  
>
```

生成される HTML タグは、次のようになります。

```

```

項目の意味：

- **empBean:** データベース内の従業員スキーマへのアクセスに使用される `JavaBean` です。これには、文字列オブジェクトとしての `empno` および `OrdImage` オブジェクトとしての `photo` など、列値を戻す `get()` メソッド、さらに `OracleConnectionCacheImpl` オブジェクトなどのデータベース接続情報が含まれます。

embedVideo

書式

```
<ord:embedVideo [ table-and-column-attributes ]
  [ database-connection-attributes ]
  [ custom-retrieval-attributes ]
  [ media-access-attributes ]
  [ media-cache-control-attributes ]

  [ height = "number | <%= jspExpression %>" ]
  [ width = "number | <%= jspExpression %>" ]
  [ alt = "string | <%= jspExpression %>" ]
  [ helperApp = "mediaPlayer | realPlayer |
    quicktimePlayer | <%= jspExpression %>" ]
  [ showControls = "true | false | <%= jspExpression %>" ]
  [ autoStart = "true | false | <%= jspExpression %>" ]
  [ loop = "true | false | <%= jspExpression %>" ]
  [ standby = "string | <%= jspExpression %>" ]
  [ video = "<%= jspExpression %>" ]/>
```

説明

ビデオ・オブジェクトをページに埋め込むための HTML `<OBJECT>` タグおよび `<EMBED>` タグを作成します。このタグにより、クライアントのブラウザで使用されるビデオ・プレーヤを指定できます。さらに、共通のビデオ属性セットが定義されます。サポートされているビデオ形式の詳細は、『Oracle *interMedia* リファレンス』および『Oracle *interMedia* ユーザーズ・ガイド』を参照してください。

パラメータ

height

表示ウィンドウの高さ。表示ウィンドウのデフォルトの高さよりも優先されます。

width

表示ウィンドウの幅。表示ウィンドウのデフォルトの幅よりも優先されます。

alt

メディアを取得または表示できない場合に表示される簡潔な代替テキスト。

helperApp

ブラウザでメディアの再生に使用されるメディア・プレーヤの名前。これが指定されていない場合、ブラウザのデフォルト・プレーヤが起動されます。現在、Multimedia Tag Library では、Windows Media Player、RealPlayer および QuickTime Player という 3 つの主要メディア・プレーヤをサポートしています。ただし、Netscape ブラウザでは、メディアの MIME タイプに基づいてメディア・プレーヤ・プラグインが起動されます。したがって、Netscape ブラウザで起動されるメディア・プレーヤは、生成された HTML タグには制御されません。

showControls

プレーヤの制御コンポーネントを表示するかどうかを決定する属性。

autoStart

ビデオが取得された場合に自動的に再生するかどうかを決定する属性。

loop

ビデオを繰り返して再生するかどうかを決定する属性。

standby

ビデオの取得時に表示する内容を指定する属性。

video

oracle.ord.im.OrdVideo オブジェクトのインスタンス。Multimedia Tag Library では、oracle.ord.im.OrdVideo オブジェクトからビデオ情報を取得する必要があるため、JSP ページでビデオ・オブジェクトが含まれる行がすでにデータベースからフェッチされている場合、Multimedia Tag Library を使用するには、video 属性を使用する方法が最も効率的です。

使用方法

なし。

例

この例では、embedVideo タグでの media-render-attributes の使用方法を示しています。この例では、helperApp 属性に、考えられるすべてのメディア・プレーヤの選択肢が含まれます。

次の例では、embedImage の例と同じ JavaBean (empBean) を使用しています。その後にくる 3 つのサンプルは、3 つのメディア・プレーヤに対応する HTML 出力を示しています。

```
<ord:embedVideo dataSourceName = "empDB"
    table = "emp" column = "greetings"
    key = "<%= empBean.getEmpno() %>"
    alt = "Employee greetings"
    height = 240 width = 300
    helperApp = "mediaPlayer/realPlayer/quicktimePlayer"
    showControls = "true"
```

```
autoStart = "true"  
loop = "true"  
standby = "Loading media player components ..." />
```

サンプル出力 1: この例は、Windows Media Player について生成された HTML タグを示しています。ファイル拡張子 .avi を持つビデオ・データが使用され、helperApp 属性の値が mediaPlayer に設定されています。

```
<OBJECT  
  ID="mediaPlayer"  
  CLASSID= "clsid:22D6F312-B0F6-11D0-94AB-0080C74C7E95"  
  CODEBASE=  
"http://activex.microsoft.com/activex/controls/mplayer/en/nsmp2inf.cab#Version=5,1,5  
2,701"  
  TYPE="application/x-oleobject"  
  STANDBY="Loading media player components..."  
  HEIGHT="240" WIDTH="300">  
  <PARAM NAME="showcontrols" VALUE="true">  
  <PARAM NAME="autostart" VALUE="true">  
  <PARAM NAME="loop" VALUE="true">  
  <PARAM NAME="filename"  
VALUE="OrdGetMediaServlet?dataSourceName=empDB&table=emp&column=greetings&key=1&time  
Stamp=111078352&ext=.avi">  
  
  <EMBED TYPE="video/x-msvideo"  
  STANDBY="Loading media player components..."  
  CONTROLLER="true"  
  CONTROLS="ImageWindow,ControlPanel"  
  SHOWCONTROLS="true"  
  AUTOSTART="true"  
  LOOP="true"  
  HEIGHT="240" WIDTH="300"  
  
SRC="OrdGetMediaServlet?dataSourceName=empDB&table=emp&column=greetings&key=1&timeSt  
amp=111078352&ext=.avi" >
```

項目の意味：

- .avi: Microsoft AVI メディアのファイル拡張子です。
- mediaPlayer: helperApp 属性の値です。
- video/x-msvideo: メディア・データの MIME タイプです。

サンプル出力 2: この例は、RealPlayer について生成された HTML タグを示しています。ファイル拡張子 .rm を持つビデオ・データが使用され、helperApp 属性の値が realPlayer に設定されています。


```

<OBJECT
  ID="RVOCX"
  CLASSID= "clsid: CFCDA03-8BE4-11cf-B84B-0020AFBCCFA"
  STANDBY="Loading media player components..."
  HEIGHT="240" WIDTH="300" >
  <PARAM NAME="controls" VALUE="ImageWindow, ControlPanel">
  <PARAM NAME="autostart" VALUE="true">
  <PARAM NAME="loop" VALUE="true">
  <PARAM NAME="filename"
  VALUE="OrdGetMediaServlet?dataSourceName=empDB&table=emp&column=greetings&key=1&time
  Stamp=111078352&ext=.rm">

  <EMBED TYPE="audio/x-pn-realaudio-plugin"
  STANDBY="Loading media player components..."
  CONTROLLER="true"
  CONTROLS="ImageWindow,ControlPanel"
  SHOWCONTROLS="true"
  AUTOSTART="true"
  LOOP="true"
  HEIGHT="240" WIDTH="300"

  SRC="OrdGetMediaServlet?dataSourceName=empDB&table=emp&column=greetings&key=1&timeSt
  amp=111078352&ext=.rm" >
  <NOEMBED>
  <P>Employee Greetings.</P>
  </NOEMBED>
</OBJECT>

```

項目の意味：

- .rm: RealNetworks Real Video メディアのファイル拡張子です。
- realPlayer: helperApp 属性の値です。
- audio/x-pn-realaudio-plugin: メディア・データの MIME タイプです。

サンプル出力 3: この例は、QuickTime Player について生成された HTML タグを示しています。ファイル拡張子 .mov を持つビデオ・データが使用され、helperApp 属性の値が quicktimePlayer に設定されています。

```

<OBJECT
  CLASSID= "clsid: 02BF25D5-8C17-4B23-BC80-D3488ABDDC6B"
  CODEBASE= "http://www.apple.com/qtactivex/qtplugin.cab"
  STANDBY="Loading media player components..."
  HEIGHT="240" WIDTH="300" >
  <PARAM NAME="controller" VALUE="true">
  <PARAM NAME="autostart" VALUE="true">

```

```
<PARAM NAME="loop" VALUE="true">
<PARAM NAME="filename"
VALUE="OrdGetMediaServlet?dataSourceName=empDB&table=emp&column=greetings&key=1&time
Stamp=111078352&ext=.mov">

<EMBED TYPE="video/quicktime"
STANDBY="Loading media player components..."
CONTROLLER="true"
CONTROLS="ImageWindow,ControlPanel"
SHOWCONTROLS="true"
AUTOSTART="true"
LOOP="true"
HEIGHT="240" WIDTH="300"

SRC="OrdGetMediaServlet?dataSourceName=empDB&table=emp&column=greetings&key=1&timeSt
amp=111078352&ext=.mov" >
<NOEMBED>
<P>Employee Greetings.</P>
</NOEMBED>
</OBJECT>
```

項目の意味：

- .mov: Apple QuickTime メディアのファイル拡張子です。
- quicktimePlayer: helperApp 属性の値です。
- video/quicktime: メディア・データの MIME タイプです。

mediaUrl

書式

```
<ord:mediaUrl [ database-connection-attributes ]  
    [ table-and-column-attributes ]  
    [ custom-retrieval-attributes ]  
    [ media-access-attributes ]  
    [ media-cache-control-attributes ]  
  
    id = "string" >  
  
    tag body..  
  
</ord:mediaUrl>
```

説明

タグ本体に使用できるメディア取得 URL オブジェクトを生成します。

パラメータ

id

スクリプト変数の名前。データベース内のメディア・オブジェクトの URL を取得するには、このスクリプト変数を使用しながら、`getUrl()` メソッドをコールします。

使用方法

なし。

例

例 1: この例では、カスタマイズした HTML タグ内で生成された URL を使用しています。

```
<ord:mediaUrl dataSourceName = "empDB"
               table = "emp" column = "photo"
               key = "1"
               id = "photo" >

    

</ord:mediaUrl>
```

例 2: この例では、従業員データベースのイメージ・ファイルを検索するため、HTML <A> (リンク) タグ内で生成された URL を使用しています。

```
<ord:mediaUrl dataSourceName = "empDB"
               table = "emp" column = "photo"
               key = "1"
               id = "photo" >

    <a href="<%= photo.getUrl() %>"> Click here to view image </a>

</ord: mediaUrl>
```

メディア取得 URL の書式

書式

```
<mediaRetrievalPath>?<databaseConnection>&<tableInfo>&<columnInfo>&<rowInfo>&<expiration>&<cache>&  
<updateTimeStamp>&<fileExtension>
```

説明

メディア取得タグにより生成される URL 文字列の書式を指定します。この URL 文字列は、メディア・データを取得するためにメディア配信コンポーネントに渡されます。

パラメータ

mediaRetrievalPath

メディア取得タグで指定された `retrievalPath` 属性の値。デフォルトでは、この値は `OrdGetMediaServlet` です。

databaseConnection

次のいずれかになります。

- `dataSourceName=name:`
name: タグで指定されたデータ・ソース名です。
- `IMConnCacheName=name:`
name: タグ・ライブラリにより指定された接続キャッシュ名です。

tableInfo

`table=tableName:`

tableName: タグで指定された名前です。

columnInfo

`column=columnName:`

columnName: タグで指定された名前です。

rowInfo

タグで指定された行情報に応じて、次のいずれかになります。

- `key=keyVal`
- `rowid=rowidVal`
- `key=keyVal&keyColumn=keyColumnName`

expiration

オプションのキャッシュ制御属性 `expiration=expirationValue`:

expirationValue: タグで指定された値です。

cache

オプションのキャッシュ制御属性 `cache=cacheValue`:

cacheValue: タグで指定された値です。

updateTimeStamp

`timeStamp=timestampvalue`:

timestampvalue: メディア・オブジェクトの最終更新時刻です。

fileExtension

`ext=.fileExt`:

fileExt: メディアのファイル拡張子です。

使用方法

HTML の ``、`<OBJECT>` および `<EMBED>` タグは、それぞれマルチメディア JSP タグの `embedImage`、`embedAudio` および `embedVideo` により生成され、これらの HTML タグ内の SRC 属性にも前述の形式が適用されます。これらのマルチメディア JSP タグの詳細は、[「embedImage」](#)、[「embedAudio」](#) および [「embedVideo」](#) を参照してください。

例

[「embedImage」](#) の例を参照してください。

メディアのアップロードに関する参照情報

Oracle Application Server 10g Multimedia Tag Library for JSP (「Multimedia Tag Library」) で提供されるメディア・アップロード・タグは、マルチメディア・アプリケーションでメディア・データをデータベースにアップロードするのに役立ちます。メディア・アップロード・タグは次のとおりです。

- storeMedia
- uploadFile
- uploadFormData

メディア・アップロード・タグは、次の要件を満たします。

- マルチパート・フォームデータ・リクエストの処理に関する操作性の向上
- データベースにおけるメディア・データの格納の単純化

アプリケーション開発者は、メディア・データをアップロードする場合、次の情報を指定する必要があります。

- データベースにメディア・データを書き込む際に使用するデータベース接続情報
- データベースにメディア・データをアップロードする際の場所を示した表、列およびキー情報

前提条件

なし。

参照情報

この項では、*interMedia* オブジェクト上で動作するメディア・アップロード・タグに関する参照情報を提供します。

storeMedia

書式

```
<ord:storeMedia conn = "<%= jspExpression %>"
  table = "string | <%= jspExpression %>"
  [ key = "string | <%= jspExpression %>" ]
  [ keyColumn = "string | <%= jspExpression %>" ]
  [ rowid = "string | <%= jspExpression %>" ]
  mediaColumns = "values"
  mediaParameters = "values"
  [ otherColumns = "values" ]
  [ otherValues = "<%= jspExpression %>" ]/>
```

where,

```
values = string | <%=jspExpression%> [, string | <%=jspExpression%>]...
```

説明

uploadFormData タグの本体内部で動作します。storeMedia タグでは、uploadFormData タグにより作成されたフォームデータ・オブジェクトが mediaParameters 属性により暗黙的に使用されます。このタグでは、アップロードされたメディア・データが、HTML フォームからデータベースの指定された表、行および列の OrdImage、OrdAudio、OrdVideo または OrdDoc オブジェクトにロードされます。指定された行が存在しない場合、新しい行が挿入され、ロードされたデータとともに更新されます。データベース接続オブジェクトで自動コミットが TRUE に設定されている場合、トランザクションが自動的にコミットされます。

mediaColumns および mediaParameters 属性により、HTML フォームで指定された複数のメディア・データを特定の行の複数の列にアップロードできます。otherColumns および otherValues 属性により、表の他の列をメディア・データとともに更新できます。

データベース BLOB でメディア・データがロードされた後、このタグにより setProperties() メソッドがコールされ、オブジェクト内のメディア・プロパティが設定されます。メディア形式が interMedia により認識されない場合、このタグでは mimeType および length プロパティのみが設定されます。

注意： 指定された行が存在せず、新しい行が表に挿入された場合、mediaColumns および otherColumns 属性で指定されていない表内の他の列は、NULL 値を受け入れる必要があります。

パラメータ

conn

メディア・データの格納に使用される JDBC 接続 (java.sql.Connection オブジェクト)。

table

メディア・データが含まれる表の名前を指定した文字列リテラルまたは式。

key

メディアのフェッチに使用するキー値を指定した文字列リテラルまたは式。keyColumn 属性が指定されていない場合、表の主キーが使用されます。表に主キーがない場合、keyColumn 属性を使用してキー列名を指定する必要があります。

keyColumn

メディアへのアクセスに使用する列を指定した文字列リテラルまたは式。

rowid

指定した表内のメディアの ROWID を示す文字列リテラルまたは式。この属性の指定は、key="rowid-value" keyColumn="rowid" を指定するのと同じです。

mediaColumns

アップロードされたメディア・データとともにロードされる列の名前。

mediaParameters

メディア・データのアップロードに使用される、HTML フォームで指定されたパラメータの名前。

otherColumns

表で更新または挿入される他の列の名前。

otherValues

otherColumns 属性に対応する表内の他の列の更新または挿入される値。これらの値は、java.util.Vector オブジェクトに格納される必要があります。

例外

なし。

使用方法

なし。

例

例 1: この例は、表の特定の行にイメージをアップロードし、その行の他の列を同時に更新する方法を示しています。

```
<ord:uploadFormData formDataId="fd">
  <%
    java.util.Vector otherValuesVector = new java.util.Vector();
    otherValuesVector.add(fd.getParameter("desc"));
    otherValuesVector.add(fd.getParameter("loc"));
  %>
  <ord:storeMedia conn = "<% albumBean.getConnection() %>"
    table = "photos"
    key = "<%= fd.getParameter( "desc" ) %>"
    mediaColumns = "photo"
    mediaParameters = "photo"
    otherColumns = "Description, Location"
    otherValues = "<%= otherValuesVector %>" />
</ord:uploadFormData>
```

例 2: この例は、表の特定の行に、イメージ・データとオーディオ・データを同時にアップロードする方法を示しています。

```
<ord:uploadFormData formDataId="fd">
  <%
    java.util.Vector otherValuesVector = new java.util.Vector();
    otherValuesVector.add(fd.getParameter("desc"));
  %>
  <ord:storeMedia conn = "<% albumBean.getConnection() %>"
    table = "photos"
    key = "<%= fd.getParameter( "desc" ) %>"
    mediaColumns = "photo, music"
    mediaParameters = "photo, music"
    otherColumns = "Description"
    otherValues = "<%= otherValuesVector %>" />
</ord:uploadFormData>
```

uploadFile

書式

```
<ord:uploadFile parameter = "string | <%= jspExpression %>"
                 [ mimeType = "string" ]
                 [ length = "string" ]
                 [ fullFileName = "string" ]
                 [ shortFileName = "string" ]
                 [ inputStream = "string" ] >
    file-information-processing
</ord:uploadFile>
```

説明

アップロード・ファイルの情報にアクセスできるようにします。このタグでは、`uploadFormData` タグにより作成されたオブジェクトが `parameter` 属性により暗黙的に使用されます。このタグは、`uploadFormData` タグ内にネストされる必要があります。

パラメータ

parameter

ファイルのアップロードに使用される、HTML フォームで指定されたパラメータの名前。

mimeType

ファイルの MIME タイプに対応するスクリプト変数名。

length

ファイルの長さに対応するスクリプト変数名。

fullFileName

ファイルの完全なファイル名に対応するスクリプト変数名。

shortFileName

ファイルの短縮ファイル名に対応するスクリプト変数名。

inputStream

アップロード・ファイルの `java.io.InputStream` オブジェクトに対応するスクリプト変数名。

例外

なし。

使用方法

なし。

例

この例は、`uploadFormData` タグ内にネストされた `uploadFile` タグを使用して、MIME タイプ、長さおよび名前などの様々なファイル属性にアクセスできるようにする方法を示しています。

```
<ord:uploadFormData formDataId="fd">
  <%
    String desc = fd.getParameter( "desc" );
    String loc fd.getParameter( "loc" );
    <ord:uploadFile parameter = "photo"
      mimeType = "photoType"
      length = "photoLength"
      fullFileName = "photoFullName"
      shortFileName = "photoShortName">
      inputStream = "photoIn">

    The mimeType of the file is <%=photoType%>
    The length is <%=photoLength%>
    .
    .
    .

  </ord:uploadFile>
  %>
</ord:uploadFormData>
```

項目の意味：

- `photo`: HTML フォーム内のアップロード・ファイルのパラメータ名です。
- `photoType`: メディアの MIME タイプが含まれる変数の名前です。
- `photoLength`: アップロード・ファイルのファイルの長さが含まれる変数の名前です。
- `photoFullName`: アップロード・ファイルの完全なファイル名が含まれる変数の名前です。
- `photoShortName`: アップロード・ファイルの短縮ファイル名が含まれる変数の名前です。
- `photoIn`: アップロード・ファイルの `java.io.InputStream` オブジェクトのインスタンスです。

uploadFormData

書式

```
<ord:uploadFormData formDataId = "string"
    [ releaseFormData = "true | false | <%= jspExpression %>" ]
    [ maxMemory = "number | <%= jspExpression %>" ]
    [ tempDir = "string | <%= jspExpression %>" ]
    form-data-processing
</ord:uploadFormData>
```

説明

マルチパート / フォームデータ HTTP リクエストを解析し、ブラウザから Web サーバーに転送されるアップロード・ファイルのテキストベース・フォーム・パラメータおよびコンテンツにアクセスできるようにします。

パラメータ

formDataId

oracle.ord.im.OrdHttpUploadFormData タイプのスクリプト変数名。JSP ページでは、ここからアップロード・ファイルのテキストベース・フォーム・パラメータおよびコンテンツを取得できます。

releaseFormData

インクルードされた JSP ページまたはリクエストの転送先となる JSP ページなどで、`uploadFormData` タグの後で処理できるようアップロード・メディアを保持するために使用される属性。デフォルトでは、この属性の値は **TRUE** であり、タグの最後にすべてのアップロード・メディアが解放されます。アップロード・メディアを保持するには、この属性の値を **FALSE** に設定します。その後、アップロード・メディアが不要になった場合、`OrdHttpUploadFormData` オブジェクトの `release()` メソッドをコールしてリソースを解放します。

maxMemory

メディアを一時的にディスクに格納する前に、1 つの HTTP POST リクエストに対してアップロード・メディアにより使用できるメモリの最大量を指定する属性。デフォルトでは、すべてのアップロード・メディアが、アプリケーションによりデータベースにロードされてタグの最後に解放されるまでは、メモリーに格納されます。ユーザーがサイズの大きいメディア・ファイル（サイズの大きいビデオ・クリップなど）をアップロードする場合、アプリケーション開発者は、アップロード・ファイルがデータベースにロードされて解放される前に、一時的にディスクに格納することを選択できます。

tempDir

デフォルトの動作を無効にできる属性。この属性が指定されていない場合、デフォルトでは、一時的にディスクに格納されるアップロード・メディアが、`javax.servlet.context.tempdir` および `java.io.tmpdir` プロパティにより識別される 2 つの場所のいずれかに格納されます。

項目の意味：

- `javax.servlet.context.tempdir` が存在する場合、`java.io.tmpdir` よりも優先されます。

注意： `uploadFormData` タグの最後より前に JSP ページの実行が割り込まれた場合、ガベージ・コレクション時またはユーザーのセッション終了時（どちらか早い方）に、テンポラリ・ファイルが削除されます。

例外

なし。

使用方法

このタグにより作成されるフォームデータ・オブジェクトは、`oracle.ord.im.OrdHttpUploadFormData` タイプのオブジェクトです。フォーム・データをアップロードするためのアクセス権は、次のメソッドにより提供されます。

- 文字列 `getParameter(文字列名)`
- 文字列 [] `getParameterValues(文字列名)`
- 列挙 `getParameterNames()`
- `OrdHttpUploadFile` `getFileParameter(文字列名)`
- `OrdHttpUploadFile` [] `getFileParameterValues(文字列名)`
- 列挙 `getFileParameterNames()`

例

この例は、マルチメディア JSP タグ `uploadFormData` を使用して、フォームデータ・オブジェクトを作成する方法を示しています。このフォームデータ・オブジェクトは、`albumBean` と呼ばれる `JavaBean` がテキストベース・フォーム・パラメータおよびアップロード・ファイルのコンテンツにアクセスする際に使用されます。

```
<ord:uploadFormData formDataId="fd">
  <%
    albumBean.setDescription( fd.getParameter( "desc" ) );
    albumBean.setLocation( fd.getParameter( "loc" ) );
    albumBean.insertNewEntry();
    albumBean.fetchRow();
  %>
</ord:uploadFormData>
```

```
albumBean.loadPhoto( fd.getFileParameter( "photo" ) );
albumBean.updateRow();
albumBean.commit();
%>
</ord:uploadFormData>
```

インストールおよび構成に関する情報

この付録では、Oracle Application Server および他の環境における Oracle Application Server 10g Multimedia Tag Library for JSP (「Multimedia Tag Library」) のインストールおよび構成に関する情報を提供しています。

前提条件

Multimedia Tag Library をインストールおよび使用する前に、システムに次の製品をインストールしておく必要があります。

- Oracle9i リリース 2 (9.2)

Oracle9i の詳細は、次の Oracle Technology Network Japan の Web サイトを参照してください。

<http://otn.oracle.co.jp/>

- JSP コンテナ (OC4J リリース 2)

OC4J の詳細は、次の Oracle Technology Network Japan の Web サイトを参照してください。

<http://otn.oracle.co.jp/software/index.html>

注意： どの JSP コンテナを選択しても、Oracle ホーム・ディレクトリにインストールする必要があります。

Multimedia Tag Library のインストール

Multimedia Tag Library のインストールには、Java クラス・ライブラリ、タグ・ライブラリ・ディスクリプタ (TLD)・ファイルおよびデフォルトのメディア配信コンポーネント OrdGetMediaServlet の 3 つのコンポーネントが含まれます。これらの 3 つのコンポーネントは、ordjsptag.jar ファイルにパッケージされています。

Oracle Application Server のインストール時、またはスタンドアロン・バージョンの OC4J のみをインストールした後に、Multimedia Tag Library をインストールできます。Oracle Application Server のインストールの一環として Multimedia Tag Library をインストールした場合、JAR ファイルが自動的に次の OC4J ディレクトリにコピーされます。

```
<ORACLE_HOME>/j2ee/home/jsp/lib/taglib
```

OC4J スタンドアロンのインストール後に Multimedia Tag Library をインストールするには、次の Oracle Technology Network Japan の Web サイトの指示に従ってください。

<http://otn.oracle.co.jp/>

Multimedia Tag Library の構成

この項では、Multimedia Tag Library を使用して、アプリケーションを配置する際に、場合によっては指定する必要がある構成パラメータおよび配置構成について説明します。

TLD ファイルの仮想パスの指定

Multimedia Tag Library の TLD ファイルは、Multimedia Tag Library の JAR ファイル (ordjsptag.jar) に含まれています。Web アプリケーションの JSP ページで Multimedia Tag Library を使用するには、Web アプリケーションの web.xml ファイルで TLD ファイルの仮想パスを指定する必要があります。例を次に示します。

```
<taglib>
  <taglib-uri>
    intermedia-taglib.tld
  </taglib-uri>
  <taglib-location>
    http://xmlns.oracle.com/j2ee/jsp/tld/ordim/intermedia-taglib.tld
  </taglib-location>
</taglib>
```

web.xml ファイルでは、タグ・ライブラリを識別するために taglib-uri 要素が使用されています。taglib-uri 要素の値は、Web アプリケーションの JSP ページ内の JSP taglib ディレクティブに対する uri 属性値と一致する必要があります。taglib-location 要素は、タグ・ライブラリの TLD ファイルの位置を特定するために使用されます。この値は、TLD ファイルが格納されている URL です。

Multimedia Tag Library が使用されている Web アプリケーションの JSP ページに、JSP taglib ディレクティブに対する uri 属性の値として、ファイル名 intermedia-taglib.tld を入力します。

メディア配信コンポーネントの指定

Multimedia Tag Library では、メディア配信サーブレットおよびメディア配信 JSP ページという 2 つのメディア配信コンポーネントが提供されます。デフォルトでは、メディア配信 JSP ページを指定するか、メディア配信サーブレットを使用できます。Web アプリケーションでは、タグ・ライブラリ構成ファイル OrdJspTag.xml の media-retrieval-path 要素を定義することで、使用するメディア配信コンポーネントを指定できます。

メディア配信サーブレット oracle.ord.im.jsp.OrdGetMediaServlet は、タグ・ライブラリとともにパッケージされており、タグ・ライブラリとともにインストールされます。これはデフォルトのメディア配信コンポーネントです。次の例のように、Web アプリケーションの web.xml ファイルで、サーブレットへの仮想パスを指定します。

```
<servlet>
  <servlet-name>oracle_ord_im_jsp_media_servlet</servlet-name>
  <servlet-class>oracle.ord.im.jsp.OrdGetMediaServlet</servlet-class>
```

```
</servlet>

<servlet-mapping>
  <servlet-name>oracle_ord_im_jsp_media_servlet</servlet-name>
  <url-pattern>/OrdGetMediaServlet</url-pattern>
</servlet-mapping>
```

メディア配信 JSP ページ `OrdGetMediaJsp.jsp` は、OTN-J からダウンロードできます。最初に、タグ・ライブラリ構成ファイルで JSP ページを定義します。その後、JSP ページを Web アプリケーションの JSP ディレクトリにコピーします。

メディア・データの配信に JSP ページを使用する方法は、指定されたページをアプリケーションの JSP ページ・ディレクトリにコピーするだけで他の構成は必要ないため、便利です。しかし、パフォーマンス上の理由から、JSP ページではなくサーブレットからメディア・データを配信することがよい、あるいは必要な場合があります。

メディア・データへのアクセスの許可

また、URL を構成するすべての Multimedia Tag Library アクションにより、各ユーザー・セッション内で表名および列名の情報が登録されます。デフォルトでは、メディア取得リクエストに回答する際に、メディア配信コンポーネントによりこの情報が検証されます。検証チェックに失敗した場合、メディア取得リクエストは拒否されます。こうした検証によって、悪意のあるユーザーが、ユーザーが作成した URL を使用してデータ・ソース、表または列からメディア・データを取得できなくなります。

一部のアプリケーションでは、特定の表および列への一般のアクセスをサポートする必要があります。（例：ユーザーがメディア URL をメール・メッセージで交換する場合があるアプリケーションでは、データへの一般のアクセスが必要です。）これらのアプリケーションでは、特定のデータ・ソースの表内の列に誰でもアクセスできるメカニズムが提供されます。このメカニズムでは、タグ・ライブラリ構成ファイル `OrdJspTag.xml` 内の `access-control` 属性の設定が必要です。

注意： このメカニズムを使用するには、アプリケーションにおいて、タグに `dataSourceName` 属性を使用してデータベース接続情報を指定する必要があります。

キャッシュ制御属性の設定

Multimedia Tag Library では、キャッシュ制御属性の限定された設定が可能です。今回のバージョンでは、指定された表内の、名前が付けられた特定の列から取得されるすべてのメディアに対し、Surrogate-Control ヘッダーの max-age、no-store または no-store-remote ディレクティブを設定できます。これらのディレクティブの値は、タグ属性 expiration および cache により指定されます。

expiration 属性の正確な書式は次のとおりです。

```
expiration_time[+removal_time]
```

この場合、expiration_time は生存可能時間 (TTL) で表され、removal_time は、リソースが失効した後キャッシュから削除されるまでの遅延です。

どちらの値も秒数で指定します (例:1800 または 10000+600)。

cache 属性の値は、no、no-remote または yes です。値 no および no-remote は、Surrogate-Control ヘッダーでそれぞれ no-store および no-store-remote ディレクティブに変換されます。値 yes は、Web キャッシュが使用されることを意味します。

Oracle Web Cache の詳細は、第 3 章の「[media-cache-control-attributes](#)」および『Oracle Application Server Web Cache 管理者ガイド』を参照してください。

注意： キャッシュ制御属性の設定についてより高度なファイングレイイン制御が必要なアプリケーションでは、デフォルトのメディア配信コンポーネントを替えて、独自のメディア配信メカニズムを設計する必要があります。あるいは、こうしたアプリケーションに Multimedia Tag Library を使用することは避けてください。

サンプル構成ファイル

ここでは、タグ・ライブラリ構成ファイル OrdJspTag.xml の例を示します。この XML ファイルはアプリケーション固有であるため、Web アプリケーションの web.xml ファイルと同じディレクトリに格納します。タグ・ライブラリ構成ファイルは最初の要求で読み取られ、以降の取得のためにアプリケーション・コンテキストに格納されます。

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE ordim-tag-library [
<!ELEMENT ordim-tag-library (media-control?, media-access?, media-player*,
media-retrieval-path?)>
<!ELEMENT media-control (data-source?)>
<!ELEMENT data-Source (table+)>
<!ELEMENT table (column+)>
<!ELEMENT column (access-control)>
<!ELEMENT access-control EMPTY>
<!ELEMENT media-access (access-unit+)>
```

```
<!ELEMENT access-unit EMPTY>
<!ELEMENT media-player (class-id)>
<!ELEMENT class-id EMPTY>
<!ELEMENT media-retrieval-path (#PCDATA)>
<!ATTLIST media-control filtering (true | false) #IMPLIED>
<!ATTLIST data-source name CDATA #REQUIRED>
<!ATTLIST table name CDATA #REQUIRED>
<!ATTLIST column name CDATA #REQUIRED>
<!ATTLIST access-control enabled (true | false) #REQUIRED>
<!ATTLIST access-unit name CDATA #REQUIRED>
<!ATTLIST access-unit dataSourceName CDATA #IMPLIED>
<!ATTLIST access-unit table CDATA #REQUIRED>
<!ATTLIST access-unit keyColumn CDATA #IMPLIED>
<!ATTLIST access-unit column CDATA #REQUIRED>
<!ATTLIST access-unit expiration CDATA #IMPLIED>
<!ATTLIST access-unit cache (yes|no|no-remote) #IMPLIED>
<!ATTLIST media-player name (mediaPlayer|realPlayer|quicktimePlayer) #REQUIRED>
<!ATTLIST class-id clsid CDATA #REQUIRED>
]>

<ordim-tag-library>
<media-control filtering="true">
  <data-source name="myMediaDataDS">
    <table name="public_photos">
      <column name="photo">
        <access-control enabled="false"/>
      </column>
      <column name="thumb">
        <access-control enabled="false"/>
      </column>
    </table>
    <table name="public_videos">
      <column name="movie">
        <access-control enabled="false"/>
      </column>
    </table>
  </data-source>
</media-control>
<media-access>
  <access-unit name="photoUnit"
    dataSourceName="myMediaDataDS"
    table="public_photos"
    keyColumn="ename"
    column="photo"
    expiration="3600+60"/>
  <access-unit name="thumbUnit"
    dataSourceName="myMediaDataDS"
```

```
        table="public_photos"
        keyColumn="ename"
        column="photo"
        expiration="3600+60"/>
<access-unit name="movieUnit"
  dataSourceName="myMediaDataDS"
  table="public_videos"
  column="movie"
  expiration="3600+60"
  cache="no"/>
<access-unit name="mp3Unit"
  table="musics"
  column="mp3"
  expiration="3600+60"
  cache="no"/>
</media-access>
<media-player name="mediaPlayer">
  <class-id clsid="22D6F312-B0F6-11D0-94AB-0080C74C7E95"/>
</media-player>
<media-retrieval-path>
  GetMediaServlet
</media-retrieval-path>
</ordim-tag-library>
```

OrdJspTag.xml ファイルでは、柔軟性と簡潔さを持つメディア取得タグが提供されます。このファイルには、**media-control**、**media-access**、**media-player** および **media-retrieval-path** の 4 つの要素が含まれます。

media-control 要素は、メディア・データへのアクセスを制御するために使用されます。この要素はオプションです。デフォルトでは、メディア取得タグが使用されている場合、そのセッション内でのみすべてのメディア・データへのアクセスが許可されます。ただし、アプリケーション・マネージャが **access-control** サブ要素の **enabled** 属性を **FALSE** に設定すれば、セッション外ですべてのクライアントがメディア・データにアクセスできるようになります。

media-control サブ要素の **filtering** 属性は **TRUE** に設定できます。**TRUE** に設定すると、HTML ファイルがブラウザに配信される際に、メディア配信コンポーネントにより特殊文字がフィルタ処理されます。

media-access 要素は、メディア・アクセス・ユニットの定義に使用されます。これによって、メディア取得タグの使用が簡単になります。この要素はオプションです。各 **access-unit** サブ要素により、名前と、メディア取得タグの共通属性 **database-connection-attributes**、**table-and-column-attributes** および **media-cache-control-attributes** に対応する属性を持つメディア・アクセス・ユニットが定義されます。各メディア・アクセス・ユニットで定義された属性は、メディア取得タグで定義された属性によって無効になる場合があります。

media-player 要素により、メディア・プレーヤについて生成された HTML **<OBJECT>** タグで使用されるクラス識別子 (**clsid**) が定義されます。この要素は、アプリケーションにおい

て、Internet Explorer ブラウザで特定のメディア・プレーヤを使用する必要がある場合のみ定義されます。

`media-retrieval-path` 要素は、Multimedia Tag Library で使用されるメディア配信コンポーネントの指定に使用されます。この要素は、アプリケーションにおいて、デフォルト (`OrdGetMediaServlet`) 以外のメディア配信コンポーネントを使用する必要がある場合のみ定義されます。

エラー・メッセージ

この付録では、Oracle Application Server 10g Multimedia Tag Library for JSP（「Multimedia Tag Library」）により発生する可能性があるエラーについて説明します。これらのエラーは、パブリック・オブジェクトのコンテンツを取得しようとしたときに表示される場合があります。

アップロード・リクエストではありません。

原因：アップロード・リクエストが、マルチパート / フォームデータのエンコード書式を使用してエンコードされていません。

処置：マルチパート / フォームデータのエンコード書式を使用してアップロード・リクエストを送信します。

タグ < タグ名 > は uploadFormData タグ内にネストされる必要があります

原因：指定されたタグが uploadFormData タグ内にネストされていません。

処置：指定されたタグは、uploadFormData タグの本体に表示される必要があります。

フォーム・フィールド:<フィールド名>にファイルが指定されていません

原因：uploadFile タグの parameter 属性の値が正しくありません。

処置：uploadFile タグの parameter 属性の値を、マルチパート / フォームデータ・アップロード・リクエスト内の正しい名前に合せます。

メディア・データとメディア列が一致しません。

原因：メディア・データ・タイプとメディア列が一致しません。

処置：mediaParameters および mediaColumns 属性の値を確認します。

プレーヤ:<プレーヤ名>が無効です

原因：指定されたメディア・プレーヤが無効です。

処置：helperApp 属性の値を確認し、有効なメディア・プレーヤを指定します。

アップロードされたファイルの入力パラメータ <パラメータ値> が正しくありません

原因: mediaParameters 属性の値が正しくありません。

処置: mediaParameters 属性の値を確認します。

<列名> を取得するメディア列が正しくありません。

原因: メディア列タイプが、取得タグで指定されたタイプと一致しません。

処置: メディア列が正しいかどうか確認します。

構成ファイルが見つかりません。

原因: Multimedia Tag Library 構成ファイルが見つかりません。

処置: Multimedia Tag Library 構成ファイルが存在し、Web アプリケーションの web.xml ファイルと同じディレクトリに格納されていることを確認します。

アクセス・ユニット <アクセス・ユニット名> が見つかりません

原因: 指定されたアクセス・ユニット名が定義されていません。

処置: 構成ファイルで access-unit 属性を定義します。

データベース接続情報が指定されていません

原因: データベース接続情報が指定されていません。

処置: connectionCache または dataSourceName パラメータを指定します。

データベース接続を確立できません。

原因: 接続情報が指定されていないか、不正な接続情報が指定されています。

処置: 接続情報が正しいかどうか確認します。

keyColumn 名を取得できません。

原因: 主キーの列名を取得できません。

処置: 主キーが存在するかどうかを確認するか、column 属性を指定します。

interMedia オブジェクトの取得に失敗しました。

原因: データベースから interMedia オブジェクトを取得しようとしたときにエラーが発生しました。

処置: データベースにこのオブジェクトが存在することを確認します。

interMedia オブジェクトの更新に失敗しました。

原因: データベースの interMedia オブジェクトを更新しようとしたときにエラーが発生しました。

処置: オラクル社カスタマ・サポート・センターにご連絡ください。

列<列名>は *interMedia* オブジェクト・タイプではありません

原因: `mediaColumns` 属性により指定された列は、有効な *interMedia* オブジェクト・タイプではありません。

処置: `mediaColumns` 属性の値を確認します。

表名または列名が指定されていません。

原因: 表名またはメディア列名が指定されていません。

処置: 表名または列名を指定します。

行が見つかりません。

原因: 表内の行が見つかりません。

処置: `key` または `rowid` 属性の値が正しいかどうかを確認します。

列名と列値が一致していません。

原因: メディア列名とその値または他の列名とその値が一致していません。

処置: `mediaParameters`、`mediaColumns`、`otherColumns` および `otherValues` 属性の値を確認します。

構成ファイルの読取りでエラーが発生しました。

原因: Multimedia Tag Library 構成ファイルを読み取る際に、I/O エラーが発生しました。

処置: Multimedia Tag Library 構成ファイルが、付録 A に示したサンプル構成ファイル内の Document Type Definition (DTD) のルールに準拠した有効な XML ファイルであることを確認します。

索引

C

custom-retrieval-attributes, 3-5

D

database-connection-attributes, 3-6

E

embedAudio タグ, 3-15

embedImage タグ, 3-17

embedVideo タグ, 3-20

I

interMedia の概念, 1-2

J

J2EE, 1-4

J2SE (Java2 Standard Edition), 1-4

Java2 Enterprise Edition (J2EE), 1-4

Java2 Standard Edition (J2SE), 1-4

JavaServer Pages (JSP), 1-4

Java サブレット, 1-4, 1-5

Java の概念, 1-4

J2EE, 1-4

JavaServer Pages, 1-4

Java サブレット, 1-4

JSP タグ・ライブラリ, 1-5

Java プログラミング・ツール, 1-5

JSP (JavaServer Pages), 1-4

JSP タグ・ライブラリ, 1-5

JSP ページ, 1-5

M

media-access-attributes, 3-8

media-cache-control-attributes, 3-10

media-render-attributes, 3-14

mediaUrl タグ, 3-25

S

storeMedia タグ, 4-3

T

table-and-column-attributes, 3-12

TLD (タグ・ライブラリ・ディスクリプタ) ファイル,
A-2

U

uploadFile タグ, 4-6

uploadFormData タグ, 4-8

URL 書式

メディア取得タグ, 3-27

い

一般的な書式

メディア取得タグ, 3-4

インストール, A-2

TLD ファイルの指定, A-2

前提条件となる製品, A-2

え

エラー, B-1

か

概念

- interMedia*, 1-2
- JavaServer Pages, 1-4
- Java サブレット, 1-4
- Java プログラミング言語, 1-4
- JSP タグ・ライブラリ, 1-5
- メディアのアップロード, 1-3
- メディアの取得, 1-3

き

- キャッシュ制御属性, A-5

こ

構成

- キャッシュ制御属性の設定, A-5
- パラメータの指定, A-3
- メディア・アクセスの許可, A-4
- メディア配信コンポーネントの指定, A-3

構成パラメータ, A-3

- キャッシュ制御属性, A-5
- メディア・アクセス, A-4
- メディア配信コンポーネント, A-3

構成ファイル, A-5

さ

サンプル構成ファイル

- OrdJspTag.xml*, A-5

サンプル・プログラム

- mediaUrl* JSP タグ, 3-1
- タグ・ライブラリ構成ファイル, A-5
- メディアのアップロード, 2-1, 2-6
- メディアの取得, 2-1, 2-2

せ

- 前提条件となる製品, A-2

そ

属性

- custom-retrieval-attributes*, 3-5
- database-connection-attributes*, 3-6

- media-access-attributes*, 3-8
- media-cache-control-attributes*, 3-10
- media-render-attributes*, 3-14
- table-and-column-attributes*, 3-12

た

タグ

- embedAudio*, 3-15
- embedImage*, 3-17
- embedVideo*, 3-20
- mediaUrl*, 3-25
- storeMedia*, 4-3
- uploadFile*, 4-6
- uploadFormData*, 4-8
- メディアのアップロード, 1-3
- メディアの取得, 1-3
- タグ・ライブラリ・ディスクリプタ (TLD) ・ファイル, A-2

て

データベース接続情報

- メディアのアップロード, 4-1

ひ

表の情報

- メディアのアップロード, 4-1

ふ

- プログラミング・ツール, 1-5

ま

マルチメディア JSP タグ

- embedAudio*, 3-15
- embedImage*, 3-17
- embedVideo*, 3-20
- mediaUrl*, 3-25
- storeMedia*, 4-3
- uploadFile*, 4-6
- uploadFormData*, 4-8

め

- メディア・アクセス, A-4
 - 構成パラメータ, A-4
- メディア・アップロード・タグ
 - storeMedia, 4-3
 - uploadFile, 4-6
 - uploadFormData, 4-8
- メディア依存情報
 - メディアのアップロード, 4-1
- メディア取得タグ
 - custom-retrieval-attributes, 3-5
 - database-connection-attributes, 3-6
 - embedAudio, 3-15
 - embedImage, 3-17
 - embedVideo, 3-20
 - media-access-attributes, 3-8
 - media-cache-control-attributes, 3-10
 - media-render-attributes, 3-14
 - mediaUrl, 3-25
 - table-and-column-attributes, 3-12
 - URL 書式, 3-27
 - 一般的な書式, 3-4
- メディアのアップロード, 1-3
 - サンプル・コード, 2-6
 - 詳細の指定, 4-1
- メディアのアップロードの概念, 1-3
- メディアの取得, 1-3
 - HTML タグ, 3-1
 - サンプル・コード, 2-2
 - メディア・タイプ
 - イメージ, 3-2
 - オーディオ, 3-2
 - ビデオ, 3-2
 - メディア配信コンポーネント, 3-2
- メディアの取得の概念, 1-3
- メディア配信コンポーネント
 - 構成パラメータ, A-3

れ

- 例
 - oracle.ord.im.jsp.OrdGetMediaServlet, A-3
 - OrdGetMediaJsp.jsp, A-4
 - OrdJspTag.xml, A-5
 - PhotoAlbumInsertPhoto.jsp, 2-6
 - PhotoAlbum.jsp, 2-2

