



Web サービス チュートリアル

Release 12

© 2002-2008 Unify Corporation All rights reserved. Sacramento California, USA

No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise without the prior written consent of Unify Corporation.

Unify Corporation makes no representations or warranties with respect to the contents of this document and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, Unify Corporation reserves the right to revise this document and to make changes from time to time in its content without being obligated to notify any person of such revisions or changes.

The Software described in this document is furnished under a Software License Agreement. The Software may be used or copied only in accordance with the terms of the license agreement. It is against the law to copy the Software on tape, disk, or any other medium for any purpose other than that described in the license agreement.

The Unify Corporation Documentation Group values and appreciates any comments you may have concerning our documents. Please address comments to:

doc@unify.com

1-800-GO-UNIFY or 1-800-24-UNIFY; (916) 928-6400
FAX (916) 928-6401

UNIFY and DataServer are registered trademarks of Unify Corporation. Unify NXJ is a trademark of Unify Corporation. Java and J2EE are registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. JReport is a trademark of Jinfonet Corporation. IBM, Lotus, Lotus Notes, Cloudscape, and WebSphere are trademarks of International Business Machines Corporation in the United States, other countries, or both. CAS AHL Technology and ecKnowledge are registered trademarks of CAS AHL Technology, Inc. in the U.S. and other countries. All other products or services mentioned herein may be registered trademarks, trademarks, or service marks of their respective manufacturers, companies, or organizations.

Name: Web Services Tutorial

Release: Unify NXJ 12

Last Revision: January 5, 2009 4:32 pm

Web サービスチュートリアル

このチュートリアルは、Unify NXJ の NXJ 開発ツールと ActiveSOA の機能を使って Web サービスを作成する方法を紹介します。Web サービスの概要を説明した後、既存の NXJ アプリケーション「Order Management (注文管理)」を変更して、Shipping (出荷) という Web サービスを追加します。この Shipping は、Order Management アプリケーションの注文データにユニークなトラッキング番号、出荷日、出荷費用を追加します。

このチュートリアルで次のタスクを習得します。

- Order Management プロジェクトを開いて準備
- Shipping ビジネスサービスの作成
- 既存コンポーネントの準備
- ビジネスサービスの呼び出し
- ビジネスサービスのテスト

このチュートリアルは、Unify NXJ と NXJ 開発ツールには、既に慣れていることを前提としています。もし、まだこれらに慣れていない場合には、このチュートリアルを行う前に NXJ チュートリアルを習得することをお勧めします。

注： このチュートリアルには、Order Management アプリケーションに追加すべきソースコードが記載されています。このドキュメントからコピーをして、Unify NXJ 開発ツールに貼り付けることができます。

Web サービスとは？

Web サービスは、他のアプリケーションが HTTP によりインターネット経由で呼び出すことができる独立したモジュールです。互いに連携して動く複数の Web サービスがサービス指向アーキテクチャ (SOA) を構成します。SOA は、プラットフォーム、言語、場所に関係なく検索をして呼び出すことができる分散コンピューティングアーキテクチャです。

プロバイダと呼ばれる企業なりビジネスが、Web サービスを作成し、WSDL (Web サービス定義言語) を使ってサービスの配置された場所とサービスが持っているメソッドを記述します。プロバイダは、サービスをサーバにアップロードして、この WSDL をディレクトリに公開します。一つのディレクトリには、様々なプロバイダが公開した数多くの他のサービスも登録されています。

利用者と呼ばれるアプリケーションや他の Web サービスは、このディレクトリを使って Web サービスを検索します。ディレクトリは、利用者がプロバイダとコンタクトをしてサービスを呼び出すために、該当するサービスの WSDL を返します。プロバイダ、ディレクトリ、利用者は、この通信に SOAP (Simple Object Access Protocol) メッセージを使います。

Unify NXJ の Web サービス

Web サービスと使用すると、既存のプログラムロジックを再利用することで NXJ アプリケーションの開発を簡単に行うことができます。また、Web サービスから NXJ アプリケーションフォームを作成することもできます。Unify NXJ は、non-NXJ Web サービス、ビジネスサービス、データサービスの 3 タイプの Web サービスをサポートします。

注： Unify NXJ における Web サービスの詳細は、『開発者ガイド』の「Web サービス」の章を参照してください。

Non-NXJ Web サービス

Non-NXJ Web サービスは、NXJ プログラミング言語以外で記述されたもので、Spam フィルタやスペルチェッカーなどのように Web サービスディレクトリからダウンロードして NXJ アプリケーションにインポートすることができます。

ビジネスサービス

ビジネスサービスは、NXJ プログラミング言語で記述された Web サービスです。Unify NXJ は、ビジネスサービスを .fj ソースファイルと .fx プロパティファイルとして 2 つのファイルに保存し、Classes の下のノードに表示します。ビジネスサービスは、Web サービスなので、NXJ と NXJ 以外のアプリケーションの両方で使用することができます。

このチュートリアルでは、ビジネスサービスを作成します。従って、このチュートリアルでは、ビジネスサービスといった場合は Shipping を示すことになります。

データサービス

データサービスは、データベーステーブルの階層をモデル化し、開発者ガイドに記載されているデータサービスデザインパターンに準拠し、FIND、INSERT、UPDATE、DELETE などのメソッドを提供する Web サービスです。ウィザードを使用して、データサービスを作成したり、NXJ プログラミング言語で直接記述することもできます。

タスク 1: Order Management プロジェクトを開いて準備

NXJ アプリケーション内で Web サービスを実行するには、Class パスに ActiveSOA.jar を含めなければなりません。この作業は、通常、Web サービスを使う新規プロジェクトを作成するときに行います。しかしながら、Order Management アプリケーションは、初めは Web サービスの使用を設定されていないため、今、改めて行わなければなりません。このタスクを行わないと、Order Management は Shipping Web サービスを使用することができません。

注： Order Management プロジェクトをコピーして、そのコピーに対して作業を行うことで、元のプロジェクトを変更なしに保存しておくことができるので、他の人が再びチュートリアルを実行することができるようになります。Order Management プロジェクトは、次の場所に保存されています。

```
<UNIFY_HOME>%NXJWork%projects%examples%ordmgmt%
```

1. Windows のスタートメニューから、**すべてのプログラム > Unify NXJ > NXJ 開発ツール** を選択します。

NXJ 開発ツールが起動して、直近に使用したプロジェクトが開きます。

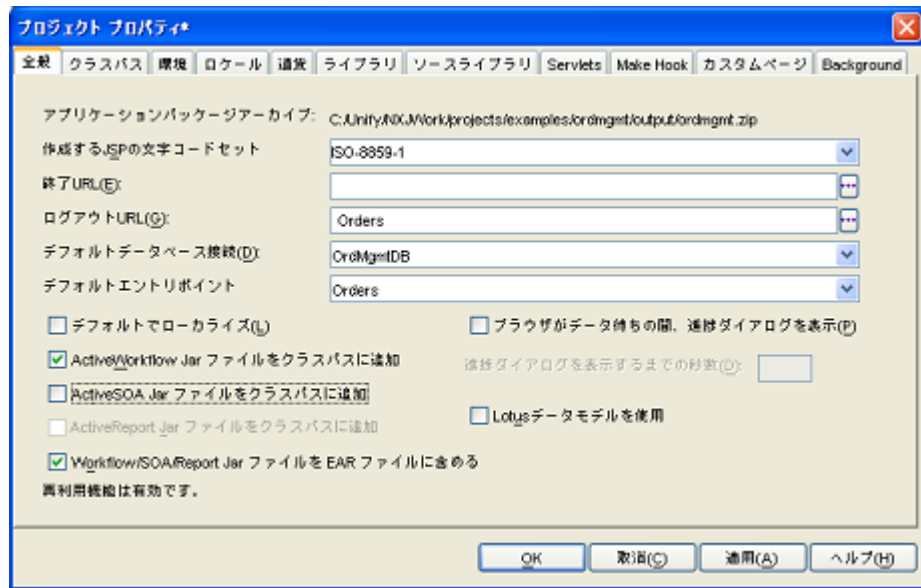
2. NXJ 開発ツールのメニューで **ファイル > プロジェクトを開く** を選択します。
3. ordmgmt.prj プロジェクトファイルをブラウザして選択します。

Order Management プロジェクトのコピーを作成している場合は、そのプロジェクトの ordmgmt.prj ファイルをブラウザします。そうでない場合は、以下のプロジェクトファイルを表示します。

```
<UNIFY_HOME>%NXJWork%projects%examples%ordmgmt%ordmgmt.prj
```

4. **ブラウザパネルのプロジェクトタブで、プロジェクトプロパティ** をクリックします。

プロジェクトプロパティダイアログが表示されます。



5. 全般タブ上の **ActiveSOA Jar** ファイルをクラスパスに追加のチェックボックスをセットします。
6. 適用をクリックしてから、**OK** をクリックします。

実行時データソースの作成 ダイアログが表示された場合、**はい** をクリックします。これにより、NXJ は、アプリケーションサーバにデータソースを作成します。

タスク 2: Shipping ビジネスサービスの作成

プロジェクトが開くと、次に、Shipping ビジネスサービスの作成に移ります。NXJ 開発ツール上で NXJ プログラミング言語を使ってこれを行います。最初にビジネスサービス用のフォルダを作成してから、ビジネスサービスを作成します。

1. NXJ 開発ツールのブラウザパネルのプロジェクトタブで、**Classes** を右クリックして、**新規作成 > フォルダ** を選択します。
2. フォルダの名前を **Services** に変更します。
3. **Services** を右クリックして、**新規作成 > ビジネスサービス** を選択します。
NXJ 開発ツールは、ビジネスサービスを作成して、それをコンテンツパネルに開きます。
4. ビジネスサービスの名前を **Shipping** に変更します。
5. コンテンツパネルの **ソース** タブで、以下のコードが表示されます。

```
SERVICE Shipping
{
}
```

6. 左右の括弧の間に以下のコードを追加します。

```
public java.math.BigDecimal getShipPrice(java.math.BigDecimal
input)
{
    return input.multiply(new java.math.BigDecimal(0.0175));
}
```

7. (オプション) **コンテンツパネルのプロパティタブ**で、**Service > Publish as Web Service** チェックボックスをセットします。

このチェックボックスをセットすると、他のプロジェクトからこのビジネスサービスが利用可能になります。このチュートリアルではオプションの作業ですが、複数のプロジェクトから利用させたい Web サービスを作成する場合には、これは必須の作業になります。

8. NXJ 開発ツールのツールバーで、**保存**をクリックします。
9. NXJ 開発ツールメニューで、**プロジェクト > Make Shipping** を選択します。
Unify NXJ は、Order Management アプリケーションに、このビジネスサービスを統合するために必要となるクラスファイルを作成します。これらのファイルは、プロジェクトの `output\web\%project%\%compilation%\Services` フォルダに置かれます。ここには、ビジネスサービスの WSDL も置かれています。

タスク 3: 既存コンポーネントの準備

OrderDetailView コンポーネントは、各注文のトータルコストを保存する変数を使用します。Order Management アプリケーションの元のデザインでは、この変数は使用していませんでしたが、ビジネスサービスは利用します。この変数をビジネスサービスに見えるようにするために、Public にしなくてはなりません。また、productChooser は、適切に注文の Shipping コストの合計を追加する変更が必要になります。

Unify NXJ における通常の Web サービス開発では、最初から Web サービスサポートを NXJ アプリケーションに組み込むことが可能で、Order Management アプリケーションに対して組み込むためのコンポーネントの再構築は必要ありません。

1. NXJ 開発ツールの **ブラウザパネル > プロジェクトタブ**で、**Classes > Components** を開いて、**OrderDetailView** を選択します。
コンテンツパネルにコンポーネントが表示されます。

2. **コンテンツパネル > ソースタブ**で、次のコードの場所を探します。

```
DATA VIEW orderDetailView
{
    NullableAmountVariable orderTotal = 0.0;
```

3. **NullableAmountVariable** の左側に、次のように `public` とタイプします。

```
DATA VIEW orderDetailView
{
    public NullableAmountVariable orderTotal = 0.0;
```

4. NXJ 開発ツールメニューで、**ファイル > 保存** を選択します。
5. NXJ 開発ツールの**ブラウザパネル > プロジェクトタブ**で、**Classes > Components** を表示し、**productChooser** を選択します。
コンテンツパネルにコンポーネントが表示されます。
6. **コンテンツパネル > デザインタブ**で、コンポーネントを選択します。
7. **ブラウザパネル > プロパティタブ**で、**List > Query** プロパティを以下のように変更します。

```
SELECT PRODUCT_DESC, PRODUCT_ID
FROM ORDINV.PRODUCT
```

既存のクエリを変更する場合、Query ダイアログを使用する点に注意してください。終了したら、**OK** をクリックして下さい。
8. **Ctrl+S** を押下して、コンポーネントを保存します。

タスク 4: ビジネスサービスの呼び出し

Order Management は、利用者が注文の shipping コストを計算したいときに、Shipping ビジネスサービスにアクセスします。この機能を追加するには、getShippingCost コマンドを呼び出す Shipping 計算 ボタンを Fill_Order activity フォームに追加しなければなりません。このコマンドは、ビジネスサービスを呼び出し、OrderDetailView コンポーネントを動かし、そしてデータベースを更新します。

1. NXJ 開発ツールの**ブラウザパネル > プロジェクトタブ**で、**Classes** の下にある **Fill_Order** をクリックします。

Unify NXJ は、ActiveWorkflow を初期化します。

2. **コンテンツパネル > ソースタブ**で、ページの先頭にある以下の部分を探します。

```
ACTIVITY FORM Fill_Order
{
```

3. この行のすぐ上に次のコードを追加します。

```
import java.math.BigDecimal;
```

4. **コンテンツパネル > ソースタブ**で、以下のコードを探します。

```
BOX box1
{
    FIELD Order_Number
    {
    }
    FIELD Process_Title
    {
    }
    FIELD CUST_CD
    {
    }
}
```

```
FIELD Process_Invoker
{
}
}
```

5. 最後の右括弧のすぐ下に、以下のコードを追加します。

```
COMMAND getShippingCost()
{
  try {
    Services.Shipping shipObject;
    shipObject = new Services.Shipping();

    BigDecimal subTotal;
    BigDecimal shippingCost;
    BigDecimal orderTotal;

    subTotal = view1.tabset1.view6.orderTotal.getBigDecimal();
    shippingCost = shipObject.getShipPrice(subTotal);
    orderTotal = subTotal.add(shippingCost);
    view1.tabset1.view5.TRACKING_NUMBER =
    view1.tabset1.view5.getNextNum("TRACKING_NO");
    view1.tabset1.view5.SHIPPING_COST.setBigDecimal(shippingCost);
    view1.tabset1.view5.SHIP_DATE = session.currentDate;
    view1.tabset1.view6.orderTotal.setBigDecimal(orderTotal);

    //Hides the Calc Shipping button and displays the Filled button
    calcShippingButton.visible = false;
    orderFilledButton.visible = true;

    //Store the shipping cost
    EXEC SQL USING CONNECTION OrdMgmtDB
    UPDATE ORDINV.ORDERS SET
      TRACKING_NUMBER = :view1.tabset1.view5.TRACKING_NUMBER,
      SHIP_DATE = :view1.tabset1.view5.SHIP_DATE,
      SHIPPING_COST = :view1.tabset1.view5.SHIPPING_COST
    WHERE ORDER_NUMBER = :box1.Order_Number;

    //Update the Total Cost
    EXEC SQL USING CONNECTION OrdMgmtDB
    INSERT INTO ORDINV.ORDER_DETAIL (
      ORDER_NUMBER,
      PRODUCT_NUMBER,
      PRODUCT_PRICE)
      values ( :box1.Order_Number, '40300',
:view1.tabset1.view5.SHIPPING_COST);

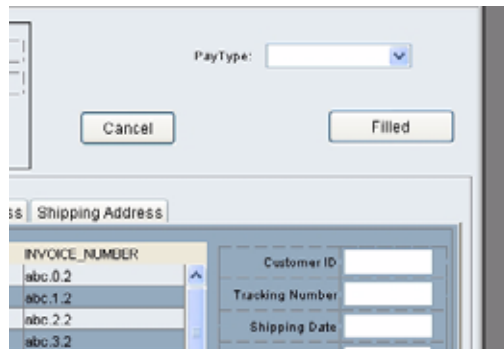
    session.queueCommand("ADD_UPDATE");
  }
}
```


```

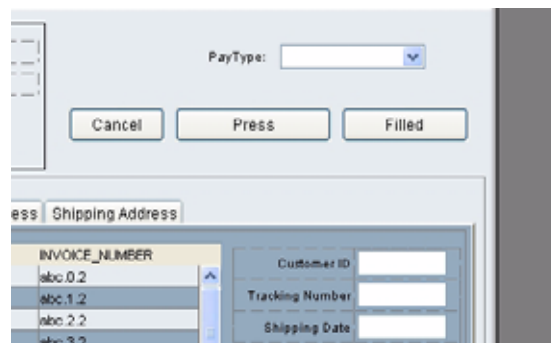
catch (Exception e) {
    view1.tabset1.view5.TRACKING_NUMBER =
    view1.tabset1.view5.getNextNum("TRACKING_NO");
    view1.tabset1.view5.SHIPPING_COST = 5.54;
    view1.tabset1.view5.SHIP_DATE = session.currentDate;
    orderFilledButton.visible = true;
}
}

```

6. コンテンツパネル > デザインタブで、**Cancel** と **Filled** ボタンの間にスペースを作り、下図のように新しいボタンを配置できる場所を確保します。



7. コンテンツパネル > デザインタブで、**Filled** ボタンを選択します。
8. ブラウザパネル > プロパティタブで、**Visual > Visible** チェックボックスをクリアします。
9. NXJ 開発ツールのコントロールパレットで、**ボタン** () をクリックします。
10. コンテンツパネル > デザインタブで、下図のように Cancel ボタンと Filled ボタンの間にボタンを配置します。



11. コンテンツパネル > デザインタブで、追加したボタンを選択します。
12. ブラウザパネル > プロパティタブで、以下の設定を行います。

Button > Title: Calc Shipping
General > Name: calcShippingButton
Interactive > Command: getShippingCost

-
13. NXJ 開発ツールメニューで、**ファイル > すべて保存** を選択します。

タスク 5: ビジネスサービスのテスト

Order Management にビジネスサービスを統合したので、次は、実行環境でこのサービスをテストします。注文管理プロセスにおいて、異なった権限をアサインされている異なったユーザで Order Management アプリケーションにログインします。

1. NXJ 開発ツールメニューで、**プロジェクト > すべて Make** を選択します。

2. NXJ 開発ツールメニューで、**プロジェクト > 配備** を選択します。

アプリケーションサーバが起動されていない場合、デフォルトのアプリケーションサーバを起動するかどうかを聞いてきますので、**はい** をクリックしてください。

実行時データソースの作成を尋ねられた場合、**はい** をクリックしてください。

しばらくすると、Unify NXJ はアプリケーションを構築して、配備を行います。

3. Web ブラウザで以下のサイトと通信します。

`http://localhost:8080/BPMAdmin/Login`

BPM のログインページが表示されます。

4. 以下の内容を入力して、ログインをクリックします。

ユーザ名 : dave

パスワード : dave

Dave のアクティビティリストが表示されますが、今は何もありません。

5. **個人ワークフローリスト** タブ上で、**個人ワークフローリストの編集** をクリックします。

これは、Dave がアクセスできるすべてのワークフローを表示します。

6. **未選択プロセス** の下にある **ordmgmt/OrdMgmt** をクリックします。

7. **追加** をクリックします。

8. **保存** をクリックして **完了** をクリックします。

Dave は、これで Order Management アプリケーションにアクセスできます。

9. **個人ワークフローリスト** 上で、**プロセス** の下の **ordmgmt/OrdMgmt** をクリックします。

Start_Order フォームが表示されます。

10. **Orders** タブ上の **New Order** をクリックします。

Order Management アプリケーションは、新規の注文を作成します。Shipping コストの欄は ¥0.00 である点に注意してください。

Finish Order

Order Complete Cancel

Process Information

CUST_CD: 0
Order_Number: 0
Process_Invoker: dave
Process_Title:

PayType: Purchase Order

Orders Customer Information Billing Address Shipping Address

ORDER_NUMBER	ORDER_DATE	INVOICE_NUMBER
400027	12/20/2005	300027
400028	12/20/2005	300028
400029	12/20/2005	300029
400030	12/20/2005	300030
400031	12/20/2005	300031
400032	12/21/2005	300032

Customer ID: 104120
Tracking Number: 0
Shipping Date:
Shipping Cost: \$0.00
New Order

Description Price

Order Number: 400033 Total: \$0.00

11. Product Description ドロップダウンリストから少なくとも1つの項目は選択します。
12. フォームの頭にある **Order Complete** ボタンをクリックします。
Start_Order フォームが閉じます。Dave が登録した注文は、次のアクティビティである Receive Order に受け渡されます。
13. ログアウトをして、次に以下の内容で再度ログインします。
ユーザ名: alice
パスワード: alice
14. アクティビティリスト タブをクリックして、名前 列の **ordmgmt/Receive Order** をクリックします。
Receive_Order フォームが表示されます。注文情報は Orders タブにあります。
15. **送信** をクリックします。
Receive_Order フォームは閉じます。Alice が承認した注文は、次のアクティビティである Fill Order に受け渡されます。
16. ログアウトして、再度次の内容でログインします。
ユーザ名: bob
パスワード: bob
17. アクティビティリスト タブ上の **名前** 列にある、**ordmgmt/Fill Order** をクリックします。
Fill_Order フォームが表示されます。

Process Information
 CUST_CD: 104120
 Order_Number: 400033
 Process_Invoker: DAVE
 Process_Title: Order # 400033

PayType: Purchase Order

Cancel
Calc Shipping

Orders | Customer Information | Billing Address | Shipping Address

ORDER_NUMBER	ORDER_DATE	INVOICE_NUMBER	
400033	12/21/2005	300033	

Customer ID: 104120
 Trading Number: 800131
 Shipping Date: 12/21/2005
 Shipping Cost: \$0.00

Description	Price
UNIFY NCI Platinum Level I Annual Support	200

Order Number: 400033
Total: \$200.00

18. **Calc Shipping** をクリックします。

ビジネスサービスが起動されて、注文の Shipping コストが計算されます。Calc Shipping ボタンが消えて、Filled ボタンが表示されます。

Process Information

CUST_CD: 104120
 Order Number: 400033
 Process_Invoice: D8VE
 Process_Title: Order # 400033

PayType: **Purchase Order**

Cancel Filled

Orders Customer Information Billing Address Shipping Address

ORDER_NUMBER	ORDER_DATE	INVOICE_NUMBER
400033	12/21/2005	300033

Customer ID: 104120
 Trading Number: 800131
 Shipping Date: 12/21/2005
 Shipping Cost: \$3.50

Description	Price
UNIFY NO Platinum Level 1 Annual Support	200

Order Number: 400033 Total: \$203.50

19. **Filled** をクリックします。

Fill_Order フォームは閉じられて、プロセスは Receive Payment アクティビティに移ります。

20. ログアウトをして次の内容で改めてログインします。

ユーザ名: charlie
 パスワード: charlie

21. アクティビティリスト タブの **名前** 列にある、**ordmgmt/Receive Payment** をクリックします。

Receive_Payment フォームが表示されます。注文情報は、Orders タブにあります。輸送コストは、Fill_Order フォームから持ち越された点に注意してください。

Process Information
CUST_CD: 104120
Order_Number: 400033
Process_Invoker: D:\...
Process_Title: Order # 400033

PayType: Purchase Order

Orders | Customer Information | Billing Address | Shipping Address

ORDER_NUMBER	ORDER_DATE	INVOICE_NUMBER
400033	12/21/2005	300033

Customer ID: 104120
Tracking Number: 000132
Shipping Date: 12/21/2005
Shipping Cost: \$3.50

Description	Price
UNIFY NO Platform Level 1 Annual Support	200

Order Number: 400033
Total: \$203.50

22. **完了**をクリックします。

Receive_Payment フォームを閉じられて、注文は完了します。

これで、Web サービスチュートリアルを終了します。