
What's New in WebObjects 5.1

WebObjects 5.1

WebObjects 5.1 の新機能

Nouveautés de WebObjects 5.1

Neuerungen in WebObjects 5.1



January 2002

Apple Computer, Inc.
© 2002 Apple Computer, Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Computer, Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Computer, Inc. Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this book. Apple retains all intellectual property rights associated with the technology described in this book. This book is intended to assist application developers to develop applications only for Apple-labeled or Apple-licensed computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Computer, Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Apple, the Apple logo, Mac, Macintosh, and WebObjects are trademarks of Apple Computer, Inc., registered in the United States and other countries.

Enterprise Objects is a trademark of NeXT Software, Inc., registered in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Simultaneously published in the United States and Canada.

Even though Apple has reviewed this manual, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS MANUAL, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS MANUAL IS SOLD "AS IS," AND YOU, THE PURCHASER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS MANUAL, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

What's New in WebObjects 5.1



January 2002

Contents

What's New in WebObjects 5.1 3

Enterprise JavaBeans	3
JSP Servlet Integration	4
Java Client	5
Class Loader	5
JavaClientLauncher	6
WOAutoOpenClientApplication	6
Stateless Remote Method Invocations	7
Security Delegate Methods	7
Direct to Java Client Actions	7
DiscussionBoard	7
LDAP and JNDI Support	8
New and Improved Documentation	9
Javadoc	11

What's New in WebObjects 5.1

WebObjects 5.1, an incremental release to WebObjects 5.0, adds new features and functionality. It also includes updated and new documentation. This document covers the following additions and changes:

- **Enterprise JavaBeans** provides support for Enterprise JavaBeans (EJB).
- **JSP Servlet Integration** lets you deploy WebObject applications on top of third party JSP or Servlet application servers.
- **Java Client** offers enhancements and changes.
- **LDAP and JNDI** supports LDAP access through a JNDI adaptor.
- **New and improved documentation** offers several new and updated books and reference in Javadoc format.

Enterprise JavaBeans

WebObjects 5.1 is the first release of WebObjects to provide support for integration of Enterprise JavaBeans (EJB). EJB is an important part of J2EE and provides an environment in which components from several manufacturers can be assembled into a working application.

With WebObjects 5.1 you can use WebObjects development tools to develop enterprise beans from scratch or to integrate third-party beans in a WebObjects application. Bean development in WebObjects 5.1 is divided in two phases: development and deployment. You develop enterprise beans by writing the .java and deployment descriptor files (Project Builder provides you with templates for all these files). You can also obtain the source code or JAR files for enterprise beans

What's New in WebObjects 5.1

from a third party. You deploy one or more beans by generating a bean framework, which contains the beans' JAR and deployment descriptor files, and placing it in the `/Library/Frameworks` directory of your development host. After a bean framework has been deployed, it's available to be integrated in client applications for their use.

For more information on WebObjects and EJB, refer to *Inside WebObjects: Developing EJB Applications*, as explained in the New and Improved Documentation section of this document. There is framework specific documentation available in the JavaOpenEJB, JavaOpenTM, and JavaOpenORB frameworks. You can reference that documentation using Project Builder by opening up the frameworks area and looking at the Documentation area in the above mentioned frameworks.

JSP Servlet Integration

WebObjects 5.1 adds the ability for generic WebObjects applications to run as a servlet inside a servlet Container such as Tomcat or WebLogic. This means that instead of running as a separate JVM process, it runs inside the servlet container's JVM (along with, potentially, many other applications). Keep in mind that when you deploy a WebObjects application as a servlet, that none of the existing deployment tools (WebServer adaptors, wotaskd, Monitor) are applicable.

With WebObjects 5.1 JSP pages can now access WebObjects components directly. JSP pages that are part of the same servlet context (equivalent to a single application) use a provided tag library to generate HTML for WebObjects components. Note that to be part of the same context, the JSP pages must be part of the same WAR archive. There are two types of components that a JSP developer can use: WOCcomponents and DirectActions.

For more information on JSP Servlet Integration, see *Inside WebObjects: Developing WebObjects Applications with Java Server Pages and Servlets*, as explained in the New and Improved Documentation section of this document.

Java Client

Java Client in WebObjects 5.1 offers refinements and updates to WebObjects 5.0. Many bugs have been fixed and performance has been tuned in several areas. WebObjects 5.1 provides more documentation in the way of additional classes reference documentation and a new *WebObjects Desktop Applications* book. This book covers conceptual information and guides you through tutorials using both Java Client and Direct to Java Client. Several new, small features improve managing security issues and starting client applications.

Class Loader

For desktop applications, WebObjects 5.1 includes a special class loader feature that speeds development cycles and increases deployment flexibility. The class loader lets you download your own application's custom classes to a desktop application client, in addition to the standard Java Client classes that need to be pre installed on the client machine. With previous releases, all classes including your own custom classes, needed to be pre installed.

This is useful for both development and deployment, as you don't need to worry about classpath configurations and you gain flexibility with your client installations. With previous releases, you had to pre install the complete client application—now you only pre install the Java Client base system (without any of your own classes). If you need to upgrade your client application's classes, you don't need to send out an installer application—you simply update your classes on the server and all clients download them automatically the next time they start up. You only need to upgrade the client application when you change the version of WebObjects you're using.

You configure the class loader in the `WOJavaClientApplet` of your application with the new `downloadClientClasses` binding. See the class description of `WOJavaClientApplet` for more details. On Mac OS X, the Project Builder assistants for creating new projects ask you which class download option you want to use and configure the `WOJavaClientApplet` automatically for you. On Windows, the

What's New in WebObjects 5.1

assistants configure the `WOJavaClientApplet` to disable the class loader and you have to specify the correct value of the `downloadClientClasses` binding by hand in the component containing your `WOJavaClientApplet` (usually the Main component).

JavaClientLauncher

On Mac OS X, WebObjects 5.1 contains a new example, called `JavaClientLauncher`. `JavaClientLauncher` shows how to create a Mac OS X launch application that contains the Java Client base system and is configured to run a URL input dialog at startup. You simply place `JavaClientLauncher` in the dock, start it, enter the URL of any Java Client server application and connect right from the URL input dialog. As long as your server applications are configured to use the new class loader feature, `JavaClientLauncher` can run any client application (since it downloads your application specific classes at startup). The `JavaClientLauncher` example's Readme file contains instructions for setting up your own launch applications.

On Windows, WebObjects 5.1 contains a new script, called `JavaClientLauncherWin.cmd`, located in the WebObjects examples directory. It has the same functionality as the `JavaClientLauncher` application on Mac OS X and is documented. You can create a shortcut to the script on your desktop, if you wish.

WOAutoOpenClientApplication

On Mac OS X, WebObjects can create client launch scripts automatically during the build of a project. You enable or disable the client launch script generation in the build settings of your Web Server target in Project Builder. WebObjects 5.1 adds a new flag to the server side application: `WOAutoOpenClientApplication`. This flag's function is similar to `WOAutoOpenInBrowser`—instead of starting the client as an applet in the browser, it uses the client launch script to start a desktop application.

If `WOAutoOpenClientApplication` is set to Yes, the client is launched as a desktop application if the client launch script can be found. If not, the application falls back to the `WOAutoOpenInBrowser` functionality and launches the client as an applet. To avoid this, set both `WOAutoOpenClientApplication` and `WOAutoOpenInBrowser` to No.

On Windows, the client launch script generation and `WOAutoOpenClientApplication` is not available and you can only use `WOAutoOpenInBrowser`.

Stateless Remote Method Invocations

A new method, `invokeStatelessRemoteMethodWithKeyPath`, has been added to the `EODistributedObjectStore` class. It is a simpler and faster version of `invokeRemoteMethodWithKeyPath` and can be used for all communication that does not relate to enterprise objects.

You can use the new method to perform remote method invocations that affect the application logic, but not the business logic. For example, you can use it to load a resource. The method is also secure for use in background threads (as long as no enterprise objects are involved) and you can use the method safely to perform background thread checks with the server side.

Security Delegate Methods

Several new methods have been added to the `EODistributedObjects` store delegate. See the class description of `EODistributedObjects.Delegate` for more details. The new methods give you more control over what kind of data client applications are allowed to fetch and modify.

Direct to Java Client Actions

With Direct to Java Client, standard actions like Save or Undo were not configurable in earlier releases. In WebObjects 5.1, you can now configure the standard actions through the rule system. See the XML generated by the rule system for the actions question for more details (you can see the XML in the XML view of the Assistant).

DiscussionBoard

WebObjects 5.1 contains another new example: DiscussionBoard, an advanced example using Direct to Java Client to implement a discussion board application. The example comes with a long Readme file and is fairly well documented.

The example shows many aspects of Java Client, especially how to handle authentication (login and logout), security related to the authentication (administrator and non-administrator users have different kind of access permissions), user interface customization, background thread checks, and others. We highly recommend you take a look at this example if you are interested in Java Client.

LDAP and JNDI Support

WebObjects 4.5 shipped a Lightweight Directory Access Protocol (LDAP) adaptor as an example adaptor written in Objective-C. WebObjects 5.1 supports LDAP access through a Java Naming and Directory Interface (JNDI) adaptor—a fully-supported adaptor written in Java and compatible with the Java runtime of WebObjects. This allows WebObjects to connect to JNDI data sources from all supported WebObjects application architectures: Cocoa/EOF applications, Java Client applications, and HTML server applications.

In WebObjects 5.1 the JNDI adaptor includes the LDAP plug-in and supports the following servers:

- OpenLDAP LDAP Server
- iPlanet Directory Server

Note the following similarities and differences between JNDI adaptors and JDBC adaptors:

- Using EOModeler, you can create or reverse engineer a JNDI model using the same approach as you would for a JDBC model.
- Unlike the JDBC adaptor, however, the JNDI adaptor does not support schema synchronization. The JNDI adaptor /LDAP Plug-in can be used with LDAP servers that come with pre-defined schemas. Third-party tools or other mechanisms exist for modifying schemas.
- Every entity in a JNDI model has an attribute called the *relativeDistinguishedName* that acts as the primary key. The relative distinguished name represents a unique, human-readable identifier for an entry on a directory server.

As a convenience, the JNDI adaptor provides a mechanism to assist with the creation of the relative distinguished name for a new entry. To turn on automatic primary key generation, you only need to specify a pattern in the write format column of the *relativeDistinguishedName* attribute. The pattern may consist of any number of literals and attribute names embedded between less than (<) and greater than (>) signs.

What's New in WebObjects 5.1

For example, in the Person entity, the pattern `cn=<cn>` instructs the adaptor to append the literal `cn=` with the value of the attribute called `cn`. An application that creates a new person with Ernest Hemingway as the common name shows `cn=Ernest Hemingway` as the relative distinguished name.

In WebObjects 5.1 the LDAP plug-in doesn't support the use of key-comparison qualifiers, as LDAP doesn't support the attribute value comparison. Also, even though you can create a relationship between entities in a JNDI model and a JDBC model, Enterprise Objects currently doesn't support the flattening of attributes or relationships between entities in models that use different adaptors. For more information on other known issues, see *WebObjects 5.1 Release Notes* in `/Developer/Documentation/WebObjects/`, and online at <http://developer.apple.com/webobjects/>.

For information on downloading and installing the JNDI packages and LDAP service provider for your system, see the *Post-Installation Guide*. This document also covers basic operations for using LDAP and JNDI functionality with your system and includes some API information for this new feature.

New and Improved Documentation

WebObjects 5.1 offers several new documents:

- *Inside WebObjects: WebObjects Desktop Applications* provides comprehensive tutorials on developing applications using Java Client and Direct to Java Client. This document replaces both the *Java Client Tutorial* and *Getting Started with Direct to Java Client*.
- *Inside WebObjects: Developing EJB Applications* explains how to use WebObjects to develop EJB applications and provides some background information on EJB. This document assumes a previous knowledge of EJB.
- *Inside WebObjects: Developing Applications using Java Server Pages and Servlets* explains how to deploy WebObjects applications as servlets and how to use WebObjects components in JSP pages.

What's New in WebObjects 5.1

- *Post-Installation Guide* covers platform-specific information for the steps you need to take after installing WebObjects on your system. It explains how to install the JDBC and JNDI adaptors, how to set up your databases, and how to uninstall earlier versions of WebObjects. This document also includes information on testing your system.
- *Inside WebObjects: Developing SMIL Presentations* details how SMIL is implemented in WebObjects and teaches you how to create a SMIL presentation in WebObjects.

The following documents have been updated for WebObjects 5.1:

WebObjects Overview
Inside WebObjects: Deploying WebObjects Applications
Installation Guide for Mac OS X
Installation Guide for Windows and Solaris
WebObjects 5.1 Release Notes

In addition, the reference documentation for the WebObjects framework is now available in Javadoc format. See the following Javadoc section for more information and a complete listing of the Javadoc packages supplied for WebObjects 5.1.

You can access the following documents in the Documents folder of the WebObjects 5.1 product CD:

What's New in WebObjects 5.1
Installation Guide for Mac OS X
Installation Guide for Windows/Solaris
Post-Installation Guide
WebObjects Overview
Inside WebObjects: WebObjects Desktop Applications
Inside WebObjects: Deploying WebObjects Applications
Inside WebObjects: Developing EJB Applications
Inside WebObjects: Developing Applications with Java Server Pages and Servlets
Inside WebObjects: Discovering WebObjects for HTML

WebObjects Developer for Mac OS X installs these documents and others (including the Javadoc reference and the *WebObjects 5.1 Release Notes*) on your system when you install WebObjects Developer. These documents get installed in the `/Developer/Documentation/WebObjects/` directory.

What's New in WebObjects 5.1

To access the developer documentation on Windows 2000, use the WebObjects InfoCenter application by going to the Start menu and selecting Program > WebObjects > WOInfoCenter.

You can find a complete set of WebObjects documentation and resources at <http://developer.apple.com/webobjects/>. For this release, *Inside WebObjects: Developing SMIL Presentations* and the localized versions of the manuals shipped with the WebObjects 5.1 product are not available on the product CD, but are available at <http://developer.apple.com/webobjects/>.

To obtain printed copies of selected documents, go to <http://www1.fatbrain.com/documentation/apple/>.

Javadoc

This is the first release of the WebObjects reference documentation in Javadoc format. Javadoc brings many improvements to the reference:

- method signatures and class fields are accurate, as they are generated directly from the source code
- cross-referencing within classes, within packages, and across packages is more robust
- the output is provided in a more consumable format
- an index of all methods in all the public WebObjects packages is provided
- a use index documents where each package and class is used throughout WebObjects
- each class shows its superclass and the inheritance hierarchy back to `java.lang.Object`
- each class lists the methods it inherits from its superclasses and implemented interfaces

WebObjects 5.1 provides Javadoc for the following packages:

- WebObjects (`com.webobjects.appserver`)
- WebObjects XML (`com.webobjects.appserver.xml`)
- DirectToWeb (`com.webobjects.directtoweb`)
- EOAccess (`com.webobjects.eoaccess`)
- EOApplication (`com.webobjects.eoapplication`)

What's New in WebObjects 5.1

EOControl (com.webobjects.eocontrol)
EODistribution (com.webobjects.eodistribution)
EODistribution Client (com.webobjects.eodistribution.client)
EOGeneration (com.webobjects.eogeneration)
EOGeneration Client (com.webobjects.eogeneration.client)
EOInterface (com.webobjects.eointerface)
EOInterface Swing (com.webobjects.eointerface.swing)
EOInterface Cocoa (com.webobjects.eointerface.cocoa)
Foundation (com.webobjects.foundation)
JDBC Adaptor (com.webobjects.jdbcadaptor)
JDBC Adaptor (com.webobjects.jdbcadaptor)
JNDI Adaptor (com.webobjects.jndiadaptor)
WebObjects Extensions (com.webobjects.woextensions)

Most importantly, the content of the reference material has been reviewed by product engineers and contains many additions since WebObjects 5.0. Specifically, EOApplication, EODistribution, DirectToWeb, EOInterface, and JDBCAdaptor are more complete. The WebObjects framework, EOControl, and Foundation have also been filled out. And since the Javadoc comments live in the source code, method descriptions are updated when the source code changes.

WebObjects 5.1 の新機能



2002年 1月

目次

WebObjects 5.1 の新機能 3

Enterprise JavaBeans	3
JSP サブレットの統合	4
Java Client	5
クラスローダ	5
JavaClientLauncher	6
WOAutoOpenClientApplication	6
ステートレスなリモートメソッドの呼び出し	7
セキュリティ委譲メソッド	7
Direct to Java Client の動作	7
DiscussionBoard	7
LDAP と JNDI への対応	8
新しいマニュアルと改訂版のマニュアル	9
Javadoc	11

WebObjects 5.1 の新機能

「WebObjects 5.0」のインクリメンタルリリースである「WebObjects 5.1」はさまざまな新機能を追加します。また、「WebObjects 5.1」には改訂版のマニュアルと新しいマニュアルが用意されています。本書では、次の追加と変更について説明します：

- **Enterprise JavaBeans**。Enterprise JavaBeans (EJB) に対応しました。
- **JSPサーブレットの統合**。他社の JSP またはサーブレットアプリケーションサーバの最上位で「WebObjects」アプリケーションを運用できるようになりました。
- **Java Client**。さまざまな拡張と変更が追加されます。
- **LDAPとJNDIへの対応**。JNDI アダプタを介したLDAP アクセスに対応しました。
- **新しいマニュアルと改訂版のマニュアル**。いくつかの新しい書籍と書籍の改訂版、そして Javadoc 形式のリファレンスが提供されます。

Enterprise JavaBeans

「WebObjects 5.1」は、Enterprise JavaBeans (EJB) の統合に対応した「WebObjects」の最初のリリースです。EJBはJ2EE の重要部分で、いくつかの製造元が開発した複数のコンポーネントを 1 つの作業アプリケーションとして組み立てることのできる環境を提供します。

「WebObjects 5.1」を使用すると、「WebObjects」の開発ツールを使って、エンタープライズ Bean をまったく新規に開発したり、他社製の Bean を「WebObjects」アプリケーションに統合したりできます。「WebObjects 5.1」による Bean 開発は 2 つの段階に分けられます：開発と運用という 2 つの段階です。エンタープライズ Bean は、.java ファイルと運用記述子ファイルを書いて開発します（「Project Builder」には、これらすべてのファイルのテンプレートが用意されています）。また、他社からエンタープライズ Bean のソースコードや JAR ファイルを入手することもできます。1 つまたは複数の Bean を運用するには、Bean の JAR ファイルと運用記述子ファイルを含む Bean フレームワークを生成し、

WebObjects 5.1 の新機能

それを運用ホストの /Library/Frameworks ディレクトリに保存します。このようにして Bean フレームワークを運用すると、クライアントアプリケーションに統合して使用することができるようになります。

「WebObjects」と EJB の詳細については、本書の「新しいマニュアルと改訂版のマニュアル」のセクションの説明に従って、「Inside WebObjects: Developing EJB Applications」を参照してください。JavaOpenEJB、JavaOpenTM、JavaOpenORBの各フレームワークには、フレームワーク固有のマニュアルが用意されています。

「Project Builder」でこれらのマニュアルを参照するには、フレームワークが保存されている場所を開いて、それぞれのフレームワークのマニュアルが保存されているフォルダを探してください。

JSPサーブレットの統合

「WebObjects 5.1」には、ジェネリックな「WebObjects」アプリケーションを「Tomcat」や「WebLogic」などのサーブレットコンテナの内部でサーブレットとして実行する機能が追加されました。これにより、「WebObjects」アプリケーションを個別の JVM プロセスとして実行するのではなく、サーブレットコンテナの JVM の内部で実行できるようになります（ほかの多くのアプリケーションと共に）。ただし、「WebObjects」アプリケーションをサーブレットとして運用するときには、既存の運用ツール（Webサーバアダプタ、「wotaskd」、「Monitor」）を適用できないという点に注意してください。

「WebObjects 5.1」の JSP ページは、「WebObjects」コンポーネントに直接アクセスできるようになりました。同じサーブレットコンテキスト（1つのアプリケーションに相当します）の一部である JSP ページは、あらかじめ用意されているタグライブラリを使って、「WebObjects」コンポーネントの HTML を生成します。なお、同じコンテキストの一部となるためには、JSP ページが同じ WAR アーカイブの一部になっている必要があります。JSP デベロッパが使用できるコンポーネントには 2 つのタイプがあります：WOComponents と DirectActions の 2 つです。

JSP サーブレットの統合の詳細については、本書の「新しいマニュアルと改訂版のマニュアル」のセクションの説明に従って、「Inside WebObjects: Developing WebObjects Applications with Java Server Pages and Servlets」を参照してください。

Java Client

「WebObjects 5.1」の Java Client は、「WebObjects 5.0」の Java Client にさまざまな拡張やアップデートを提供します。多くのバグが修正され、いくつかの領域ではパフォーマンスの向上も実現されています。「WebObjects 5.1」ではマニュアルがより充実し、クラス参照のマニュアルや新しい「WebObjects Desktop Applications」という書籍が追加されています。この書籍では概念的な情報を取り扱い、Java Client と Direct to Java Client を使用して、段階的に読み進むことのできるチュートリアルを提供します。さらに、セキュリティ問題の管理やクライアントアプリケーションの起動を向上させる新機能も追加されています。

クラスローダ

デスクトップアプリケーションのため、「WebObjects 5.1」には、開発サイクルをスピードアップし、運用時の柔軟性を向上させる特殊なクラスローダ機能が用意されています。このクラスローダを使用すると、クライアントコンピュータにあらかじめインストールしておく必要のある標準 Java Client クラスに加えて、アプリケーション独自のカスタムクラスをデスクトップアプリケーションクライアントにダウンロードできるようになります。以前のリリースでは、独自のカスタムクラスを含むすべてのクラスをあらかじめインストールしておく必要がありました。

クラスパスの設定を考慮する必要がなく、またクライアントのインストールをより柔軟に行うことができるため、この機能は開発と運用の両方に役立ちます。以前のリリースでは完全なクライアントアプリケーションをあらかじめインストールしておく必要がありましたが、今回のリリースからは Java Client のベースシステム（独自のクラスを含まない）だけをあらかじめインストールすればよくなりました。クライアントアプリケーションのクラスをアップグレードする必要がある場合、インストーラアプリケーションを送る必要はありません。サーバ上のクラスをアップデートするだけで、すべてのクライアントは次に起動するときにアップデートされたクラスを自動的にダウンロードします。必要なのは、使用する「WebObjects」のバージョンを変更したときに、クライアントアプリケーションをアップグレードすることだけです。

クラスローダの設定は、アプリケーションの `WOJavaClientApplet` の中で、新しい `downloadClientClasses` バインディングを使って行います。詳細については、`WOJavaClientApplet` クラスの記述を参照してください。「Mac OS X」では、新しいプロジェクトを作成する「Project Builder」のアシスタントが使用するクラスダウンロードオプションを問い合わせ、ユーザの指示に従って、`WOJavaClientApplet` を自動的に設定します。「Windows」では、アシスタントが `WOJavaClientApplet` を設定してクラスローダ

WebObjects 5.1 の新機能

を無効にするため、WOJavaClientAppletを含むコンポーネント（通常はMainコンポーネント）の中で、downloadClientClassesバインドイングの正しい値を手入力で指定する必要があります。

JavaClientLauncher

「Mac OS X」では、「WebObjects 5.1」に JavaClientLauncher という名前の新しいサンプルが含まれています。JavaClientLauncher では、Java Client のベースシステムを含み、起動時に URL 入力ダイアログを実行するように設定されている「Mac OS X」用の起動アプリケーションを作成する方法を示しています。JavaClientLauncher を Dock に置き、JavaClientLauncher を起動して、任意の Java Client サーバアプリケーションの URL を入力するだけで、URL 入力ダイアログから直接サーバに接続できます。サーバアプリケーションが新しいクラスローダ機能を使用するように設定されている限り、JavaClientLauncher は任意のクライアントアプリケーションを実行できます（アプリケーション固有のクラスが起動時にダウンロードされるため）。JavaClientLauncher サンプルの Readme ファイルには、独自の起動アプリケーションを設定する手順が示されています。

「Windows」では、「WebObjects 5.1」のサンプルディレクトリに JavaClientLauncherWin.cmd という名前の新しいスクリプトが含まれています。このスクリプトは、「Mac OS X」の JavaClientLauncher アプリケーションと同じ機能を提供し、マニュアル化されています。必要な場合は、このスクリプトへのショートカットをデスクトップに作成できます。

WOAutoOpenClientApplication

「Mac OS X」の場合、「WebObjects」はプロジェクトのビルド中にクライアント起動スクリプトを自動的に作成できます。クライアント起動スクリプトの生成を行うかどうかは、「Project Builder」で Web サーバターゲットのビルド設定を使って切り替えます。「WebObjects 5.1」にはサーバ側アプリケーションに対する新しいフラグが追加されています：このフラグは WOAutoOpenClientApplication です。このフラグの機能は、クライアントをブラウザ内のアプレットとして起動する WOAutoOpenInBrowser に似ていますが、このフラグではクライアント起動スクリプトを使ってデスクトップアプリケーションを起動する点が異なります。

WOAutoOpenClientApplication が Yes に設定されていると、クライアント起動スクリプトが検出された場合に、クライアントはデスクトップアプリケーションとして起動されます。そうでない場合、アプリケーションは WOAutoOpenInBrowser の機能にさかのぼり、クライアントをアプレットとして起動します。このような動作を避けるには、WOAutoOpenClientApplication と WOAutoOpenInBrowser の両方に No を設定します。

WebObjects 5.1 の新機能

「Windows」の場合、クライアント起動スクリプトの生成と W0AutoOpenClientApplication は使用できず、W0AutoOpenInBrowser の使用のみ可能です。

ステートレスなリモートメソッドの呼び出し

新しいメソッドの `invokeStatelessRemoteMethodWithKeyPath` が `EODistributedObjectStore` クラスに追加されました。これは、`invokeRemoteMethodWithKeyPath` のより単純で、より高速に動作するバージョンで、エンタープライズオブジェクトに関連しないすべてのデータのやり取りに使用できます。

このメソッドを使用すると、アプリケーションロジックには影響し、ビジネスロジックには影響しないリモートメソッドの呼び出しを実行できます。たとえば、このメソッドを使用してリソースを読み込むことができます。また、このメソッドはバックグラウンドスレッドとしても安全に使用でき（エンタープライズオブジェクトが関係していない限り）、このメソッドを使って、サーバ側でバックグラウンドスレッドのチェックを安全に実行できます。

セキュリティ委譲メソッド

いくつかの新しいメソッドが `EODistributedObjects` ストア委譲に追加されました。詳細については、`EODistributedObjects.Delegate` クラスの記述を参照してください。これらの新しいメソッドにより、フェッチと変更が許可されるデータクライアントアプリケーションの種類をより詳細に制御できるようになります。

Direct to Java Client の動作

Direct to Java Client の場合、以前のリリースでは「保存」や「取り消し」などの標準的な動作を設定できませんでした。「WebObjects 5.1」では、規則システムを介して標準的な動作を設定できるようになりました。詳細については、動作の問題に対する規則システムによって生成される XML を参照してください（この XML はアシスタントの XML 表示で確認できます）。

DiscussionBoard

「WebObjects 5.1」にはもう一つの新しいサンプルが含まれています：Direct to Java Client を使用した高度なサンプルで、ディスカッションボードアプリケーションを実装する DiscussionBoard です。このサンプルには長い README ファイルが付属していて、かなり詳細にマニュアル化されています。

WebObjects 5.1 の新機能

このサンプルは、特に、認証（ログインとログアウト）、認証関連のセキュリティ（管理者ユーザと非管理者ユーザは異なる種類のアクセス権を持ちます）、ユーザインタフェースのカスタマイズ、バックグラウンドスレッドのチェックなどを処理する方法を中心に、Java Client のさまざまな側面を具体的に示しています。Java Client に関心がある場合は、このサンプルを参照することを強くお勧めします。

LDAP と JNDI への対応

「WebObjects 4.5」には、Objective-Cで書かれたサンプルアダプタとして、LDAP (Lightweight Directory Access Protocol) アダプタが付属していました。「WebObjects 5.1」は、JNDI (Java Naming and Directory Interface) アダプタを介した LDAP アクセスに対応しました。この JNDI アダプタは Java で書かれたフルサポートアダプタで、「WebObjects」の Java ランタイムと互換性があります。これにより、「WebObjects」は対応するすべての「WebObjects」アプリケーションアーキテクチャ (Cocoa/EOFアプリケーション、Java Clientアプリケーション、およびHTMLサーバアプリケーション) から JNDIデータソースに接続できるようになります。

「WebObjects 5.1」の JNDI アダプタは LDAP プラグインを含み、次のサーバに対応しています：

- OpenLDAP LDAPサーバ
- iPlanetディレクトリサーバ

なお、JNDIアダプタとJDBCアダプタの間には次の類似点と相違点があります：

- 「EOModeler」を使用すると、JDBC モデルと同じ方法で、JNDI モデルの作成とリバースエンジニアリングを行うことができます。
- ただし、JDBC アダプタとは異なり、JNDI アダプタはスキーマの同期に対応していません。JNDI アダプタと LDAP プラグインは、あらかじめ定義されているスキーマが付属する LDAP サーバと共に使用できます。スキーマを変更するため、いくつかの他社製ツールやその他のメカニズムが用意されています。
- JNDI モデルに含まれるすべてのエンティティは、relativeDistinguishedName と呼ばれ、主キーとして動作する属性があります。相対識別名は、ディレクトリサーバ上のエントリに対する重複のない表示用の識別子を表します。

便宜上、JNDI アダプタには、新しいエントリに対する相対識別名の作成を支援するメカニズムが用意されています。自動的な主キーの生成を入にするには、relativeDistinguishedName 属性の書き込みフォーマットの列にパターンを指定するだけです。このパターンは、任意の数のリテラルと不等号 (< と >) に囲まれた属性名から構成されます。

WebObjects 5.1 の新機能

たとえば、Person エンティティの場合、`cn=<cn>` というパターンは、`cn` という名前の属性の値を含むリテラル `cn=` を追加することをアダプタに命令します。一般名として Ernest Hemingway を含む新しい人物を作成するアプリケーションは、相対識別名として `cn=Ernest Hemingway` を表示します。

「WebObjects 5.1」の LDAP プラグインは、LDAP が属性値の比較に対応していないため、キー比較修飾子の使用に対応していません。また、JNDIモデルのエンティティと JDBC モデルのエンティティの間にリレーションシップを確立することはできませんが、Enterprise Objects は現在のところ、異なるアダプタを使用するモデムのエンティティ間での属性またはリレーションシップのフラット化には対応していません。その他のこれまでに報告されている問題の詳細については、「WebObjects 5.1 Release Notes」を参照してください。このマニュアルは /Developer/Documentation/WebObjects/ ディレクトリにあります。また、オンラインで <http://developer.apple.com/webobjects/> から入手できます。

使用中のシステムに対応した JNDI パッケージと LDAP サービスプロバイダをダウンロードしてインストールする方法については、「Post-Installation Guide」を参照してください。また、このマニュアルでは、システムで LDAP および JNDI 機能を使用するための基本操作と、この新機能に対応したいくつかの API に関する情報も説明されています。

新しいマニュアルと改訂版のマニュアル

「WebObjects 5.1」にはいくつかの新しいマニュアルが追加されました：

- 「Inside WebObjects: WebObjects Desktop Applications」には、Java Client と Direct to Java Client を使用してアプリケーションを開発するための総合的なチュートリアルが用意されています。このマニュアルは、これまでの「Java Client Tutorial」と「Getting Started with Direct to Java Client」に代わるものです。
- 「Inside WebObjects: Developing EJB Applications」では、「WebObjects」を使って EJB アプリケーションを開発する方法が説明されています。また、EJB の背景となる情報も示されています。このマニュアルを読むためには、EJB に関するある程度の予備知識が必要です。
- 「Inside WebObjects: Developing Applications using Java Server Pages and Servlets」では、「WebObjects」アプリケーションをサーブレットとして運用する方法と、「WebObjects」コンポーネントを JSP ページ内で使用する方法が説明されています。
- 「Post-Installation Guide」では、「WebObjects」を使用中のシステムにインストールした後で実行する必要があるプラットフォーム固有の操作手順が説明されています。また、JDBC および JNDI アダプタをインストールする方法、データベースを設定する

WebObjects 5.1 の新機能

方法、および「WebObjects」の以前のバージョンをアンインストールする方法も説明されています。さらに、このマニュアルには、システムのテストに関する情報も含まれています。

- 「Inside WebObjects: Developing SMIL Presentations」では、「WebObjects」に SMIL を実装する方法が詳細に説明されています。また、「WebObjects」で SMIL プレゼンテーションを作成するためのチュートリアルも用意されています。

次のマニュアルは、「WebObjects 5.1」の新機能に対応して改訂されました：

「WebObjects の概要」
「Inside WebObjects: Deploying WebObjects Applications」
「Mac OS X 版インストールガイド」
「Windows 版／ Solaris 版インストールガイド」
「WebObjects 5.1 Release Notes」

また、「WebObjects」フレームワークのリファレンスマニュアルを Javadoc 形式で利用できるようになりました。詳細と「WebObjects 5.1」のために用意されている Javadoc パッケージの一覧については、以下の Javadoc のセクションを参照してください。

「WebObjects 5.1」製品CDの「Documents」フォルダには次のマニュアルが用意されています：

「What's New in WebObjects 5.1」（本書の英語版）
「Installation Guide for Mac OS X」
「Installation Guide for Windows and Solaris」
「Post-Installation Guide」
「WebObjects Overview」
「Inside WebObjects: WebObjects Desktop Applications」
「Inside WebObjects: Deploying WebObjects Applications」
「Inside WebObjects: Developing EJB Applications」
「Inside WebObjects: Developing Applications with Java Server Pages and Servlets」
「Inside WebObjects: Discovering WebObjects for HTML」

「WebObjects Developer」をインストールするとき、「WebObjects Developer for Mac OS X」はこれらのマニュアルとその他のマニュアル（Javadoc リファレンスと「WebObjects 5.1 Release Notes」を含む）をシステムにインストールします。これらのマニュアルは、/Developer/Documentation/WebObjects/ ディレクトリにインストールされます。

「Windows 2000」でデベロッパマニュアルにアクセスするには、「WebObjects InfoCenter」アプリケーションを使用します。このアプリケーションを起動するには、「スタート」ボタンをクリックし、「プログラム」、「WebObjects」を順にポイントして、「WOInfoCenter」をクリックします。

WebObjects 5.1 の新機能

「WebObjects」のマニュアルとリソースの完全なセットは、<http://developer.apple.com/webobjects/> に用意されています。今回のリリースの場合、「WebObjects 5.1」製品に付属する「Inside WebObjects: Developing SML Presentations」とそのローカライズバージョンは製品 CD に収録されていませんが、<http://developer.apple.com/webobjects/> で入手できます。

選択したマニュアルの印刷バージョンが必要な場合は、<http://www1.fatbrain.com/documentation/apple/> にアクセスしてください。

Javadoc

これは、Javadoc 形式で提供される「WebObjects」のリファレンスマニュアルの最初のリリースです。Javadoc はリファレンスに多くの改善をもたらします：

- ソースコードから直接生成されるため、メソッドのシグネチャとクラスのフィールドが正確になります。
- クラス内、パッケージ内、複数のパッケージ間の相互参照がより豊富になります。
- 出力がより使いやすい形式で提供されます。
- すべての public WebObjects パッケージに含まれるすべてのメソッドの索引が用意されています。
- 索引の使用により、各パッケージとクラスが「WebObjects」内で使用されている場所がマニュアル化されます。
- それぞれのクラスはそのスーパークラスと `java.lang.Object` までさかのぼる継承の階層を示します。
- それぞれのクラスはそのスーパークラスと実装されたインタフェースから継承するメソッドの一覧を表示します。

「WebObjects 5.1」には、次のパッケージに対する Javadoc が用意されています：

```
WebObjects (com.webobjects.appserver)
WebObjects XML (com.webobjects.appserver.xml)
DirectToWeb (com.webobjects.directtoweb)
EOAccess (com.webobjects.eoaccess)
EOApplication (com.webobjects.eoapplication)
EOControl (com.webobjects.eocontrol)
EODistribution (com.webobjects.eodistribution)
EODistribution Client (com.webobjects.eodistribution.client)
EOGeneration (com.webobjects.eogeneration)
EOGeneration Client (com.webobjects.eogeneration.client)
```

WebObjects 5.1 の新機能

EOInterface (com.webobjects.eointerface)
EOInterface Swing (com.webobjects.eointerface.swing)
EOInterface Cocoa (com.webobjects.eointerface.cocoa)
Foundation (com.webobjects.foundation)
JDBC Adaptor (com.webobjects.jdbcadaptor)
JDBC Adaptor (com.webobjects.jdbcadaptor)
JNDI Adaptor (com.webobjects.jndiadaptor)
WebObjects Extensions (com.webobjects.woextensions)

最も重要なことは、リファレンス資料の内容がプロダクトエンジニアによって校閲されていることと、「WebObjects 5.0」以降に加えられた多くの追加が記載されていることです。特に、EOApplication、EODistribution、DirectToWeb、EOInterface、および JDBCAdaptor の記述がより完全なものになっています。「WebObjects」フレームワーク、EOControl、および Foundation の記述も充実しました。そして、Javadoc のコメントはソースコードにリンクしているため、ソースコードが変更される時、メソッドの記述も自動的にアップデートされます。

Nouveautés de WebObjects 5.1



Janvier 2002

Table des matières

Nouveautés de WebObjects 5.1 3

Enterprise JavaBeans	3
JSP Servlet Integration	4
Java Client	5
Chargeur de classes	5
JavaClientLauncher	6
WOAutoOpenClientApplication	6
Appels de méthodes à distance sans état	7
Méthodes de délégation de sécurité	7
Actions Direct to Java Client	7
DiscussionBoard	8
Gestion LDAP et JNDI	8
Nouvelle documentation améliorée	10
Javadoc	12

Nouveautés de WebObjects 5.1

WebObjects 5.1, une mise à jour de WebObjects 5.0, comporte de nouvelles fonctions et fonctionnalités. Elle inclut également une nouvelle documentation mise à jour. Ce document traite des nouveautés et modifications suivantes :

- **Enterprise JavaBeans** fournit la gestion des composants Enterprise JavaBeans (EJB).
- **JSP Servlet Integration** vous permet de déployer des applications WebObject sur des serveurs d'applications JSP ou Servlet de tierce partie.
- **Java Client** offre améliorations et modifications.
- **LDAP et JNDI** gère l'accès LDAP via un adaptateur JNDI.
- **Nouvelle documentation améliorée** comporte plusieurs nouveaux manuels mis à jour et des références au format Javadoc.

Enterprise JavaBeans

WebObjects 5.1 est la première version de WebObjects à gérer l'intégration de composants Enterprise JavaBeans (EJB). Ces composants constituent une partie importante de J2EE et fournissent un environnement dans lequel les composants de divers fabricants peuvent être assemblés pour former une application qui fonctionne.

Avec WebObjects 5.1, vous pouvez utiliser les outils de développement WebObjects pour développer de nouveaux beans entreprise ou pour intégrer des beans de tierce partie dans une application WebObjects. Le développement de beans dans WebObjects 5.1 se fait en deux étapes : le développement et le déploiement. Les beans entreprise se développent en écrivant des fichiers de description de déploiement

Nouveautés de WebObjects 5.1

et .java (Project Builder fournit des modèles de tous ces fichiers). Vous pouvez également obtenir les codes source ou fichiers JAR des beans entreprise d'un autre fabricant. Les beans se déploient en générant une structure de beans contenant leurs fichiers JAR et de description de déploiement, et en plaçant cette dernière dans le répertoire `/Library/Frameworks` de l'hôte de développement. Une fois une structure de beans déployée, elle est prête à être intégrée dans des applications client.

Pour plus d'informations sur WebObjects et EJB, consultez *Inside WebObjects: Developing EJB Applications*, comme expliqué dans la section Nouvelle documentation améliorée de ce document. Les structures JavaOpenEJB, JavaOpenTM, et JavaOpenORB contiennent de la documentation spécifique aux structures. Vous pouvez utiliser comme référence cette documentation à l'aide de Project Builder, en ouvrant les structures et en consultant la zone de documentation des structures mentionnées ci-dessus.

JSP Servlet Integration

WebObjects 5.1 permet aux applications génériques WebObjects d'être exécutées en tant que servlet dans un conteneur de servlet Tomcat ou WebLogic. Ceci signifie qu'elles ne s'exécutent pas comme processus JVM séparé, mais dans la JVM du conteneur de servlet (avec, potentiellement, beaucoup d'autres applications). Rappelez-vous que lorsque vous déployez une application WebObjects en tant que servlet, aucun des outils de déploiements existants (adaptateurs WebServer, wotaskd, Monitor) ne peuvent être utilisés.

Grâce à WebObjects 5.1, les pages JSP peuvent désormais accéder aux composants WebObjects directement. Les pages JSP appartenant à un même contexte de servlet (équivalent à une seule application) emploient une bibliothèque de balises qui leur est fournie pour générer du code HTML pour les composants WebObjects. Remarquez que pour faire partie d'un même contexte, les pages JSP doivent appartenir au même fichier WAR. Les développeurs JSP peuvent utiliser deux types de composants : WOComponents et DirectActions.

Pour plus d'informations sur JSP Servlet Integration, consultez *Inside WebObjects: Developing WebObjects Applications with Java Server Pages and Servlets*, comme expliqué dans la section Nouvelle documentation améliorée de ce document.

Java Client

Dans WebObjects 5.1, Java Client offre des perfectionnements et des mises à jours de WebObjects 5.0. De nombreux bogues ont été éliminés et les performances ont été améliorées dans plusieurs domaines. WebObjects 5.1 fournit une documentation plus fournie quant aux références de classes supplémentaires ainsi qu'un nouveau manuel *WebObjects Desktop Applications*. Ce manuel contient des informations conceptuelles et vous guide au cours des initiations à Java Client et Direct to Java Client. De nouvelles petites fonctionnalités améliorent la gestion des problèmes de sécurité et le démarrage d'applications client.

Chargeur de classes

Pour les applications de bureau, WebObjects 5.1 comprend un chargeur de classes spécial qui accélère les cycles de développement et augmente la flexibilité de déploiement. Le chargeur de classes vous permet de télécharger vos propres classes personnalisées sur une application de bureau ainsi que les classes Java Client standard qui nécessitent une préinstallation sur l'ordinateur client. Avec les versions précédentes, toutes les classes, y compris les classes personnalisées, devaient être préinstallées.

Ce chargeur de classes est utile aussi bien pour le développement que pour le déploiement puisqu'il vous évite la configuration de chemins de classe et que vous gagnez en flexibilité avec vos installations client. Avec les versions antérieures, il fallait préinstaller entièrement l'application client. Désormais, il vous suffit de préinstaller le système de base Java Client (sans aucune de vos classes). Si vous devez mettre à niveau les classes de votre application client, il est désormais inutile de lancer une application d'installation. En effet, il vous suffit de mettre à jour vos classes sur le serveur et tous les clients les téléchargent automatiquement lors de leur prochain démarrage. L'application client ne doit être mise à niveau que lorsque vous souhaitez changer de version WebObjects.

Le chargeur de classes du `WOJavaClientApplet` de votre application se configure à l'aide de la nouvelle liaison `downloadClientClasses`. Pour plus de détails, consultez la description de classe de `WOJavaClientApplet`. Sous Mac OS X, les assistants Project Builder de création de nouveaux projets vous demandent quelle est l'option de

Nouveautés de WebObjects 5.1

téléchargement de classe souhaitée et configurent automatiquement le `WOJavaClientApplet` pour vous. Sous Windows, les assistants configurent le `WOJavaClientApplet` pour désactiver le chargeur de classes et il vous revient de saisir à la main la valeur de la liaison `downloadClientClasses` qui convient dans le composant contenant votre `WOJavaClientApplet` (habituellement le composant principal).

JavaClientLauncher

Sous Mac OS X, WebObjects 5.1 comporte un nouvel exemple intitulé `JavaClientLauncher`. `JavaClientLauncher` indique comment créer une application de lancement Mac OS X contenant le système de base Java Client et est configuré pour afficher une zone de dialogue de saisie d'URL au démarrage. Il suffit de placer `JavaClientLauncher` dans le Dock, de le lancer, de saisir l'URL de n'importe quelle application de serveur Java Client et de se connecter directement à partir de la zone de dialogue de saisie de l'URL. Si vos applications de serveur sont configurées pour utiliser la nouvelle fonction de chargeur de classes, `JavaClientLauncher` peut exécuter tous les types d'applications client (à partir du moment où il télécharge les classes propres à votre application au démarrage). Le fichier `Readme` de `JavaClientLauncher` contient les instructions de configuration de vos applications de lancement.

Sous Windows, WebObjects 5.1 contient un nouveau script, intitulé `JavaClientLauncherWin.cmd`, situé dans le répertoire des exemples WebObjects. Il possède la même fonctionnalité que l'application `JavaClientLauncher` sous Mac OS X et inclut une documentation. Si vous le souhaitez, vous pouvez créer un raccourci de ce script sur votre bureau.

WOAutoOpenClientApplication

Sous Mac OS X, WebObjects peut créer automatiquement des scripts de lancement client au cours de la création d'un projet. La génération de scripts de lancement client s'active et se désactive via les réglages de création de votre cible Web Server dans Project Builder. WebObjects 5.1 ajoute un nouvel indicateur à l'application côté serveur : `WOAutoOpenClientApplication`. La fonction de cet indicateur est similaire à celle de `WOAutoOpenInBrowser` : au lieu de démarrer le client en tant qu'applet dans le navigateur, il utilise le script de lancement client pour démarrer les applications de bureau.

Nouveautés de WebObjects 5.1

Si `WOAutoOpenClientApplication` est réglé sur `Yes`, le client est lancé en tant qu'application de bureau si son script de lancement est détectable. Dans le cas contraire, l'application revient à la fonctionnalité `WOAutoOpenInBrowser` et lance le client en tant qu'applet. Pour éviter cela, réglez `WOAutoOpenClientApplication` et `WOAutoOpenInBrowser` sur `No`.

Sous Windows, la génération de scripts de lancement client et `WOAutoOpenClientApplication` ne sont pas disponibles et seul `WOAutoOpenInBrowser` peut être utilisé.

Appels de méthodes à distance sans état

Une nouvelle méthode, `invokeStatelessRemoteMethodWithKeyPath`, a été ajoutée à la classe `EODistributedObjectStore`. Il s'agit d'une version simplifiée et plus rapide de `invokeRemoteMethodWithKeyPath`, qui peut être utilisée pour toutes les communications n'ayant pas trait aux objets d'entreprise.

Vous pouvez employer cette nouvelle méthode pour réaliser des appels de méthodes à distance affectant la logique de l'application mais pas la logique métier. Par exemple, vous pouvez l'employer pour charger une ressource. Cette méthode, qui est également sûre pour une utilisation dans les tâches en arrière-plan (si aucun objet d'entreprise n'est impliqué), peut être utilisée sans risque pour réaliser des contrôles de tâches en arrière-plan côté serveur.

Méthodes de délégation de sécurité

Plusieurs méthodes ont été ajoutées au délégué `EODistributedObjects`. Consultez la description de classe de `EODistributedObjects.Delegate` pour plus de détails. Les nouvelles méthodes vous donnent un meilleur contrôle du type de données que les applications client sont autorisées à télécharger et à modifier.

Actions Direct to Java Client

Dans les versions ultérieures, les actions standard Direct to Java Client telles que `Save` ou `Undo` ne pouvaient être configurées. Avec WebObjects 5.1, vous pouvez désormais configurer ces actions standard dans le système de règles. Pour plus de détails sur ces actions, consultez le document XML (dans la présentation XML de l'Assistant) généré par le système de règles.

DiscussionBoard

WebObjects 5.1 contient un autre nouvel exemple : DiscussionBoard, un exemple avancé utilisant Direct to Java Client pour exécuter une application de forum de discussion. L'exemple contient un long fichier Readme et est très bien documenté.

Il montre de nombreux aspects de Java Client, particulièrement la gestion de l'authentification (ouverture et fermeture de session), la sécurité liée à l'authentification (les utilisateurs administrateurs et non-administrateurs ont des autorisations d'accès différentes), la personnalisation de l'interface utilisateur, le contrôle des tâches en arrière-plan, etc. Si vous êtes intéressé par Java Client, nous vous recommandons vivement de consulter cet exemple.

Gestion LDAP et JNDI

WebObjects 4.5 fournissait un adaptateur LDAP (Lightweight Directory Access Protocol) comme exemple d'adaptateur écrit en Objective-C. WebObjects 5.1 gère l'accès LDAP via un adaptateur JNDI (Java Naming and Directory Interface), un adaptateur complètement géré, écrit en langage Java et compatible avec le moteur d'exécution Java de WebObjects. Ceci permet à WebObjects de se connecter aux sources de données JNDI à partir de toutes les architectures d'application WebObjects gérées : applications Cocoa/EOF, Java Client, et de serveur HTML.

Avec WebObjects 5.1, l'adaptateur JNDI inclut le module LDAP et gère les serveurs suivants :

- Serveur LDAP OpenLDAP
- Serveur iPlanet

Prenez note des similarités et différences suivantes entre les adaptateurs JNDI et JDBC :

- Avec EOModeler, vous pouvez créer ou procéder à l'ingénierie inverse d'un modèle JNDI à l'aide de la même approche que celle à employer pour un modèle JDBC.

Nouveautés de WebObjects 5.1

- Toutefois, à la différence de l'adaptateur JDBC, l'adaptateur JNDI ne gère pas la synchronisation de schémas. Le module adaptateur JNDI/LDAP peut être utilisé avec les serveurs LDAP dotés de schémas prédéfinis. Des outils et autres mécanismes de tierce partie existent pour la modification de schémas.
- Dans un modèle JNDI, chaque entité possède un attribut appelé *relativeDistinguishedName*, qui agit comme clé principale. Le nom distinctif relatif constitue un identifiant unique et lisible par l'utilisateur d'une entrée sur un serveur de répertoires.

Par commodité, l'adaptateur JNDI fournit un mécanisme d'assistance à la création du nom distinctif relatif d'une nouvelle entrée. Pour activer la génération automatique de clé principale, il vous suffit de spécifier un filtre dans la colonne "write format" de l'attribut *relativeDistinguishedName*. Le filtre doit être formé d'un nombre quelconque de formules littérales et de noms d'attribut situés entre les signes inférieur à (<) et supérieur à (>).

Dans l'entité *Person*, par exemple, le filtre `cn=<cn>` ordonne à l'adaptateur d'ajouter à la formule littérale `cn=` la valeur de l'attribut appelé `cn`. Une application qui crée une nouvelle personne avec Ernest Hemingway comme nom commun affiche `cn=Ernest Hemingway` comme nom distinctif relatif.

Dans WebObjects 5.1, le module LDAP ne gère pas l'utilisation de qualificatif de comparaison de clé puisque LDAP ne gère pas la comparaison de valeur d'attribut. En outre, même si vous pouvez créer une relation entre les entités d'un modèle JNDI et d'un modèle JDBC, Enterprise Objects ne gère pas actuellement l'écrasement d'attributs ou de relations entre entités de modèles utilisant des adaptateurs différents. Pour plus d'informations sur d'autres questions, consultez le document *WebObjects 5.1 Release Notes* situé dans `/Developer/Documentation/WebObjects/` et sur le site <http://developer.apple.com/fr/webobjects/>.

Pour des informations sur le téléchargement et l'installation des packages JNDI et le fournisseur de services LDAP pour votre système, consultez le document *Post-Installation Guide*. Ce document aborde également les opérations fondamentales à réaliser pour l'utilisation des fonctionnalités LDAP et JNDI avec votre système et comporte des informations API pour cette nouvelle fonction.

Nouvelle documentation améliorée

WebObjects 5.1 comporte plusieurs nouveaux documents :

- *Inside WebObjects: WebObjects Desktop Applications*, qui fournit des guides d'initiation complets sur le développement d'applications avec Java Client et Direct to Java Client. Il remplace les documents *Java Client Tutorial* et *Getting Started with Direct to Java Client*.
- *Inside WebObjects: Developing EJB Applications*, qui explique comment utiliser WebObjects pour développer des applications EJB et fournit de la documentation sur EJB. Ce document suppose une connaissance préalable d'EJB.
- *Inside WebObjects: Developing Applications using Java Server Pages and Servlets*, qui explique comment déployer des applications WebObjects en tant que servlets et comment utiliser des composants WebObjects dans les pages JSP.
- *Post-Installation Guide*, dans lequel figurent des informations propres à chaque plate-forme sur les étapes à suivre après l'installation de WebObjects sur votre système. Ce document explique comment installer les adaptateurs JDBC et JNDI, comment configurer vos bases de données et comment désinstaller des versions ultérieures de WebObjects. Ce document contient également des informations de test de votre système.
- *Inside WebObjects: Developing SMIL Presentations*, qui détaille comment SMIL est implémenté dans WebObjects et vous apprend à créer une présentation SMIL dans WebObjects.

Les documents suivants ont été mis à jour pour WebObjects 5.1 :

Présentation de WebObjects

Inside WebObjects: Deploying WebObjects Applications

Guide d'installation pour Mac OS X

Guide d'installation pour Windows et Solaris

WebObjects 5.1 Release Notes

Nouveautés de WebObjects 5.1

Enfin, la documentation de référence sur la structure WebObjects est désormais disponible au format Javadoc. Consultez la section Javadoc suivante pour obtenir davantage d'informations et une liste complète des packages Javadoc fournis pour WebObjects 5.1.

Les documents ci-dessous figurent dans le dossier Documents du CD WebObjects 5.1 :

- Nouveautés de WebObjects 5.1*
- Guide d'installation pour Mac OS X*
- Guide d'installation pour Windows et Solaris*
- Post-Installation Guide*
- Présentation de WebObjects*
- Inside WebObjects: WebObjects Desktop Applications*
- Inside WebObjects: Deploying WebObjects Applications*
- Inside WebObjects: Developing EJB Applications*
- Inside WebObjects: Developing Applications with Java Server Pages and Servlets*
- Inside WebObjects: Discovering WebObjects for HTML*

Lorsque vous l'installez, WebObjects Developer pour Mac OS X installe ces documents ainsi que d'autres (dont la référence Javadoc et *WebObjects 5.1 Release Notes*) sur votre système. Ces documents sont installés dans le répertoire `/Developer/Documentation/WebObjects/`.

Pour accéder à la documentation de développeur sur Windows 2000, utilisez l'application WebObjects InfoCenter en allant dans le menu Démarrer et en sélectionnant Programmes > WebObjects > WOInfoCenter.

Vous trouverez une documentation et des ressources WebObjects complètes sur <http://developer.apple.com/fr/webobjects/>. Pour cette version, *Inside WebObjects: Developing SMIL Presentations* et les versions localisées des manuels fournis avec WebObjects 5.1 ne se trouvent pas sur le CD mais sont disponibles sur <http://developer.apple.com/fr/webobjects/>.

Pour obtenir des exemplaires imprimés de documents, rendez-vous sur <http://www1.fatbrain.com/documentation/apple/>.

Nouveautés de WebObjects 5.1

Javadoc

Il s'agit de la première version de la documentation de référence au format Javadoc. Javadoc apporte de nombreuses améliorations à cette documentation :

- les signatures de méthodes et les champs de classe sont précis étant donné qu'ils sont générés directement à partir du code source
- les références croisées dans les classes, dans les packages et entre les packages sont plus robustes
- le résultat est fourni dans un format plus simple
- un index de toutes les méthodes de tous les packages WebObjects publics est fourni
- un index indique le lieu d'utilisation de chaque package et chaque classe dans WebObjects
- chaque classe affiche sa superclasse et sa hiérarchie d'héritage jusqu'à `java.lang.Object`
- chaque classe répertorie les méthodes dont elle hérite de ses superclasses et des interfaces mises en œuvre

WebObjects 5.1 fournit Javadoc pour les packages suivants :

WebObjects (`com.webobjects.appserver`)
WebObjects XML (`com.webobjects.appserver.xml`)
DirectToWeb (`com.webobjects.directtoweb`)
EOAccess (`com.webobjects.eoaccess`)
EOApplication (`com.webobjects.eoapplication`)
EOControl (`com.webobjects.eocontrol`)
EODistribution (`com.webobjects.eodistribution`)
EODistribution Client (`com.webobjects.eodistribution.client`)
EOGeneration (`com.webobjects.eogeneration`)
EOGeneration Client (`com.webobjects.eogeneration.client`)
EOInterface (`com.webobjects.eointerface`)
EOInterface Swing (`com.webobjects.eointerface.swing`)
EOInterface Cocoa (`com.webobjects.eointerface.cocoa`)
Foundation (`com.webobjects.foundation`)
JDBC Adaptor (`com.webobjects.jdbcadaptor`)

Nouveautés de WebObjects 5.1

JDBC Adaptor (`com.webobjects.jdbcadaptor`)

JNDI Adaptor (`com.webobjects.jndiadaptor`)

WebObjects Extensions (`com.webobjects.woextensions`)

Les documents de référence ont été revus par des ingénieurs et comportent de nombreux ajouts par rapport à WebObjects 5.0. En particulier, EOApplication, EODistribution, DirectToWeb, EOInterface et JDBCAdaptor sont plus complets. La structure WebObjects, EOControl et Foundation ont également été étoffés. Enfin, les commentaires Javadoc évoluant avec le code source, les descriptions de méthode sont mises à jour à chaque modification du code source.

Neuerungen in WebObjects 5.1



Januar 2002

Inhalt

Neuerungen in WebObjects 5.1 3

Enterprise JavaBeans	3
Integration von JSP Servlets	4
Java Client	5
Class Loader	5
JavaClientLauncher	6
WOAutoOpenClientApplication	6
Stateless Remote Method Invocations	7
Security Delegate Methoden	7
Direct to Java Client Aktionen	8
DiscussionBoard	8
Unterstützung für LDAP und JNDI	8
Neue und erweiterte Dokumentation	10
Javadoc	12

Neuerungen in WebObjects 5.1

WebObjects 5.1, eine erweiterte Version von WebObjects 5.0, stellt neue Funktionen und Funktionalitäten zur Verfügung. Diese neue Version umfasst außerdem eine aktualisierte und neue Dokumentation. In diesem Dokument werden die folgenden Neuerungen und Änderungen beschrieben:

- **Enterprise JavaBeans** bietet Unterstützung für Enterprise JavaBeans (EJB).
- Mit **JSP Servlet Integration** können Sie WebObject Anwendungen auf JSP oder Servlet Application Servern bereitstellen.
- **Java Client** umfasst Erweiterungen und Änderungen.
- **LDAP und JNDI** unterstützen den LDAP Zugriff über einen JNDI Adaptor.
- Die **neue und erweiterte Dokumentation** umfasst einige neue und aktualisierte Handbücher und Referenzdokumente im Javadoc Format.

Enterprise JavaBeans

WebObjects 5.1 ist die erste Version von WebObjects, die die Integration von Enterprise JavaBeans (EJB) unterstützt. EJB ist ein wichtiger Bestandteil von J2EE. EJB stellt eine Umgebung bereit, in der Komponenten anderer Hersteller in einer Arbeitsanwendung zusammen gestellt werden können.

Bei WebObjects 5.1 können Sie mithilfe der WebObjects Developer Tools Enterprise Beans neu erstellen oder Beans von Drittanbietern in eine WebObjects Anwendung integrieren. Die Entwicklung von Beans erfolgt bei WebObjects 5.1 in zwei Phasen: Entwicklung (Development) und Bereitstellung (Deployment). Enterprise Beans werden entwickelt, indem Sie .java und Deployment Deskriptor Dateien erstellen; (die Komponente „Project Builder“ liefert Ihnen Vorlagen für diese Dateien).

Neuerungen in WebObjects 5.1

Außerdem können Sie den Quellcode oder JAR-Dateien für Enterprise Beans for Drittanbietern beziehen. Für die Bereitstellung eines oder mehrerer Beans generieren Sie ein Bean-Framework, das die JAR und die Deployment Descriptor Dateien des bzw. der Beans enthält. Dieses Framework müssen Sie auf dem für die Bereitstellung verwendeten Host in das Verzeichnis `/Library/Frameworks` kopieren. Nachdem das Bean-Framework bereitgestellt wurde, kann es in Client-Anwendungen integriert und darin verwendet werden.

Weitere Informationen über WebObjects und EJB finden Sie in der Dokumentation *Inside WebObjects: Developing EJB Applications* (vgl. Abschnitt „Neue und erweiterte Dokumentation“ in diesem Dokument). Framework-spezifische Dokumente stehen für die Frameworks JavaOpenEJB, JavaOpenTM und JavaOpenORB zur Verfügung. Sie können diese Dokumente in Project Builder zu Rate ziehen. Öffnen Sie dazu zunächst den Framework-Bereich und danach den Dokumentationsbereich in den oben genannten Frameworks.

Integration von JSP Servlets

WebObjects 5.1 unterstützt nun generische WebObjects Anwendungen, die als Servlet innerhalb eines Servlet-Containers wie Tomcat oder WebLogic ausgeführt werden können. Das bedeutet, dass das Servlet nicht als eigener JVM-Prozess, sondern (zusammen mit potenziell vielen anderen Anwendungen) im JVM des Servlet-Containers ausgeführt wird. Bei der Bereitstellung einer WebObjects Anwendung als Servlet sollten Sie beachten, dass keines der vorhandenen Werkzeuge für die Bereitstellung (WebServer Adaptor, wotaskd, Monitor) verfügbar ist.

Bei WebObjects 5.1 können JSP Seiten nun direkt auf WebObjects Komponenten zugreifen. JSP Seiten, die demselben Servlet-Kontext angehören (vergleichbar einer einzelnen Anwendung), verwenden eine vorgegebene Tag-Bibliothek, um die HTML-Äquivalente für die WebObjects Komponenten zu generieren. Dabei ist zu beachten, dass die JSP Seiten demselben WAR-Archiv angehören müssen, um demselben Kontext anzugehören. Als JSP Entwickler können Sie zwei Arten von Komponenten verwenden: WOCComponents und DirectActions.

Weitere Informationen über die Integration von JSP Servlets finden Sie in der Dokumentation *Inside WebObjects: Developing WebObjects Applications with Java Server Pages and Servlets* (siehe Abschnitt „Neue und erweiterte Dokumentation“ in diesem Dokument).

Java Client

Java Client in WebObjects 5.1 bietet zahlreiche Optimierungen und Aktualisierungen gegenüber WebObjects 5.0. Zahlreiche Fehler wurden behoben, und in verschiedenen Bereichen wurde die Leistung verbessert. Für WebObjects 5.1 wurde die Dokumentation erweitert und verbessert. Neu hinzugekommen sind Referenzdokumente für Klassen und das Handbuch *WebObjects Desktop Applications*. Dieses Handbuch enthält unter logischen Gesichtspunkten strukturierte Informationen und führt Sie durch Lernprogramme, bei denen sowohl die Funktion „Java Client“ als auch „Direct to Java Client“ angesprochen wird. Eine Vielzahl kleinerer neuer Funktionen dienen dem Ziel, Sicherheitsaspekte zu optimieren und das Starten von Client-Anwendungen zu verbessern.

Class Loader

Im Hinblick auf Desktop-Anwendungen stellt WebObjects 5.1 eine spezielle Funktion zum Laden von Klassen bereit, die die Entwicklungszyklen beschleunigt und ein höheres Maß an Flexibilität bei der Bereitstellung erlaubt. Mit dieser Funktion können Sie zusätzlich zu den Java Client Standardklassen, die auf dem Client-Computer vorinstalliert sein müssen, eigene spezifische Klassen Ihrer Anwendung in den Desktop-Anwendungs-Client laden. Bei früheren Versionen mussten alle Klassen (auch die eigenen spezifischen Klassen) vorinstalliert sein.

Dies erweist sich sowohl bei der Entwicklung als auch bei der Bereitstellung als Vorteil, da Sie sich nicht um die Konfiguration von Klassenpfaden kümmern müssen und bei Client-Installation sehr viel flexibler agieren können. Bei früheren Versionen war es erforderlich, die komplette Client-Anwendung vorzinstallieren — nun müssen Sie nur noch das Java Client Basissystem vorinstallieren (ohne irgendwelche eigene Klassen). Wenn Sie die Klassen Ihrer Client-Anwendung aktualisieren wollen, müssen Sie kein Installationsprogramm senden — es genügt vielmehr, wenn Sie Ihre Klassen auf dem Server aktualisieren, sodass die Clients sie beim nächsten Starten automatisch laden können. Sie müssen die Client-Anwendung nur aktualisieren, wenn Sie die verwendete WebObjects Version ändern.

Neuerungen in WebObjects 5.1

Die Funktion zum Laden von Klassen wird im `WOJavaClientApplet` Ihrer Anwendung mithilfe der neuen Bindeart `downloadClientClasses` konfiguriert. Weitere Informationen finden Sie in der Klassenbeschreibung für `WOJavaClientApplet`. Unter Mac OS X werden Sie von den Project Builder Assistenten beim Erstellen neuer Projekte gefragt, welche Option zum Laden von Klassen Sie verwenden wollen. Entsprechend Ihrer Antwort wird `WOJavaClientApplet` automatisch für Sie konfiguriert. Unter Windows konfigurieren die Assistenten `WOJavaClientApplet` so, dass die Funktion zum Laden von Klassen deaktiviert ist. In diesem Fall müssen Sie den korrekten Wert für die Bindeart „downloadClientClasses“ in der Komponente mit `WOJavaClientApplet` manuell konfigurieren; (in der Regel handelt es sich dabei um die Komponente „Main“).

JavaClientLauncher

Unter Mac OS X umfasst WebObjects 5.1 mit `JavaClientLauncher` ein neues Beispiel. `JavaClientLauncher` veranschaulicht, wie Sie ein Startprogramm für Mac OS X erstellen, das das Java Client Basissystem umfasst und so konfiguriert ist, dass beim Starten ein Dialogfenster zur Eingabe einer URL ausgeführt wird. Alles was Sie tun müssen ist, `JavaClientLauncher` in das Dock zu platzieren, zu starten und die URL einer Java Client Serveranwendung einzugeben, um die Verbindung direkt aus dem Dialogfenster für die URL-Adresse herzustellen. Sofern Ihre Serveranwendungen so konfiguriert sind, dass die neue Funktion zum Laden von Klassen verwendet wird, können Sie mit `JavaClientLauncher` jede Client-Anwendung ausführen (da sie Ihre anwendungsspezifischen Klassen beim Starten lädt). Das Readme-Dokument für das Beispiel „`JavaClientLauncher`“ enthält Anweisungen für das Konfigurieren eigener Startprogramme.

Unter Windows stellt WebObjects 5.1 ein neues Skript bereit. Es heißt `JavaClientLauncherWin.cmd` und befindet sich im Verzeichnis mit den WebObjects Beispielen. Es hat die gleiche Funktionalität wie das Programm „`JavaClientLauncher`“ unter Mac OS X und ist ausführlich dokumentiert. Bei Bedarf können Sie auf Ihrem Schreibtisch eine Verknüpfung für dieses Skript anlegen.

WOAutoOpenClientApplication

Für Mac OS X können Sie mit WebObjects beim Erstellen eines Projekts die Skripts zum Starten von Clients automatisch erstellen lassen. Sie aktivieren bzw. deaktivieren die Generierung von Skripten für das Starten von Clients in Project Builder in den Build-Einstellungen für Ihren Zielservers. WebObjects 5.1 stellt einen

Neuerungen in WebObjects 5.1

neuen Merker (Flag) für die serverseitige Anwendung zur Verfügung: `WOAutoOpenClientApplication`. Dieser Merker hat eine ähnliche Funktion wie `WOAutoOpenInBrowser` — der Client wird nicht als Applet im Browser gestartet, vielmehr wird das Client-Start-Skript verwendet, um eine Desktop-Anwendung zu starten.

Wenn Sie `WOAutoOpenClientApplication` auf den Wert `Yes` setzen, wird der Client als Desktop-Anwendung gestartet, sofern das Skript zum Starten des Clients gefunden wird. Wird es nicht gefunden, verwendet die Anwendung die Funktionalität `WOAutoOpenInBrowser` und startet den Client als Applet. Wenn Sie dieses Verhalten ausschalten wollen, müssen Sie sowohl `WOAutoOpenClientApplication` als auch `WOAutoOpenInBrowser` auf `No` setzen.

Unter Windows werden die Generierung von Skripts für das Starten von Clients und `WOAutoOpenClientApplication` nicht unterstützt. In diesem Fall steht Ihnen nur `WOAutoOpenInBrowser` zur Verfügung.

Stateless Remote Method Invocations

Die Klasse `EODistributedObjectStore` enthält nun die neue Methode `invokeStatelessRemoteMethodWithKeyPath`. Es handelt sich dabei um eine schnellere und einfachere Version von `invokeRemoteMethodWithKeyPath`, die für die gesamte Kommunikation eingesetzt werden kann, die sich nicht auf Enterprise Objekte bezieht.

Sie können diese neue Methode für Remote Method Invocations verwenden, die nur die Anwendungs-, nicht jedoch die Business-Logik betreffen. Sie können sie zum Beispiel zum Laden einer Ressource heranziehen. Die Methode ist auch für Threads im Hintergrund sicher (sofern keine Enterprise Objekte involviert sind), und es ist auch eine sichere Methode für serverseitige Prüfungen von Threads im Hintergrund.

Security Delegate Methoden

Der Store-Delegat `EODistributedObjects` enthält mehrere neue Methoden. Weitere Informationen finden Sie in der Klassenbeschreibung für `EODistributedObjects.Delegate`. Mit den neuen Methoden können Sie noch besser steuern, welche Arten von Daten Client-Anwendungen abrufen und ändern dürfen.

Direct to Java Client Aktionen

Bei früheren Versionen von Direct to Java Client konnten Standardaktionen wie Save oder Undo nicht konfiguriert werden. Mit WebObjects 5.1 können Sie diese Standardaktionen nun über ein Regelsystem konfigurieren. Weitere Informationen finden Sie in der vom Regelsystem generierten XML für die Aktionsfrage. (Sie können die XML in der XML-Anzeige des Assistenten ansehen.)

DiscussionBoard

WebObjects 5.1 enthält ein weiteres neues Beispiel: DiscussionBoard. Hierbei handelt es sich um ein erweitertes Beispiel, das mithilfe von Direct to Java Client eine Anwendung für eine Diskussionsrunde implementiert. Zum Lieferumfang dieses Beispiels gehören ein Readme-Dokument und eine umfassende Dokumentation.

Das Beispiel veranschaulicht viele Aspekte von Java Client, insbesondere Aspekte der Identifizierung (An- und Abmeldung), der Sicherheit im Hinblick auf die Identifizierung (unterschiedliche Zugriffsrechte für Administratoren und normale Benutzer), der Anpassung der Benutzeroberfläche, der Prüfung von Threads im Hintergrund und andere mehr. Wenn Sie Java Client verwenden wollen, sollten Sie sich dieses Beispiel unbedingt ansehen.

Unterstützung für LDAP und JNDI

Zum Lieferumfang von WebObjects 4.5 gehört ein LDAP-Adaptor (Lightweight Directory Access Protocol) als Beispiel für einen in Objective-C erstellten Adaptor. WebObjects 5.1 unterstützt den LDAP Zugriff mithilfe eines JNDI Adaptors (Java Naming and Directory Interface) – ein in Java erstellter Adaptor, der voll unterstützt wird und mit Java Runtime von WebObjects kompatibel ist. Mit WebObjects können Sie damit Verbindungen zu JNDI Datenquellen von allen unterstützten WebObjects Anwendungsarchitekturen herstellen: Cocoa/EOF Anwendungen, Java Client Anwendungen und HTML Serveranwendungen.

Neuerungen in WebObjects 5.1

Bei WebObjects 5.1 umfasst der JNDI Adaptor das LDAP Zusatzmodul, und er unterstützt die folgenden Server:

- OpenLDAP LDAP Server
- iPlanet Directory Server

Beachten Sie die folgenden Analogien und Unterschiede von JNDI und JDBC Adaptoren:

- Mithilfe von EOModeler können Sie die Erstellung oder das Reverse Engineering für ein JNDI Modell mit demselben Ansatz wie für ein JDBC Modell durchführen.
- Im Gegensatz zum JDBC Adaptor unterstützt der JNDI Adaptor aber nicht das Synchronisieren von Schemata. Der/das JNDI Adaptor/LDAP Zusatzmodul kann für LDAP Server verwendet werden, die mit vordefinierten Schemata geliefert werden. Zum Ändern von Schemata stehen Werkzeuge von Drittanbietern und andere Mechanismen zur Verfügung.
- Jede Entität in einem JNDI Modell verfügt über das Attribut *relativeDistinguishedName*, das als Primärschlüssel fungiert. Der RDN (Relative Distinguished Name) steht für eine eindeutige, für den Menschen lesbare Kennung eines Eintrags auf einem Verzeichnisserver.

Um Ihnen die Arbeit zu erleichtern, unterstützt der JNDI Adaptor einen Mechanismus, der Sie bei der Erstellung eines RDN für einen neuen Eintrag unterstützt. Wenn Sie die automatische Generierung von Primärschlüsseln aktivieren wollen, müssen Sie nur ein Muster in der Schreibformatpalte des Attributs *relativeDistinguishedName* angeben. Das Muster darf eine beliebige Anzahl von Literalen und Attributnamen umfassen, die zwischen den Zeichen < (kleiner als) und > (größer als) eingebettet sind.

Beispiel: Für die Entität „Person“ weist das Muster `cn=<cn>` den Adaptor an, das Literal `cn=` mit dem Wert des Attributs `cn` anzuhängen. Für eine Anwendung, die eine neue Person mit Namen Ernest Hemingway als Common Name erstellt, wird `cn=Ernest Hemingway` als RDN angezeigt.

Bei WebObjects 5.1 bietet das LDAP Zusatzmodul keine Unterstützung für Qualifier für den Vergleich von Schlüsseln, da LDAP den Vergleich von Attributwerten nicht unterstützt. Außerdem ist zu beachten, dass Sie zwar Relationen zwischen Entitäten in einem JNDI und einem JDBC Modell herstellen können, dass Enterprise Objects derzeit aber das Verflachen von Attributen oder Relationen zwischen Entitäten in Modellen nicht unterstützt, die unterschiedliche

Neuerungen in WebObjects 5.1

Adaptoren verwenden. Weitere Informationen über andere bekannte Aspekte finden Sie in den *WebObjects 5.1 Release Notes* im Verzeichnis `/Developer/Documentation/WebObjects/` und online unter der Adresse <http://developer.apple.com/webobjects/>.

Informationen über das Laden und Installieren der JNDI Pakete und LDAP Service Provider für Ihr System finden Sie im Handbuch *Post-Installation Guide*. In diesem Dokument werden auch die Basisoperationen für den Einsatz der LDAP und JNDI Funktionalität auf Ihrem System beschrieben. Außerdem finden Sie hier nützliche API-Informationen für diese neue Funktion.

Neue und erweiterte Dokumentation

Zum Lieferumfang von WebObjects 5.1 gehören einige neue Dokumente:

- *Inside WebObjects: WebObjects Desktop Applications*: Hier finden Sie umfassende Lernprogramme für die Entwicklung von Anwendungen mithilfe von Java Client und Direct to Java Client. Dieses Dokument ersetzt die früheren Dokumente *Java Client Tutorial* und *Getting Started with Direct to Java Client*.
- *Inside WebObjects: Developing EJB Applications*: Hier erfahren Sie, wie Sie WebObjects für die Entwicklung von EJB Anwendungen einsetzen können. Außerdem finden Sie hier nützliche Hintergrundinformationen über EJB. Dieses Dokument setzt die Kenntnis von EJB voraus.
- *Inside WebObjects: Developing Applications using Java Server Pages and Servlets*: Hier erfahren Sie, wie Sie WebObjects Anwendungen als Servlets bereitstellen und wie Sie WebObjects Komponenten in JSP Seiten verwenden.
- *Post-Installation Guide*: Hier finden Sie plattformspezifische Informationen über die Schritte, die Sie ausführen müssen, nachdem Sie WebObjects auf Ihrem System installiert haben. Das Handbuch beschreibt, wie Sie JDBC und JNDI Adaptoren installieren, wie Sie Ihre Datenbanken konfigurieren und wie Sie ältere Versionen von WebObjects deinstallieren. Außerdem enthält dieses Dokument Informationen darüber, wie Sie Ihr System testen.
- *Inside WebObjects: Developing SMIL Presentations*: Hier finden Sie detaillierte Informationen darüber, wie SMIL in WebObjects implementiert ist. Außerdem erfahren Sie, wie Sie in WebObjects eine SMIL Präsentation erstellen.

Neuerungen in WebObjects 5.1

Die folgenden Dokumente wurden für WebObjects 5.1 aktualisiert:

WebObjects Overview
Inside WebObjects: Deploying WebObjects Applications
Installation unter Mac OS X
Installation Guide for Windows and Solaris
WebObjects 5.1 Release Notes

Darüber hinaus steht die Referenzdokumentation für das WebObjects Framework nun auch im Javadoc Format zur Verfügung. Ausführliche Informationen und eine umfassende Liste der für WebObjects 5.1 verfügbaren Javadoc Pakete finden Sie im nachfolgenden Abschnitt „Javadoc“.

Im Ordner „Documents“ auf der CD-ROM für WebObjects 5.1 sind die folgenden Dokumente enthalten:

Neuerungen in WebObjects 5.1
Installation unter Mac OS X
Installation Guide for Windows/Solaris
Post-Installation Guide
WebObjects Overview
Inside WebObjects: WebObjects Desktop Applications
Inside WebObjects: Deploying WebObjects Applications
Inside WebObjects: Developing EJB Applications
Inside WebObjects: Developing Applications with Java Server Pages and Servlets
Inside WebObjects: Discovering WebObjects for HTML

Diese und andere Dokumente (einschließlich der Javadoc Referenz und der *WebObjects 5.1 Release Notes*) werden automatisch auf Ihrem System installiert, wenn Sie WebObjects Developer for Mac OS X installieren. Die Dokumente werden im Verzeichnis `/Developer/Documentation/WebObjects/` installiert.

Unter Windows 2000 können Sie auf die Entwicklerdokumentation mithilfe der Anwendung „WebObjects InfoCenter“ zugreifen. Klicken Sie dazu auf „Start“ und wählen Sie „Programme > WebObjects > WOInfoCenter“.

Eine komplettes Set von WebObjects Dokumenten und Ressourcen finden Sie auch unter der Adresse: <http://developer.apple.com/webobjects/>. *Inside WebObjects: Developing SMIL Presentations* und die lokalisierten Versionen der Handbücher, die

Neuerungen in WebObjects 5.1

mit dem Produkt „WebObjects 5.1“ geliefert werden, sind bei dieser Version nicht auf der Produkt-CD-ROM enthalten; sie können aber <http://developer.apple.com/webobjects/> geladen werden.

Wenn Sie gedruckte Kopien bestimmter Dokumente erhalten wollen, besuchen Sie bitte die Web-Site: <http://www1.fatbrain.com/documentation/apple/>.

Javadoc

Dies ist die erste Version der WebObjects Referenzdokumentation im Javadoc Format. Javadoc bringt eine Vielzahl von Verbesserungen und Vorteilen für die Referenzinformationen:

- Signaturen für Methoden und Felder für Klassen sind präzise, da sie direkt aus dem Quellcode generiert werden.
- Querverweise innerhalb von Klassen und Paketen und zwischen Paketen sind zuverlässiger.
- Die Ausgabe wird in einem besser erfassbaren Format bereitgestellt.
- Es gibt einen Index aller Methoden in allen öffentlichen WebObjects Paketen.
- Ein Nutzungsindex dokumentiert, an welchen Stellen in WebObjects jedes Paket und jede Klasse verwendet wird.
- Jede Klasse gibt Auskunft über die eigene Superklasse und die Vererbungshierarchie bis hin zu `java.lang.Object`.
- Jede Klasse gibt Auskunft über die Methoden, die von den eigenen Superklassen und den implementierten Schnittstellen geerbt wurden.

WebObjects 5.1 stellt Javadoc Referenzdokumente für die folgenden Pakete bereit:

WebObjects (`com.webobjects.appserver`)
WebObjects XML (`com.webobjects.appserver.xml`)
DirectToWeb (`com.webobjects.directtoweb`)
EOAccess (`com.webobjects.eoaccess`)
EOApplication (`com.webobjects.eoapplication`)
EOControl (`com.webobjects.eocontrol`)
EODistribution (`com.webobjects.eodistribution`)
EODistribution Client (`com.webobjects.eodistribution.client`)
EOGeneration (`com.webobjects.eogeneration`)

Neuerungen in WebObjects 5.1

EOGeneration Client (`com.webobjects.eogeneration.client`)

EOInterface (`com.webobjects.eointerface`)

EOInterface Swing (`com.webobjects.eointerface.swing`)

EOInterface Cocoa (`com.webobjects.eointerface.cocoa`)

Foundation (`com.webobjects.foundation`)

JDBC Adaptor (`com.webobjects.jdbcadaptor`)

JDBC Adaptor (`com.webobjects.jdbcadaptor`)

JNDI Adaptor (`com.webobjects.jndiadaptor`)

WebObjects Extensions (`com.webobjects.woextensions`)

Hervorzuheben ist auch die Tatsache, dass der Inhalt der Referenzdokumente von Produktingenieuren überarbeitet wurde und die Dokumente gegenüber WebObjects 5.0 zahlreiche Ergänzungen und Erweiterungen umfassen. Insbesondere EOApplication, EODistribution, DirectToWeb, EOInterface und JDBCAdaptor sind sehr viel umfassender dokumentiert. Auch die Informationen für WebObjects Framework, EOControl und Foundation wurden ergänzt und erweitert. Und da die Javadoc Kommentare im Quellcode angesiedelt sind, werden die Beschreibungen der Methoden aktualisiert, wenn sich der Quellcode ändert.

www.apple.com