



*ActiveWorkflow*  
チュートリアル

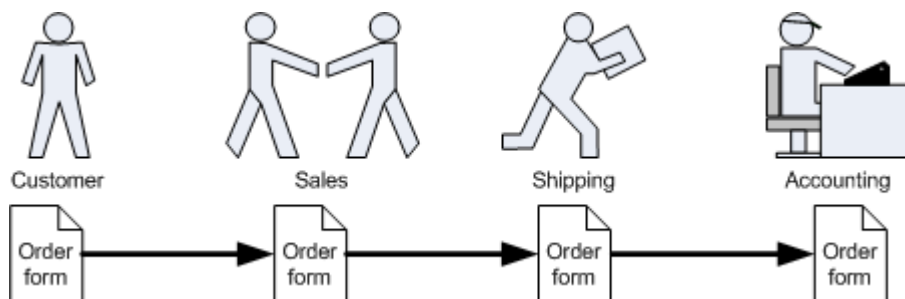
*Release12*

# はじめに

# 1

このチュートリアルでは、Unify NXJ を使用した定義済みのビジネスプロセスを基にして、自動化された order processing アプリケーションの開発、配備、実行を習得します。

このビジネスプロセスは、注文を受けて製品を出荷するまでの多くのビジネスで共通する仕事を表わしています。このアプリケーションを具体的に想像するために、印刷された注文書を想像してください。顧客が、この注文書に自分は誰で、何が欲しいのかを書き込むことからこのプロセスはスタートします。次に、会社の営業部門がこの注文書を受け取り、情報を確認してから、出荷部門に送ります。出荷部門では、出荷の準備をして、パッケージに郵送のトラッキング番号を割り当てます。最後に、この注文書は経理部門に渡し、代金の回収がされることとなります。



これらの手動によるステップを、一貫した自動ビジネスプロセスに変えるために、Unify NXJ のビジネスプロセス管理 ( BPM ) 機能を使います。最初に、ActiveWorkflow デザイナのグラフィカルモデリングツールを使って、プロセスのステップを定義します。各ステップでは、誰がそのステップを実行する権限を持ち、どのような情報が必要かを指定します。ActiveWorkflow デザイナは、その結果、プロセスを記述する XML 形式のファイルを作成します。

---

プロセスができ上がると、J2EE 環境の ActiveWorkflow エンジンに配備されます。ActiveWorkflow エンジンは、例えば、注文 1 件毎にあたる、プロセスインスタンスを管理して、注文が誰かの処理や承認を受ける状況になっていることを確実にその担当者に伝わるようにします。ログインをして自分のタスクリストを確認すると、自分の処理待ちになっている注文を確認することができます。例えば、営業部門の人であれば、承認待ちになっている新規の注文のリストを見ることができますが、既に承認されて出荷に送られた注文は、処理待ちリストには記載されていません。

# タスクの概要

## 2

このチュートリアルで実行するステップは、次のタスクにまとめられています。

- タスク 1: NXJ プロジェクトの作成
- タスク 2: ビジネスプロセスの定義
- タスク 3: アクティビティフォームの生成とプレビュー
- タスク 4: アクティビティフォームのカスタマイズ
- タスク 5: ActiveWeb コンポーネントの構築
- タスク 6: ActiveWeb コンポーネントをアクティビティフォームに配置
- タスク 7: アプリケーションの配備・実行

それぞれのタスクは、以下の章で詳細に説明されています。

サンプルアプリケーションの完成版は、Unify NXJ に含まれています。Unify NXJ 開発ツールから、以下のプロジェクトファイルを開くとアプリケーションの完成版を見ることができます。

```
<UNIFY_WORK>%projects%examples%ordmgmt%ordmgmt.prj
```

ここで、完成版のプロジェクトを開けば、タスク7までスキップすることも可能です。自分自身で構築する代わりに、この完成版のプロジェクトを利用する場合、“ MyBusinessProcess ”となっている部分はすべて“ ordmgmt ”になることに注意してください。

---

**注** – チュートリアルガイドにそって作成したプロジェクトと完成版プロジェクトでは、実装している機能が異なります。

---

<UNIFY\_WORK> ディレクトリは、Unify NXJ アプリケーションのファイルが保存されている場所です。これは、Unify NXJ がインストールされているディレクトリとは異なります。これらのディレクトリは、Unify NXJ がインストールされるときに決められます。デフォルトでは、<UNIFY\_WORK> は C:%Unify%NXJWork です。

---

Unify NXJ のインストールに関する詳細は、『 *Unify NXJ インストールガイド* 』を参照してください。

# タスク 1: NXJ プロジェクトの作成

---

3

NXJ プロジェクトには、ビジネスプロセス定義とその他のアプリケーションコンポーネントが保存されます。以下の手順に従って、サンプルアプリケーションの新規プロジェクトを作成します。

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。

サンプルアプリケーションでは、ActiveWorkflow デザイナと NXJ フォーム処理の Unify NXJ 開発ツールだけを使用します。

Unify NXJ 開発ツールが開き、サンプルプロジェクトが表示されます。このプロジェクトは、注文処理アプリケーションですが、Unify NXJ の ActiveWorkflow の機能を使用していません。これから作成する注文処理アプリケーションは、ActiveWorkflow の機能を使用します。

2. Unify NXJ 開発ツールメニューから、**ファイル > 新規プロジェクト** を選択します。

新規プロジェクトオプションダイアログが表示されます。

3. **新規プロジェクトオプションダイアログ**にて、次の値を入力あるいは選択をします。

名称： MyBusinessProcess

プロジェクトディレクトリ： C:\Unify\NXJWork\projects

新規プロジェクトの作成：（選択する）

ActiveWorkflow Jar ファイルをクラスパスに追加：（チェックする）

ActiveSOA Jar ファイルをクラスパスに追加：（チェックしない）

Workflow/SOA/Report Jar ファイルを EAR ファイルに含める：

（チェックする）

再利用機能を有効にする（チェックする）

ソース管理からプロジェクトを取り出す：（選択しない）

---

#### 4. 次へ ボタンをクリックします。

デフォルトデータベース接続ダイアログが表示されます。サンプルアプリケーションは、実行環境にてデータベースアクセスが必要となるので、Unify NXJ 開発ツール上においてもデータベースアクセスを必要とします。設計時にアプリケーションのデータソースと接続することで、Unify NXJ 開発ツールは、関連する ActiveWeb コントロールの生成および構成を自動化することができます。ActiveWeb コントロールは、追加プログラミングの必要のないサーバサイドデータベースのテーブル、もしくは ActiveWorkflow プロセスアクティビティオペランドのような、データソースエレメントに自動接続を提供する一組の詳細な data-aware なコントロールです。

#### 5. デフォルトデータベース接続ダイアログにて、次の値を入力あるいは選択します。

名称： OrdMgmtDB

タイプ： Other JDBC Database ( ドロップダウンリストから選択する )

Jar/Zip ファイル： C:\Unify\NXJ\lib\jdbcDrivers\Fositex.jar

( 選択ボタンをクリックして、Jar ファイルのリストにアクセスする )

ユーザ名： ( ブランクのまま )

パスワード： ( ブランクのまま )

実行時に使用するデータソース： OrdMgmtDB

ドライバ： com.inet.csv.CsvDriver

URL: jdbc:csv:C:\Unify\NXJWork\projects\examples\ordmgmt\database\DB

---

**注** - Jar/Zip ファイルと URL の上記の値は、Unify NXJ のデフォルトの場所を前提としています。(C:\Unify\) もし、Unify NXJ を別の場所にインストールした場合は、これらを適切な値に変更してください。

---

#### 6. ダイアログ上で、テスト ボタンをクリックします。

「接続できました。」のメッセージが表示されます。了解ボタンをクリックします。

接続に失敗した場合、ダイアログに入力した情報がステップ 5 の内容と一致するかを確認してください。

#### 7. 完了 ボタンをクリックします。

実行時データソースの作成ダイアログが表示され、このデータソースがアプリケーションサーバに登録されたことがわかります。

---

ほとんどのアプリケーションサーバにおいて、データソースの定義は面倒な作業ですが、このデータソース作成の自動化が、アプリケーション開発時間を効率化します。そして、アプリケーションの設計および開発において、自動化することができないタスクに集中することができます。

**8. 実行時データソースの作成ダイアログで、はい ボタンをクリックします。**

これで、NXJプロジェクトを作成して、そのデータベース接続の構成が完了しました。このアプリケーションでは、アクセスされるデータはすべて同じデータベースに存在します。NXJアプリケーションは、レガシーアプリケーションやXMLのようなカスタムデータソースおよび、複数のJDBCデータベースに同時にアクセスすることができます。

# タスク2: ビジネスプロセスの定義

---

4

ビジネスプロセスの定義は、以下のタスクから成り立っています。

サブタスク 1: アクティビティの定義

サブタスク 2: オペランドの定義

サブタスク 3: フロー制御エレメントの追加

サブタスク 4: インボイス送付アクティビティのポリシーの定義

## サブタスク 1: アクティビティと関連する スイムレーンの定義

アクティビティは、ビジネスプロセスの仕事の基本となる構成単位です。今回のプロセスは、5つのアクティビティを持っています。

- View Order
  - 一旦、注文が登録されると、顧客は注文番号で注文の状況をチェックすることができます。
- Receive Order
  - 注文が最初に登録された時に、営業担当はその注文を承認します。
- Fill Order
  - 営業により注文が承認された時に、出荷担当はその注文の出荷を行います。
- Send Invoice
  - 支払いが注文書ごとに行われる場合は、請求書を送付します。
- Receive Payment
  - 入金を確認されたら、経理担当は入金処理を行って注文を完了させます。

これらのアクティビティを定義するために、以下のことを行います。

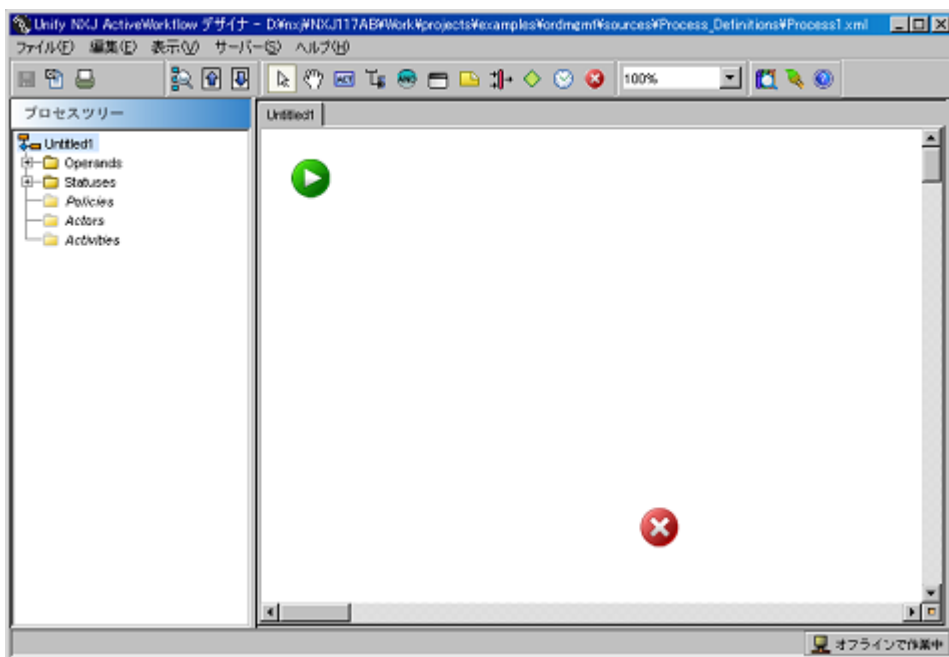
1. Unify NXJ 開発ツールのプロジェクトタブを開きます。

プロジェクトタブ上で、アプリケーションのコンポーネントの管理が一括して行えます。

2. プロジェクトタブで、**Process Definitions** フォルダを右クリックして**新規作成 > Process** を選択します。

"ActiveWorkflow エンジンの初期化中 ..." というポップアップメッセージがしばらく表示されます。ActiveWorkflow エンジンは、実行時のプロセスインスタンスを管理します。

ActiveWorkflow デザイナーのウィンドウが表示され、編集用の新規プロセスが表示されます。



ActiveWorkflow デザイナーのウィンドウは、左側のプロセスツリーと右側のプロセス図の2つのセクションに分割されています。プロセスツリーは、作成されるプロセスを階層形式で表示するもので、プロセス内のすべてのコンポ

---

ーメントを見ることができます。プロセス図は、プロセスをグラフィカルに表示し、標準的な WYSIWYG 技術を使用して、プロセス図を編集することができます。

新規のプロセスには、プロセスのスタート（緑の矢印）とストップ（赤の×）のアイコンが既に配置されています。

プロセスはデフォルトの名前で作成されるので、プロセスの名前を変更するには、Unify NXJ 開発ツールに戻る必要があります。

3. Unify NXJ 開発ツールのプロジェクトタブで、"**Process1**" エントリを右クリックしてメニューから名前の変更を選択します。
4. "**OrdMgmt**" を新しいプロセス名として入力します。

ActiveWorkflow デザイナに戻って、ビジネスプロセスの構築を行います。

5. ActiveWorkflow デザイナにおいても、プロセスツリーの "**Untitled1**" エントリを右クリックして編集を選択し、プロセスの名前の変更を行います。ラベルフィールドを "**OrdMgmt**" に変更して、**OK** ボタンをクリックします。

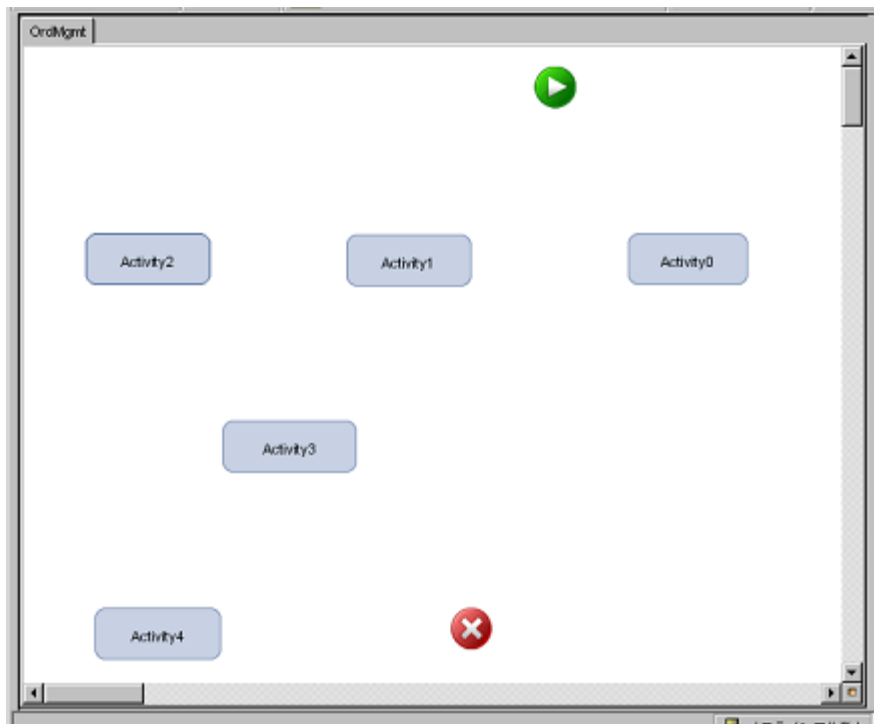
次に、図に 5 つのプロセスのアクティビティをそれぞれ追加します。

6. ActiveWorkflow デザイナツールバーで、**アクティビティの追加** ボタンをクリックして、プロセスレイアウトのスタートとストップアイコンの間をクリックして、アクティビティを配置します。



アクティビティの追加ボタン

7. ステップ 6 をあと 4 回繰り返して、合計 5 つのアクティビティを配置します。
8. アクティビティとスタート / ストップアイコンを調整して、次のような見た目になります。



アクティビティの配置場所はプロセスの実行順には関係なく、見た人の見易さの問題です。今回のプロセスは、上から下への流れです。

9. アクティビティをダブルクリックして、アクティビティの編集ダイアログを表示して、各アクティビティの構成変更をします。

アクティビティ	入力する情報
Activity0	ラベル : View Order
Activity1	ラベル : Receive Order
Activity2	ラベル : Fill Order
Activity3	ラベル : Send Invoice
Activity4	ラベル : Receive Payment

OK ボタンをクリックして、各ダイアログを終了します。

- 
10. ActiveWorkflow デザインツールバーで、**スイムレーンの追加** ボタンをクリックします。



スイムレーンの追加ボタン

これにより、スイムレーンオブジェクトをプロセス図内に配置することができます。

11. View Order アクティビティの左すぐ上をクリックして、そのまま左クリックボタンを保持したままドラッグして、スイムレーンが View Order アクティビティを囲うようになったところでクリックボタンを離します。



スイムレーンは、誰がそのアクティビティを実行するかを定義します。View Order アクティビティは、定義内容に従って、Customers のロールを持ったユーザだけが実効できます。

12. ステップ 10 と 11 を Receive Order 、 Fill Order 、 Receive Payment アクティビティに対して繰り返して行います。
13. 各スイムレーンをダブルクリックして、スイムレーンの編集ダイアログを表示します。各ダイアログで、以下の情報を入力します。

---

アクティビティ	スイムレーンの編集ダイアログに以下の情報を入力
View Order	アクターのロール : Initiative Participant タイプ : Group 値 : Customers
Receive Order	アクターのロール : Initiative Participant タイプ : Group 値 : Sales
Fill Order	アクターのロール : Initiative Participant タイプ : Group 値 : Shipping
Send Invoice	(スイムレーン無し)
Receive Payment	アクターのロール : Initiative Participant タイプ : Group 値 : Accounting

---

---

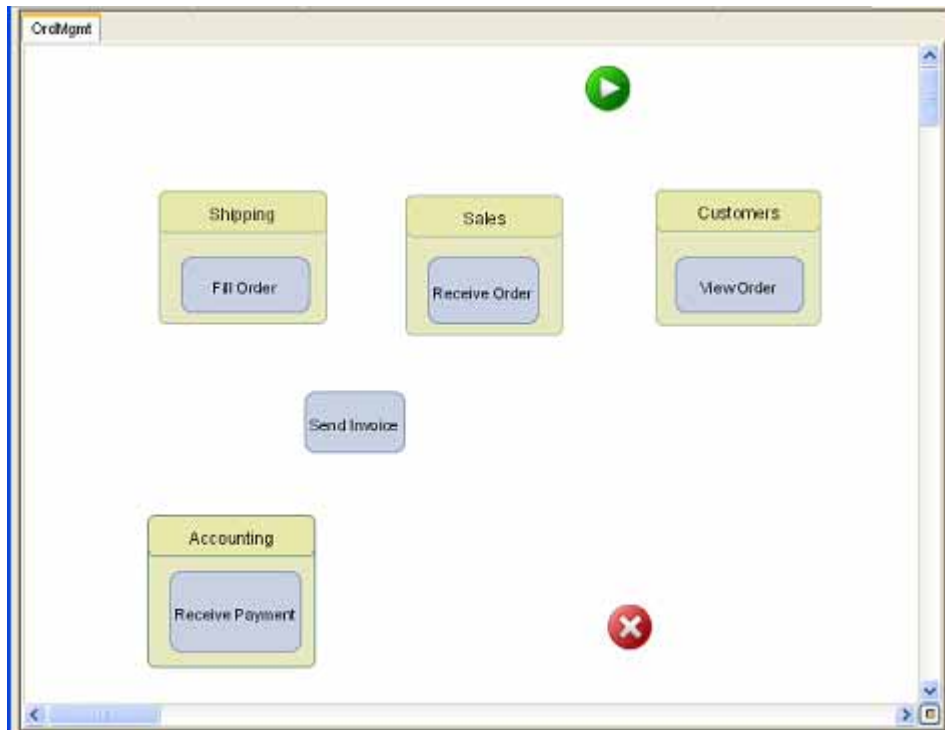
OK ボタンをクリックして、各ダイアログを終了します。

---

注 - 各アクティビティでは、タイプフィールドに "Group" を確実に設定してください。もし、タイプがデフォルトの "User" のままだと、実行時には、適切な権限が与えられていないために、ユーザがアクティビティフォームにアクセスできなくなります。

---

プロセス図は、以下のようにになります。



---

## サブタスク 2: オペランドの定義

アクティビティが完了したら、アクティビティが次のアクティビティに順に受け渡すデータが必要になります。今回の場合は、注文を入れた顧客の情報とユニークな注文番号がこのデータになります。プロセスでは、データは、オペランドを介して受け渡されます。アクティビティが利用するオペランドの定義は、以下のステップで行います。

1. プロセスツリーで、**Operands** フォルダを右クリックして、**新規オペランドの作成**を選択します。

2. **オペランドの編集**ダイアログで、以下の値を入力して、**OK** ボタンをクリックします。

名前 : CUST\_CD

データタイプ : Number

値 : 0

作成プロセス : OrdMgmt

スコープ : すべてのサブツリーに公開 (チェックする)

3. 前の 2 つのステップを繰り返して、次の 2 つのオペランドを作成します。

名前 : Order Number

データタイプ : Number

値 0

作成プロセス : OrdMgmt

スコープ : すべてのサブツリーに公開 (チェックする)

名前 : PayType

データタイプ : Picklist

デフォルト値 : PO

リスト値 : PO, CC

作成プロセス : OrdMgmt

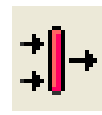
スコープ : すべてのサブツリーに公開 (チェックする)

## サブタスク 3:

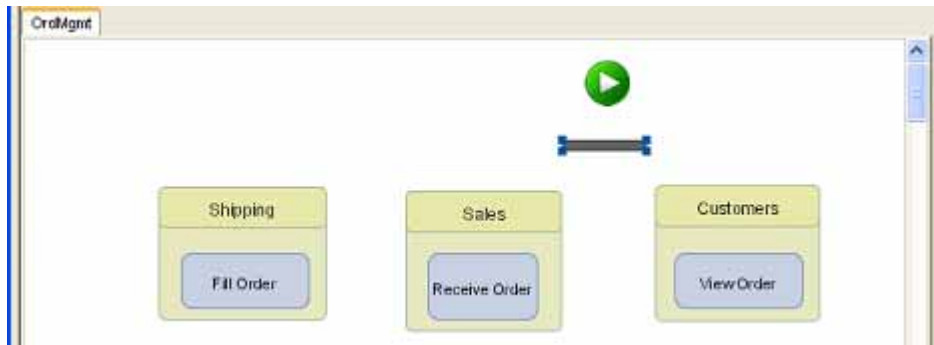
### フローコントロールエレメントの追加

フローコントロールは、アクティビティが発生する順番を指定します。例えば、Fill Order アクティビティは、Receive Order アクティビティが正しく終了した後に発生しなければなりません。接続ラインは、2つのアクティビティの順序付けるフローコントロールの1つです。ここで使用する他のフローコントロールは、結合 / 分岐バーの追加と決定アクティビティです。

1. ActiveWorkflow デザイナツールバーから、**結合 / 分岐バーの追加**をクリックしてからスタートバーの下をクリックして、バーを配置します。
2. バーを右クリックして、**縦 / 横を変更**を選択すると、バーは水平になります。



結合 / 分岐  
バーの追加  
ボタン

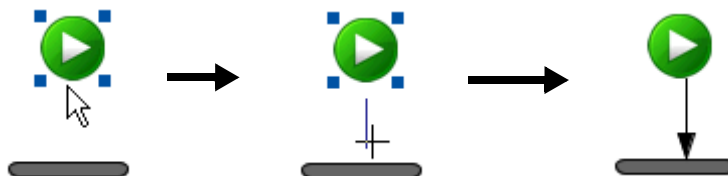


結合 / 分岐バーは、扇状に複数のプロセスへのパスを作成したり（分岐）、逆に複数のパスからひとつのパスに接続すること（結合）もできます。今回の例では、パスはプロセスの開始直後に分岐し、Receive Order、もしくは、View Order が最初のアクティビティとなります。

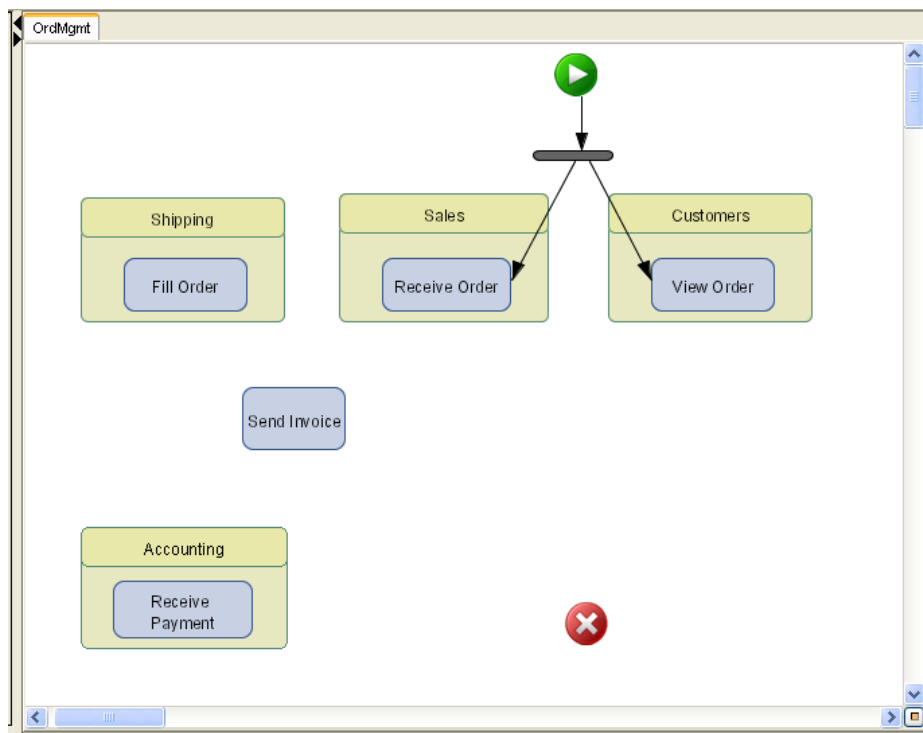
3. ActiveWorkflow デザイナーツールバーで、**接続ラインの追加**をまずクリックして、スタートアイコンから統合 / 分岐バーにドラッグすることで、スタートアイコンを統合 / 分岐バーに接続します。



接続ライン  
の追加  
ボタン



4. ステップ 3 の説明を参照して、以下の図に示されるように、統合 / 分岐バーを View Order アクティビティと Receive Order アクティビティに接続します。



**注** - 接続ラインをアクティビティにドラッグするときには、周りを囲むスィムレーションではなく、アクティビティのオブジェクトまでドラッグしてください。

---

接続ラインは、定義されたステータス値が必要です。ステータスは、ActiveWorkflow エンジンにアクティビティが終了したことを示すフラグにあたるものです。

5. プロセスツリーで、**Statuses** フォルダを右クリックして**新規ステータスの作成**を選択します。

6. **ステータスの編集**ダイアログで、名前フィールドに Approved と入力します。

ダイアログの他のフィールドに入っているデフォルトの値は、このビジネスプロセスではそのままでもかまいません。

7. **OK** ボタンをクリックします。

8. 前の 3 ステップを繰り返して、次のステータスを追加します。

Filled  
Received  
Sent

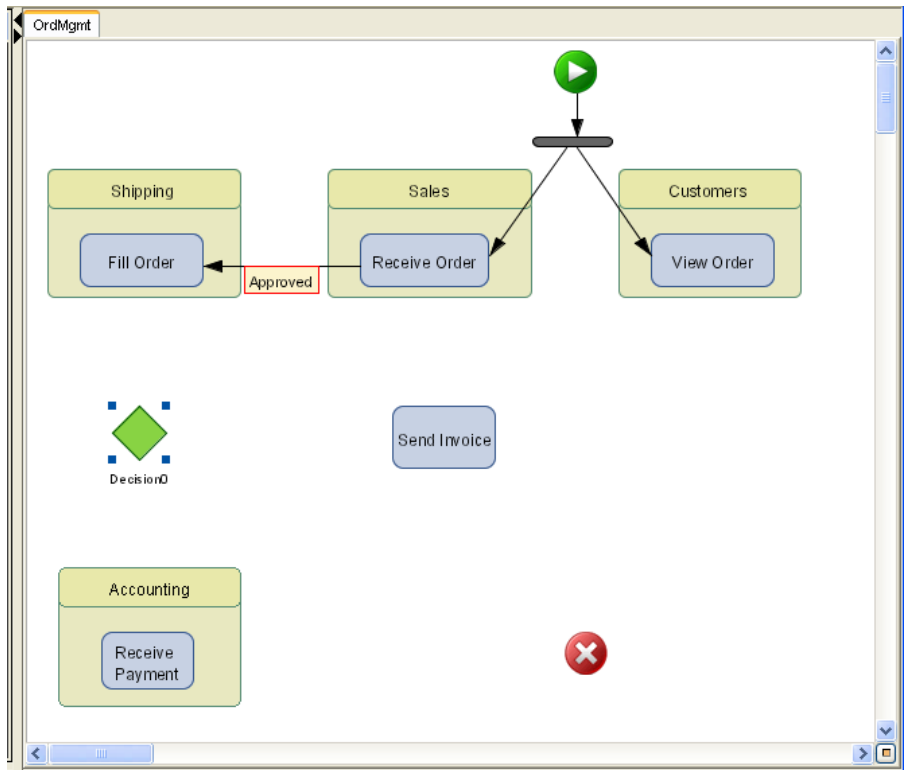
9. ステップ 3 に説明されたプロセスを使って、Receive Order アクティビティを Fill Order アクティビティに接続します。

10. 新規で追加された接続ラインを右クリックして、**ステータスの変更 > Approved** を選択します。

11. ActiveWorkflow デザイナーツールバーで、**決定アクティビティの追加** ボタンをクリックして、プロセスレイアウトをクリックして決定アクティビティを次の図に示された場所に配置します。



決定  
アクティビティ  
の追加ボタン



決定アクティビティコンポーネントは、2つのパスを選択するのに使用されます。決定アクティビティは、固有のロジック表記の設定の評価・判定によりパスを選択します。決定アクティビティに入ったときに、ActiveWorkflow エンジンが、次に行くパスを決定するアクティビティに定義されたロジック表記を判定します。

12. 決定アクティビティアイコンをダブルクリックして、Decision の編集ダイアログを表示します。ダイアログでは、次の情報を入力します。

ラベル : Credit Card または Purchase Order?

- 
13. Decision の編集ダイアログ上で、**追加** ボタンを押して、二値式の追加ダイアログを表示します。

この Decision ブロックでは、クレジットカード支払いの場合は請求書の送付が不要となるため、支払い方法がクレジットカードかどうかをプロセスが判断できなければなりません。( Send Invoice アクティビティを実行する必要がなくなります。)

14. 二値式の追加ダイアログで、以下の値を入力します。

ラベル : Credit Card

式 : PayType='CC'

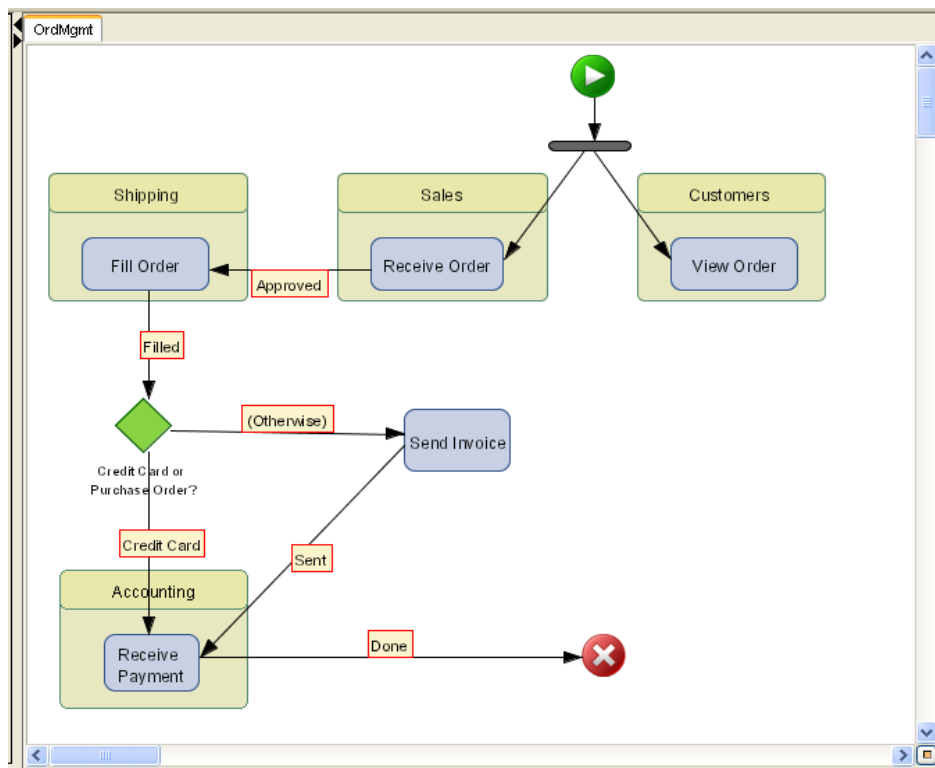
15. **OK** ボタンをクリックして、二値式の追加ダイアログを終了します。
16. Decision の編集ダイアログで、**OK** ボタンをクリックしてダイアログを終了します。
17. ステップ 3 に説明されたプロセスを使って、残りのアクティビティを以下のように接続します。

---

接続ラインを引く始めのアクティビティ	接続ラインを繋ぐ アクティビティ またはアイコン	セットするステータス (右クリックでキーワードを選択)
Fill Order	Decision Node	Filled
Decision Node	Send Invoice	Otherwise
Decision Node	Receive Payment	Credit Card
Send Invoice	Receive Payment	Sent
Receive Payment	Stop	Done

---

18. 以下に示されるように、プロセスが、アクティビティ、統合 / 分岐バー、スイムレーン、接続ライン、ステータスが定義されているかを確認します。



オブジェクト間の配置は、図に一致していなくても構いません。すべての要素があり、名前が正しければビジネスプロセスは正しく定義されています。

---

## サブタスク 4:

### Send Invoice アクティビティのポリシー定義

ポリシーは、事前に定義されたイベントの発生に基づいていくつかの定義されたアクションを起動するのに使われます。ポリシーは、イベントとアクションの二つの部分から成っています。イベントは、アクションを実行するポリシーのトリガとなるものです。今回定義する Generate Invoice ポリシーは、支払い方法が Purchase Order の場合に、請求書を作成、送付を行います。

1. プロセスツリーで、アクティビティフォルダを開いて、Send Invoice アクティビティを開きます。

2. **Policies** フォルダを右クリックして、**新規ポリシーの作成**を選択します。

3. ポリシーの編集ダイアログで、以下の値を入力あるいは選択をします。

イベント：（選択） ProcessStarted

アクション：（選択） AddStatusToProcess

説明：（自動的に他のダイアログの値に基づいて設定される）

プロセス：（選択） OrdMgmt/Send Invoice

ステータス：（選択） OrdMgmt /Sent

ポリシーラベル：Generate Invoice

4. **OK** ボタンをクリックして、ポリシー情報を保存して、ダイアログを終了します。

5. **ファイル > 保存** を選択します。

プロセス定義を終了したので、ActiveWorkflow デザイナを最小化するか閉じることができます。

# タスク 3: アクティビティフォームの 生成とプレビュー

5

ビジネスプロセスの定義が完了したので、次に、アクティビティフォームと呼ばれるプロセスへのユーザインタフェースの構成 / 作成を Unify NXJ 開発ツールを使って行います。Unify NXJ が高い生産性を実現する特長の 1 つが、様々なデータソースから関連するデータで、迅速にリッチで機能的なユーザインタフェースをビジネスプロセスに付けられることです。ここで言うデータソースには、データベース、Web サービス、メインフレーム上の基幹系アプリケーション等が含まれます。このタスクで見られるように、このアプリケーションのフロントエンドのフレームワークは、自動的に生成され、操作可能なプロセスへのインタフェースになっています。

以下のタスクでは、これらのフォームにデータベースに割当てられたフィールドを追加します。NXJ アプリケーションでは、フィールドは、アプリケーションデータに割当てても可能です。

このタスクは、以下のサブタスクを持っています。

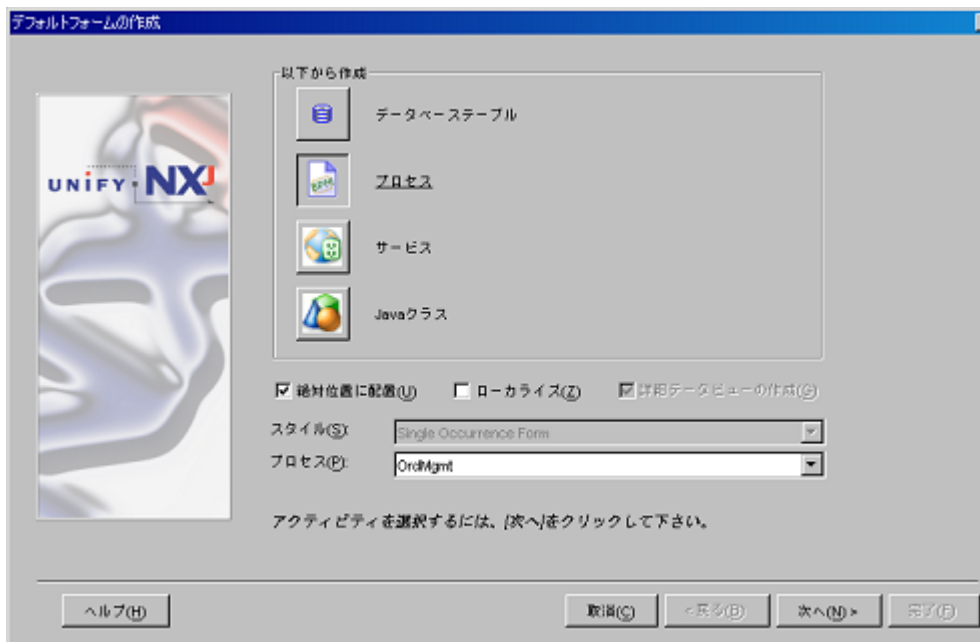
- サブタスク 1: デフォルトアクティビティフォームの生成
- サブタスク 2: アクティビティフォームをプロセスにマップ
- サブタスク 3: 実行環境におけるデフォルトフォームの表示
- サブタスク 4: OrdMgmt プロセスを開始し、Start\_Order フォームを使用して注文を入力

## サブタスク 1: デフォルトアクティビティフォームの生成

以下のステップでアクティビティフォームを生成します。

1. Unify NXJ 開発ツールで、**ファイル > デフォルトフォームの作成** を選択します。

デフォルトフォームの作成ウィザードが表示されます。



2. **プロセス** ボタンをクリックします。
3. OrdMgmt がプロセスフィールドに表示されていることを確認します。  
パネル上の残ったオプションのデフォルト設定は、このアプリケーションには適切なものになっています。
4. **次へ** ボタンをクリックします。
5. アクティビティパネルで、以下のそれぞれのアクティビティの作成チェックボックスをチェックします。

<Start>  
Fill Order  
Receive Order  
Receive Payment  
View Order

---

これらのアクティビティはそれぞれが関連付けられたアクティビティフォームを持つこととなります。Send Invoice アクティビティは、ポリシーで定義されたルールによって管理され実行されるので、アクティビティフォームを持つ必要はありません。

6. **完了** ボタンをクリックします。

アクティビティフォームが生成されて、Classes フォルダの下に追加されます。

7. 生成されたアクティビティフォームを見る場合、プロジェクトタブで Classes フォルダを開きます。アクティビティフォームをクリックして、その内容を見ます。

フォームレイアウトが表示されます。各アクティビティフォームクラスは、JSP 表記（JSP タブで見ることができます）と必要に応じてアプリケーションソースコードを追加できるフレームワーク（ソースタブで見ることができます）と EJB 版のクラス（アプリケーション完了後、EJB タブで見ることができます）から成り立っています。Unify NXJ 開発ツールは、フォームレイアウトとソースプログラミングを元に JSP コードを自動生成します。

OrdMgmt フォームは、デフォルトで名づけられます。それはプロセスのスタートを表わしており、必要に応じて名前を変更してください。

8. OrdMgmt エントリを右クリックして、**名前の変更** を選択します。

名前を変更すると、オブジェクト名の変更確認のメッセージボックスが表示されます。**はい** ボタンをクリックして名前の変更を承認します。

この確認メッセージは、Unify NXJ の再利用機能に関係しています。この機能は、大規模な Web アプリケーション開発の生産性を極端に向上できるものです。基本的に、再利用とはクラス構造の階層化を意味します。Java のクラス階層と同様に、NXJ のサブクラスはその親から機能を継承し、必要に応じてオーバーライドできます。このメッセージは、開発者に親クラスの名前の変更は、そのサブクラスへの継承を壊してしまうことを、警告しているものです。再利用機能については後ろのタスクで学習します。

9. Start\_Order と入力して、変更確認メッセージボックスで **はい** ボタンをクリックします。

---

## サブタスク 2:

### アクティビティフォームをプロセスにマップ

このタスクでは、プロセスデザインを変更して、アクティビティフォームを関連するプロセスアクティビティに紐付けます。

以下のステップで、関連付けを行います。

1. Unify NXJ 開発ツールのプロジェクトタブで、Process Definitions フォルダにある OrdMgmt プロセスエントリをクリックします。  
プロセスが、ActiveWorkflow デザイナ上に開かれます。
2. プロセスダイアグラム上で、スタートアイコンを右クリックして**プロセスの編集**を選択します。
3. プロセスの編集ダイアログで、以下の値を入力あるいは選択します。  
ラベル : OrdMgmt  
説明 : ( ブランクのまま )  
Activity JSP の使用 : このオプションをチェックして、ドロップダウンリストから "Start\_Order" を選択します。
4. **OK** ボタンをクリックして、変更を保存し、ダイアログを終了します。
5. プロセスダイアグラム上で、以下にリストされた各アクティビティを右クリックして、それぞれのアクティビティの編集ダイアログにアクセスして、関連付けるフォームの名前を以下のように設定します。

---

#### アクティビティの編集ダイアログ

**編集するアクティビティ :**    **ドロップダウンリストから以下を選択 :**

View Order	Activity JSP の使用 : View_Order
Receive Order	Activity JSP の使用 : Receive_Order
Fill Order	Activity JSP の使用 : Fill_Order
Receive Payment	Activity JSP の使用 : Receive_Payment

6. **ファイル > 保存** を選択します。

定義したプロセスが ActiveWorkflow エンジンで利用可能になります。

---

## サブタスク 3:

### 実行環境におけるデフォルトフォームの表示

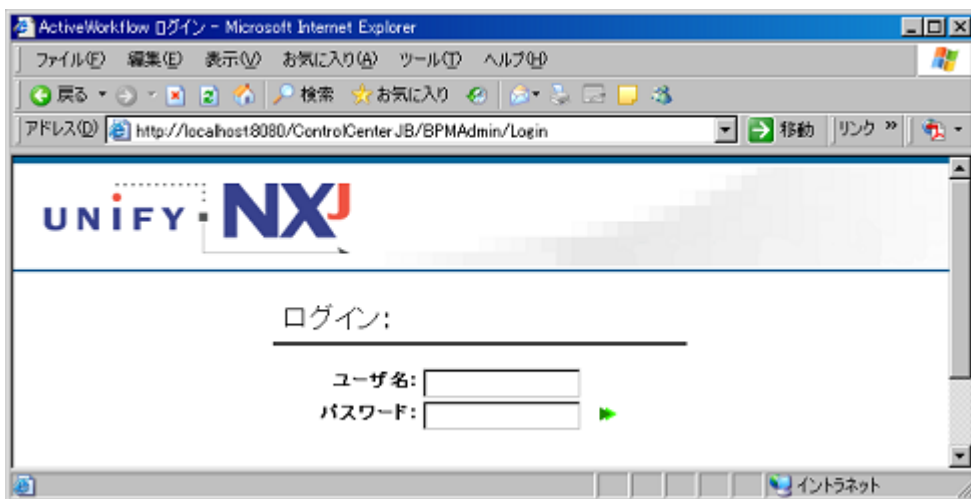
デフォルトアクティビティフォームでも ActiveWeb コンポーネントおよび現実の注文処理アプリケーションを構成するその他の機能強化が必要となります。しかし、デフォルトアクティビティフォームは、ユーザが OrdMgmt ビジネスプロセスを起動して利用する、すなわち、新規の注文を作成してプロセスの中をそれをどのように動かすのかを見してみるための機能を持っています。

以下のステップでアプリケーションを配備し実行します。

1. Unify NXJ 開発ツールメニューで、**プロジェクト > 配備** を選択する。  
しばらくすると、アプリケーションが構築されて、アプリケーションサーバに配備されます。
2. Web ブラウザから次の URL を表示します。

**<http://localhost:8080/ControlCenter/BPMAdmin/Login>**

ActiveWorkflow 管理者のログインページが表示されます。



## サブタスク 4: OrdMgmt プロセスを開始し Start\_Order フォームを使用して注文を入力

完成したアプリケーションを一通り見るためには、ロールプレイをする必要があります。顧客として、注文を入力してから、OrdMgmt プロセスの流れに従って、営業、出荷、経理の各グループのメンバとしてプロセスを進めます。

1. ActiveWorkflow のログインページで、次の値を入力してログイン ボタンをクリックします。

ユーザ名 : dave  
パスワード : dave

Dave はデモ組織のメンバです。このユーザおよび組織は、NXJ で事前に設定されています。

ActiveWorkflow 管理者が表示されます。

2. 個人ワークフローリスト タブをクリックします。

Dave のプロセスリストが表示されます。これは、Dave が起動することができるプロセスのリストです。OrdMgmt プロセスによるリストを生成することになります。

3. ページの左下にある **個人ワークフローリストの編集** のリンクをクリックします。

Dave が起動する権限のあるプロセスが、次のステップのスクリーンショットに見られるように個人ワークフローリストの未選択プロセス列に表示されます。

4. **MyBusinessProcess/OrdMgmt** エントリを選択して、**追加** ボタンを押下すことで、未選択プロセス列から選択済みプロセス列に移します。

個人ワークフローリストの編集

未選択プロセス

- Audit Trail Install
- Automated Activity
- Business Rule
- External Process
- FinishedProcessCleaner
- Leave Request
- Looping Example
- Meeting Reminder
- MyBusinessProcess/OrdMgmt**
- New Project Request

追加 >

< 削除

選択済みプロセス

上へ

下へ

上記はワークフローリストに追加可能なプロセスの一覧です。上記のプロセスがワークフローリストに、この順番 [追加]、[削除] ボタンを使って、リスト間でプロセスを移動して表示されます。表示順序は、[上へ]、[下へ] ボタンで変更することが出来ます。

各ページに表示するプロセス数 10

保存 完了

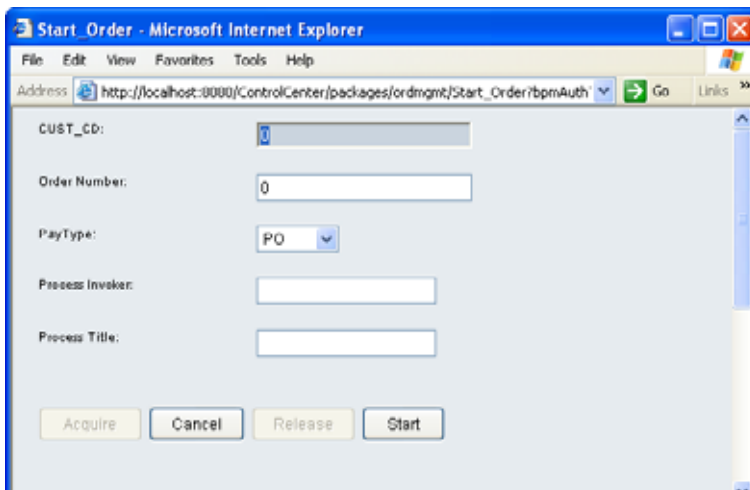
MyBusinessProcess/OrdMgmt プロセスが未選択プロセス列に無い場合、サブタスク 3 のプロジェクトの配備が行われたかを確認する必要があります。

5. **保存** ボタンをクリックします。
6. **完了** ボタンをクリックします。

個人ワークフローリストページに、MyBusinessProcess/OrdMgmt プロセスが含まれます。

7. 個人ワークフローリストの **MyBusinessProcess/OrdMgmt** リンクをクリックします。

新たに作成された Start\_Order フォームが、新しいブラウザウィンドウに表示されます。



The screenshot shows a web browser window titled "Start\_Order - Microsoft Internet Explorer". The address bar displays "http://localhost:8000/ControlCenter/packages/ordmgmt/Start\_Order/bpmAuth". The form contains the following fields and controls:

- CUST\_CD: Text input field containing "0".
- Order Number: Text input field containing "0".
- PayType: Dropdown menu with "PO" selected.
- Process Invoker: Empty text input field.
- Process Title: Empty text input field.
- Buttons: "Acquire", "Cancel", "Release", and "Start".

このフォームから、ユーザは注文プロセスを起動することができます。フォームには、注文明細やそのコストのようなデータは事前には作成されていませんが、ActiveWorkflow プロセスのデフォルトの値が入っています。

**8. フォームに次の値を入力します。**

CUST\_CD = 104120  
Order\_Number = 99999  
PayType = PO  
Process\_Invoker= dave  
Process\_Title = ORDER 99999

**9. Start ボタンをクリックします。**

Start\_Order フォームは閉じられて、プロセスは最初のアクティビティ View Order に進みます。

ユーザが対応する必要のあるアクティビティは、ユーザのアクティビティリストに表示されます。ユーザが、自分が必要とするアクティビティがなくなった場合、アクティビティリストからアクティビティはなくなります。それは、アクティビティが完了したか、アクティビティが他のユーザに渡された場合です。

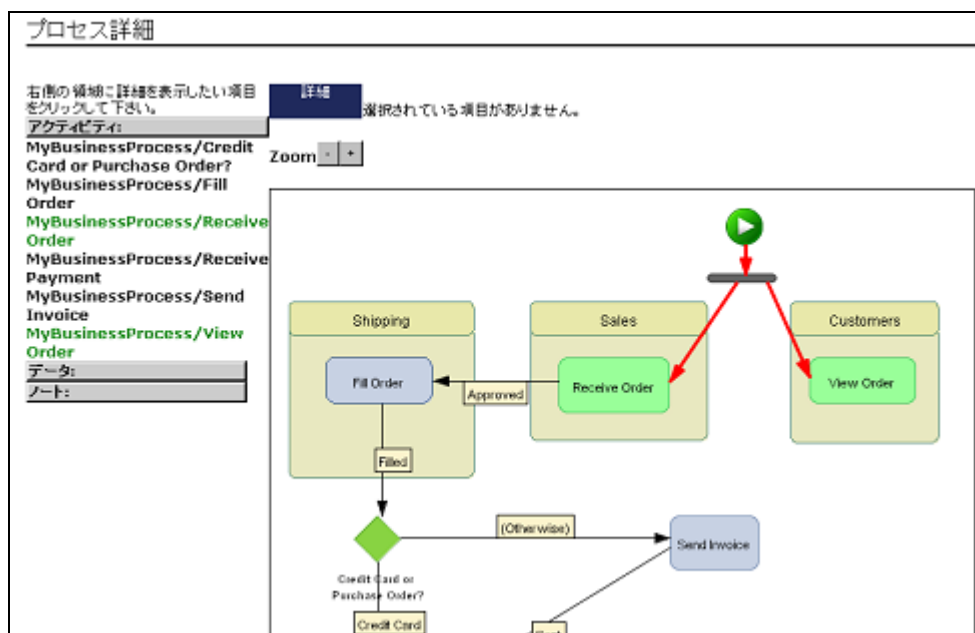
## 10. アクティビティリスト タブをクリックします。

Dave が実行することができるアクティビティがリストに表示されます。この場合、View Order アクティビティです。プロセスのインスタンスが起動された時間などの関連する情報もそこには表示されています。

## 11. 表示ドロップダウンリストから、完了アクティビティを選択します。

## 12. "ORDER 99999" エントリをインスタンス列から選択します。

プロセスの現在の状況が表示されます。プロセスの現在のアクティビティ、つまり、注文がどこにあるかは緑色で表示されているので、簡単に判別することができます。この場合、Dave が今起動した注文は、現在 Receive Order アクティビティにあることが分かります。（View Order アクティビティは、Customers グループのユーザに対して、常にカレントアクティビティになります。）



プロセスを Fill Order アクティビティに進めるには、Sales グループのユーザが注文を承認する必要があります。

13. ログアウトをクリックします。

ログアウトは、プロセス管理の右上にあります。NXJ は、Dave をログアウトして、ActiveWorkflow ログインページを表示します。

14. "alice"/"alice" を入力します。

Sales グループの Alice でログインしました。

15. アクティビティリスト タブをクリックします。

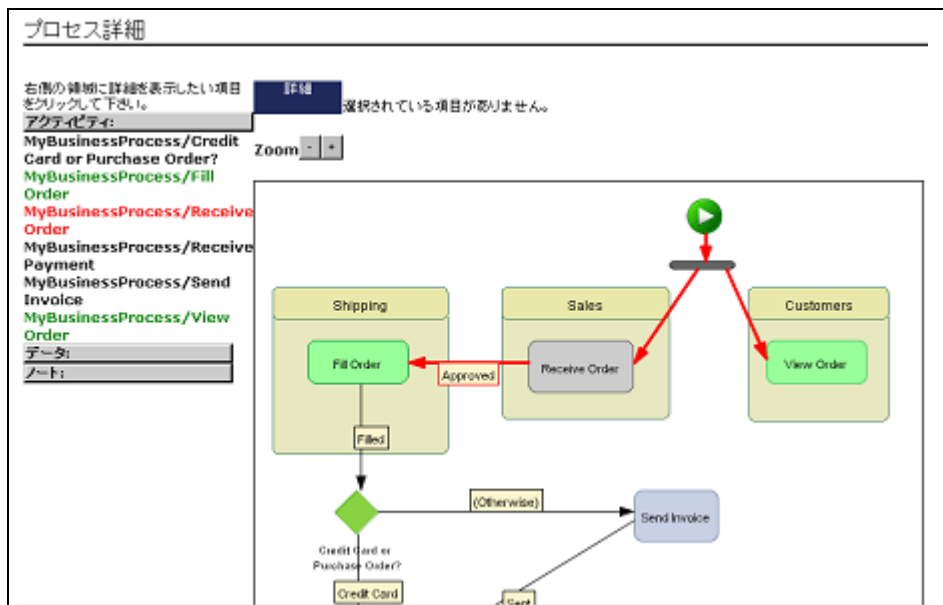
Alice のアクティビティリストには、Receive Order アクティビティが生成されています。

16. Receive Order リンクをクリックします。

Receive Order フォームには、Start\_Order フォームで入力した内容が表示されます。

17. Approved ボタンをクリックします。

Receive Order フォームは閉じられ、注文は Shipping グループの Fill Order アクティビティに移動します。Alice は、以下のプロセスのビューを見ることができます。

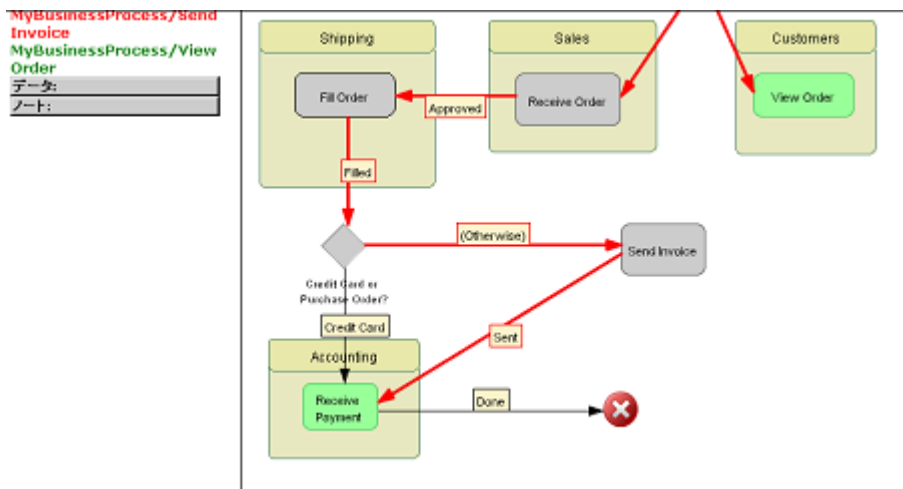


---

Alice は既にタスクを終了したので、Receive Order アクティビティはもはやリストには存在しません。表示ドロップダウンリストで完了アクティビティを選択すると、既に完了したアクティビティを見ることができます。

18. ログアウトして、今度は Shipping グループのユーザ **"bob" / "bob"** でログインします。
19. アクティビティリストタブをクリックします。
20. アクティビティリストの **Fill Order** リンクをクリックします。  
Fill Order フォームが開きます。
21. **Filled** ボタンをクリックして、注文を次のアクティビティに進めます。  
この場合、注文が PO 形式なので、注文は Send Invoice アクティビティに行きます。Send Invoice アクティビティは、自動的に請求書を送って、プロセスは Receive Payment アクティビティに進みます。
22. ログアウトして、Accounting グループのユーザ **"charlie" / "charlie"** でログインします。
23. アクティビティリストタブをクリックします。
24. アクティビティリストのインスタンス列にある **"Order 99999"** リンクをクリックします。

プロセスダイアグラムは、注文が Receive Payment アクティビティにあることを示します



25. Web ブラウザのウィンドウを閉じてアクティビティリストに戻り、Receive Payment アクティビティをクリックします。

26. Done ボタンをクリックします。

Order99999 のプロセスは、現在完了して、保存されています。

# タスク4: アクティビティ フォームのカスタマイズ

6

デフォルトのアクティビティフォームの設計は、エンハンスを行う必要があります。それらのフォームは、注文の内容や出荷先のような注文を処理するために必要な情報をまだ持っていません。

このタスクは、以下のサブタスクから成ります。

サブタスク 1: Start\_Order フォームのカスタマイズ

サブタスク 2: View\_Order フォームのカスタマイズ

サブタスク 3: Receive\_Order フォームのカスタマイズ

サブタスク 4: Fill\_Order フォームのカスタマイズ

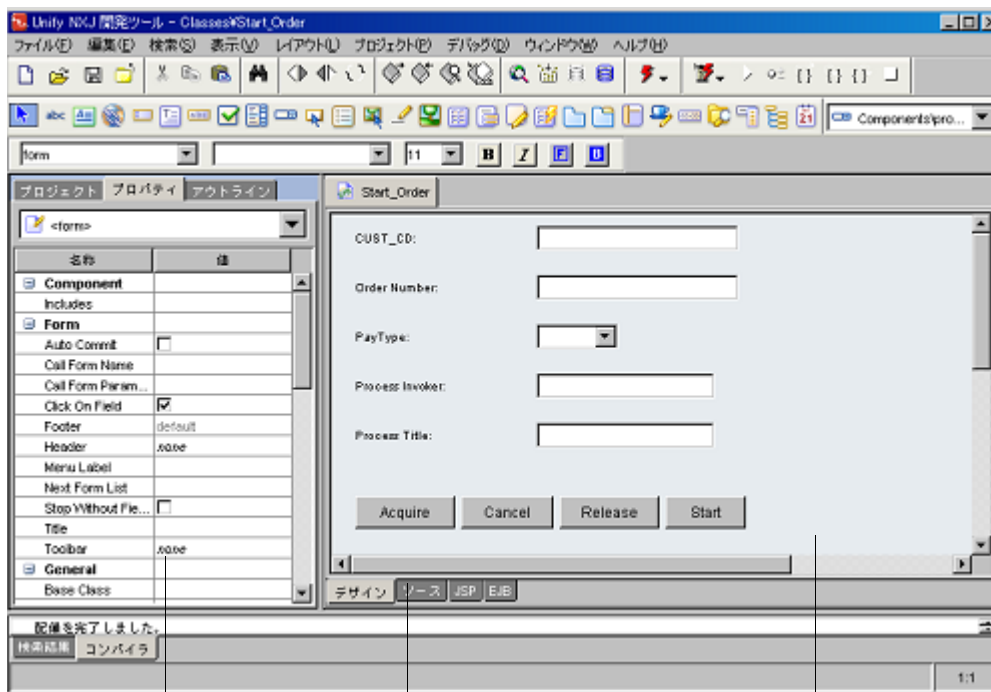
サブタスク 5: Receive\_Payment フォームのカスタマイズ

このタスクの PDF ファイルからコードを直接コピーすることができます。コード内容を正しく NXJ スクリプトエディタに貼り付けできない場合は、それらをまずノードパッドに貼り付けて特殊文字を削除してください。正しい内容になったテキストを NXJ スクリプトエディタに貼り付けてください。テキスト内のスペースと位置はコンパイルおよび実行にはなんの影響もありません。

## サブタスク 1: Start\_Order フォームのカスタマイズ

1. Unify NXJ 開発ツールのプロジェクトタブにて、Start\_Order エントリをクリックします。

Start\_Order フォームレイアウトがデザインパネルに表示されます。



デザインパネルで選択されているオブジェクトに適用されているプロパティ

ソースタブをクリックして、フォームあるいはコンポーネントのスク립トを表示

フォームあるいはコンポーネントのレイアウト

ActiveWorkflow デザイナのプロセスアクティビティパネルと同様に、Unify NXJ 開発ツールのデザインパネルは、標準的な WYSIWYG 技術を使用してフォームレイアウトを見ることができます。

必要に応じて、レイアウトパネル上のオブジェクトをグリッドに整列することができます。グリッド表示をオンにするには、**編集 > オプション > レイアウト** を選択して、“**Web レイアウトにグリッドを表示**” チェックボックスをチェックします。

この例のような自動生成されたフォーム上のアクティビティの処理を進めるためのラベル、フィールド、ボタンは関連するプロセス定義を基に作成されています。PayType フィールド以外のフィールドは、実行時に自動的に値が

---

設定されます。値は、プロジェクトのデータベース接続で定義したリレーショナルデータベースからのものです。読み取り専用の値であれば、そのフィールドは表示専用のフィールドとして構成されます。表示専用フィールドは、実行時にカーソルは止まりません。

フォーム上のフィールドを構成するには、Unify NXJ 開発ツールのプロパティタブを使用して、デフォルトのプロパティをカスタマイズします。

2. **CUST\_CD**、**Order\_Number**、**Process\_Invoker**、**Process\_Title** フィールドを同時に選択します。

デザインパネルで複数のオブジェクトを同時に選択するには、Ctrl あるいは Shift を押しながらクリックします。ラベルではなく、必ずフィールドを選択してください。

3. プロパティタブにて、以下のプロパティを設定します。

Stop For Input: チェックをしない

Updateable: チェックをしない

Background: 透過を選択

( **選択** ボタンをクリックして、**透過** ボタンをクリックし、**RGB** タブをクリックして **OK** ボタンをクリックします。 )

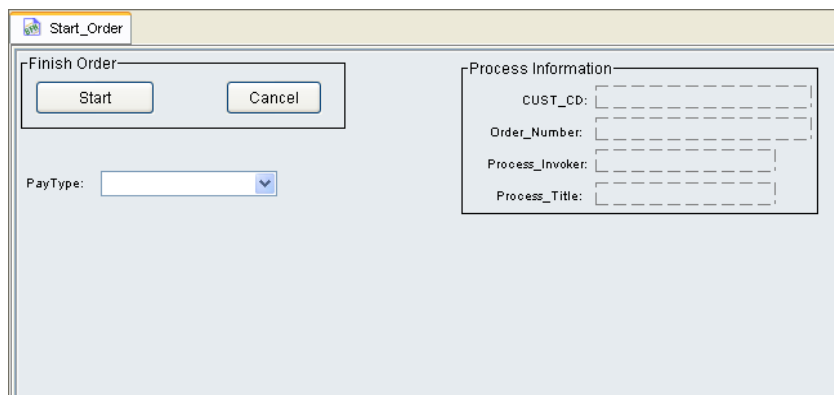
Border Style: None を選択

4. **Acquire** と **Release** ボタンを同時に選択して、**Delete** キーを押下します。


Acquire あるいは Release ボタンを使うと、そのグループの特定のユーザが特定の注文のアクティビティを行っている間は、そのアクティビティをロック ( Acquire ) したり、アンロック ( Release ) したりできます。ロックされたアクティビティは、そのグループの他のユーザのアクティビティリストには表示されなくなります。

Acquire/Release 動作をサポートするには、それぞれのアクティビティのスイムレーンプロパティに、現在入っている値とは異なる値を設定する必要があります。"アクターのロール" フィールドには、"Initiative Participant" の代わりに "Eligible Participant" が指定される必要があります。このアプリケーションではこの機能は使用しないので、アクティビティフォームのこれら 2 つのボタンは必要ありません。

5. これらのオブジェクトの配置を変更して、好きな Look and Feel に変更することができます。ここでは、次のように変更します。



以下の手順で、フィールドとボタンをボックスにグルーピングします。

- a. Unify NXJ 開発ツールのツールバーで、**Box** ボタンをクリックします。  
ツールバーで選択されたオブジェクトが、デザインパネル上でフォームレイアウトをクリックしたところに配置されます。  
 Box ボタン
- b. 該当のフィールドとラベルを囲うように左上からクリック&ドラッグをしてボックスを配置します。  
ボックスは、デフォルトのタイトル“ Legend ”で配置されます。
- c. ボックスコントロールはまだ選択されているので、Title プロパティを例えば、“ Process Information ”に変更します。  
Title プロパティは、プロパティの Visual グループにあります。

6. Start ボタンを選択して、Title プロパティを Order Complete に変更します。  
必要に応じて、ボタンのサイズを変更して、新しいテキストが一行に収まるようにしてください。

Start ボタンを使って、ユーザは注文情報がフォームに集約されて、注文プロセスを起動することができます。Cancel ボタンは、現在の注文と関連するプロセスインスタンスをキャンセルします。

---

Process\_Invoker フィールドは、注文プロセスを誰が起動したかを設定するために必要です。これを使って、フォームにビジネスロジックを追加することもできます。このコードは、ActiveWorkflow エンジンからユーザ名を取ってきて、その値を Process\_Invoker フィールドに設定します。

7. 以下のステップでコードを追加します。
  - a. ソース タブをクリックして、フォームのコードテンプレートにアクセスします。
  - b. 左括弧のすぐ下に以下のコードを入れます。

```
BPMProxy bpm = (BPMProxy)session.createBPMProxy();
AFTER FIND
{
    box1.Process_Invoker = bpm.getUser();
}
```

8. プロジェクト > 変更分のみ Make を選択して、スクリプトをコンパイルします。

このステップで、スクリプトにシンタックスエラーがないことを確認します。

変更分の Make ダイアログが表示されて、コンパイルが終了したときに消えます。Make がエラーなく終了したメッセージが、Unify NXJ 開発ツールの下部のコンパイラ領域に表示されるはずですが、Make 作業の結果として、EAR ファイルが生成され、任意の J2EE アプリケーションサーバに配備することができます。

コンパイラ領域にエラーの発生が表示されたら、該当するコードの確認を行ってください。

これにより、Start\_Order フォームは完成です。

これと同様のカスタマイズが他のアクティビティフォームにも必要になります。このフォームの部品を他のフォームにコピーするために、この Start\_Order フォームを開いたままにしておきます。

## サブタスク 2: View\_Order フォームのカスタマイズ

以下の手順で View\_Order フォームをカスタマイズします。

1. プロジェクトタブにて、View\_Order フォーム名をクリックします。

デザインパネルにフォームのレイアウトが表示されます。

---

Start\_Order フォームから Process\_Information ボックスをコピーして、同じ名前のフィールドと置き換えて貼り付けを後程行います。

2. CUST\_CD、Order\_Number、Process\_Invoker、Process\_Title のラベルとフィールドを削除します。

フォームレイアウトからオブジェクトを削除するには、選択して DELETE キーを押下すると削除できます。Ctrl + Z を押すと削除したものを元に戻すことができます。

隣接するオブジェクトをまとめて選択するときは、マウスの範囲指定で行うことができます。

3. Start\_Order タブをクリックして、そのフォームのレイアウトをデザインタブの前面に持ってきます。

4. Process Information ボックスをコピーします。

ボックスをコピーするには、Start\_Order フォームレイアウトの “ Process Information ” ボックスの境界線を右クリックしてからコピーします。

5. View\_Order タブをクリックして、View\_Order フォームレイアウトを前面に持ってきます。

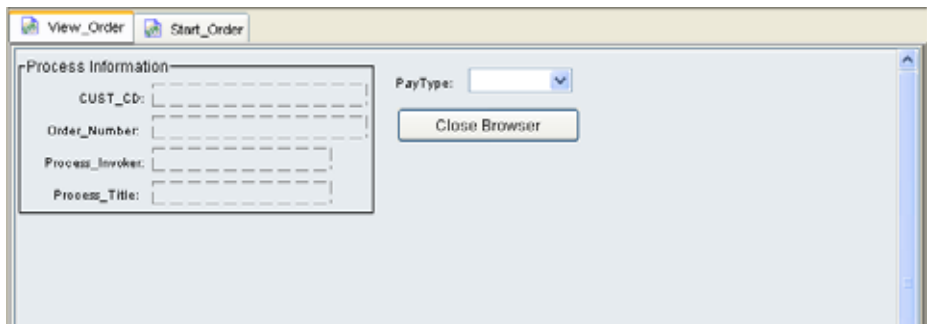
6. Process\_Information ボックスを貼り付けます。

7. "Finished" ボタンを View\_Order フォームレイアウトから削除します。

このアクティビティは、顧客から見えるアクティビティなので、Customers グループのユーザがプロセスを終了させることができはいけません。

8. Cancel ボタンを選択して、Title プロパティを Close Browser に変更します。

完了したフォームは、以下のようになります。



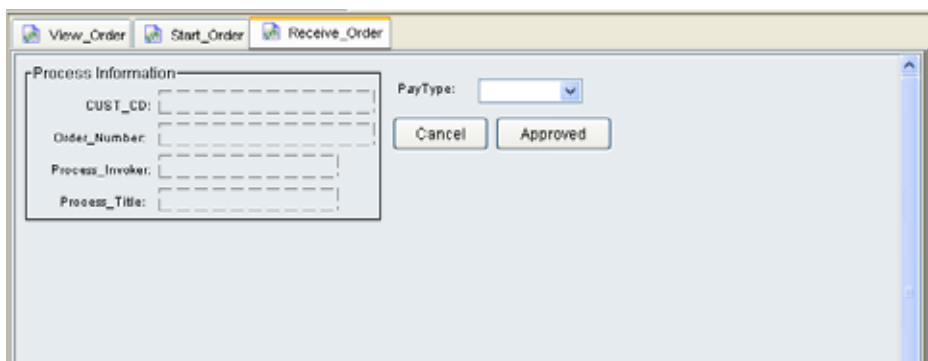
---

## サブタスク 3:

### Receive\_Order フォームのカスタマイズ

以下の手順で、Receive\_Order フォームをカスタマイズします。

1. プロジェクトタブで、Receive\_Order フォームをクリックして、開きます。
2. 前のタスクで行ったように、CUST\_CD、Order\_Number、Process\_Invoker、Process\_Title のラベルとフィールドを削除します。
3. Start\_Order フォームからこのフォームに Process Information ボックスをコピーして貼り付けします。
4. フォーム上のオブジェクトを調整して以下のような見た目になります。

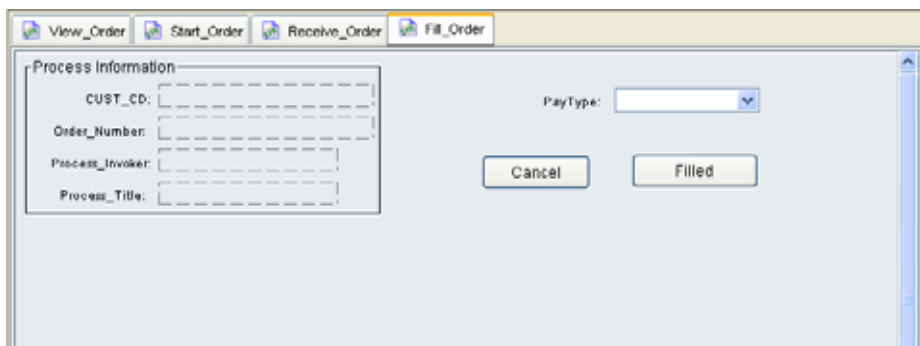


## サブタスク 4: Fill\_Order フォームのカスタマイズ

以下の手順で、Fill\_Order フォームをカスタマイズします。

1. プロジェクトタブで、Fill\_Order フォームをクリックして、開きます。
2. 前のタスクで行ったように、CUST\_CD、Order\_Number、Process\_Invoker、Process\_Title のラベルとフィールドを削除します。
3. Start\_Order フォームからこのフォームに Process Information ボックスをコピーして貼り付けします。

4. フォーム上のオブジェクトを調整して以下のような見た目になります。



5. Filled ボタンの Name プロパティに "orderFilledButton" を設定します。  
デフォルトで、このボタン名は "Button2" です。続くタスクにおいて、このボタンを参照するコードを記述するため、意味のある名前にします。
6. Filled ボタンの Title プロパティに "Filled" を設定します。

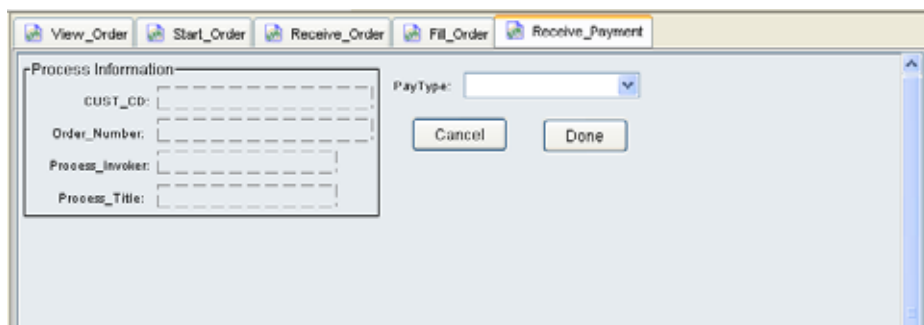
## サブタスク 5:

### Receive\_Payment フォームのカスタマイズ

以下の手順で Receive\_Order フォームのカスタマイズをします。

1. プロジェクトタブで、Receive\_Payment フォームをクリックして、開きます。
2. 前のタスクで行ったように、CUST\_CD、Order\_Number、Process\_Invoker、Process\_Title のラベルとフィールドを削除します。
3. Start\_Order フォームからこのフォームに Process Information ボックスをコピーして貼り付けします。

4. フォーム上のオブジェクトを調整して以下のような見た目になります。



The screenshot shows a software window with a title bar containing five tabs: 'View\_Order', 'Start\_Order', 'Receive\_Order', 'Fill\_Order', and 'Receive\_Payment'. The 'Receive\_Payment' tab is selected. The main area contains a 'Process Information' section with four text input fields: 'CUST\_CD:', 'Order\_Number:', 'Process\_Invoice:', and 'Process\_Title:'. To the right of these fields is a 'PayType:' dropdown menu. Below the dropdown are two buttons: 'Cancel' and 'Done'.

5. **ファイル > すべて保存** を選択します。

フォームは、より使いやすいフォーマットでプロセス情報を提供していますが、この注文プロセスを行うには外部データも必要になります。これを行うためにまずは、再利用可能なコンポーネントを作成します。

# タスク 5: ActiveWeb コンポーネントの構築

---

7

アクティビティフォームでは、既に配置したフィールドやボタンに加えて ActiveWeb コンポーネントを使用します。ActiveWeb コンポーネントは、ActiveWeb コントロールの名前でグループ化されています。このタスクでは、以下の ActiveWeb コンポーネントを作成します。

- addressView  
住所を示すフィールド群
- customerInformationView  
名前や電話番号等の顧客の詳細情報を示すフィールド群
- customerView  
顧客名と ID を提供する二つのフィールド
- orderDetailView  
特定の注文に関連するすべての項目、注文合計、出荷された注文を識別するためのトラッキング番号を一覧するグリッド
- orderView  
顧客に関連するすべての注文を一覧するグリッド
- productChooser  
顧客が注文を入力するときに選択可能なすべての製品を表示するドロップダウンリスト
- masterView  
すべてのコンポーネントをグループ化して表示するタブセット

このタスクは、以下のサブタスクからなっています。

サブタスク 1: Components フォルダの作成

サブタスク 2: addressView コンポーネントの作成

---

サブタスク 3: customerInformationView コンポーネントの作成

サブタスク 4: customerView コンポーネントの作成

サブタスク 5: productChooser コンポーネントの作成

サブタスク 6: orderDetailView コンポーネントの作成

サブタスク 7: orderView コンポーネントの作成

サブタスク 8: masterView コンポーネントの作成

この PDF ドキュメントから、このタスクで使用するコードを直接コピーすることができます。

## サブタスク 1: Components フォルダの作成

最初にコンポーネントを収める新規のフォルダをプロジェクトに作成します。

1. プロジェクトタブで、Classes フォルダを右クリックして、**新規作成 > フォルダ** 選択します。
2. 変更する名前として “ Components ” と入力し、Enter キーを押下します。

## サブタスク 2: addressView コンポーネントの作成

以下の手順で addressView コンポーネントを作成します。

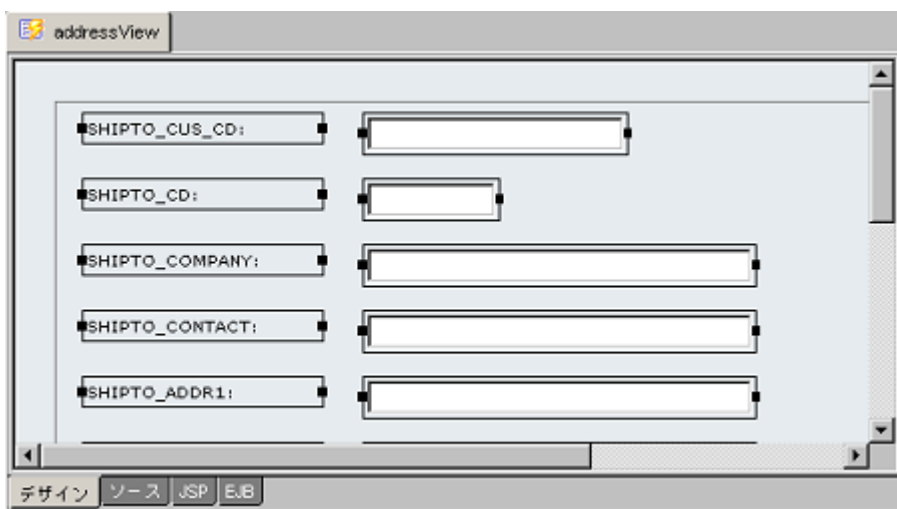
1. プロジェクトタブで、Components フォルダを右クリックして、**新規作成 > Data View** を選択します。  
データビューがデフォルト名の "Dataview1" で作成され、空のレイアウトがデザインパネル上に開きます。
2. 変更する名前として “ addressView ” と入力し、**Enter** キーを押下します。
3. デザインパネルのデータビューの境界内で、右クリックをして **データウィザード** を選択します。  
データの関連付けウィザードが表示されます。
4. データの関連付けパネルで、以下の値を選択します。

---

コネクション : OrdMgmtDB  
テーブル : ORDINV.SHIPTO  
スタイル : Single Occurrence Form  
絶対位置に配置 : (チェックする)  
Navigation Bar を使用 : (使用不可)

5. データの関連付けパネルで、**完了** ボタンをクリックします。

以下のように、データビューには SHIPTO テーブルのラベルとそのフィールドが自動で生成されます。



2つのフィールドは、アプリケーションが処理をするには必要ですがユーザーが見る必要のない情報が入っています。これらのフィールドを隠すには、フォーム上にフィールドが表示されないように Visible プロパティを設定するか、あるいはフォーカスが当たらないように Stop for Input プロパティを設定します。

6. フォームの背景をクリックして、すべての選択状態をクリアして、SHIPTO\_CUS\_CD と SHIPTO\_CD フィールドを同時に選択します。

ラベルではなく、フィールドを選択するようにしてください。

---

7. 以下のように、これらのフィールドのプロパティを設定します。

Stop For Input: (チェックを外す)

Updateable: (チェックを外す)

Background: 透過

(透過 ボタンをクリックして、OK ボタンをクリックする)

Border-Style: None

Visible: (チェックを外す)

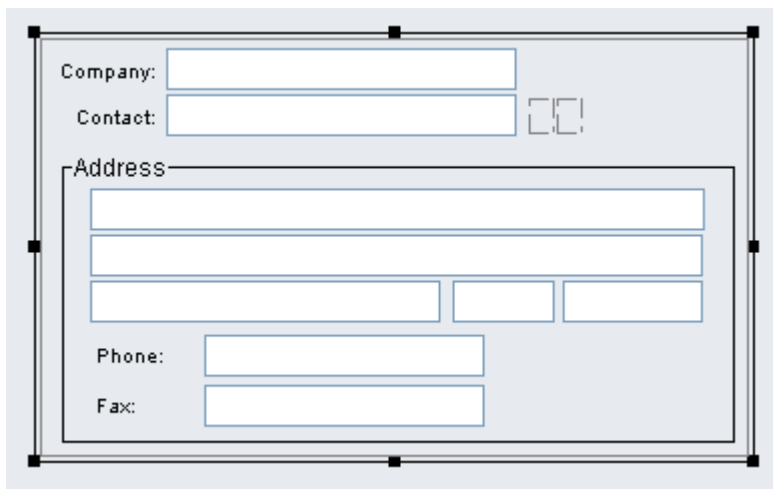
8. SHIPTO\_CUS\_CD と SHIPTO\_CD ラベルを削除します。

9. SHIPTO\_CD フィールドの Width プロパティに 15 を設定します。

10. SHIPTO\_CUS\_CD フィールドの Width プロパティに 15 を設定します。

11. 必要に応じて、スペースを調整したり、フィールドをグループ化してレイアウトを調整します。

例えば、address フィールドをボックスにグループ化をして、ラベルを削除したりできます。



The image shows a form layout with the following fields and labels:

- Company: [text input field]
- Contact: [text input field]
- Address: [grouped area containing three stacked text input fields]
- Phone: [text input field]
- Fax: [text input field]

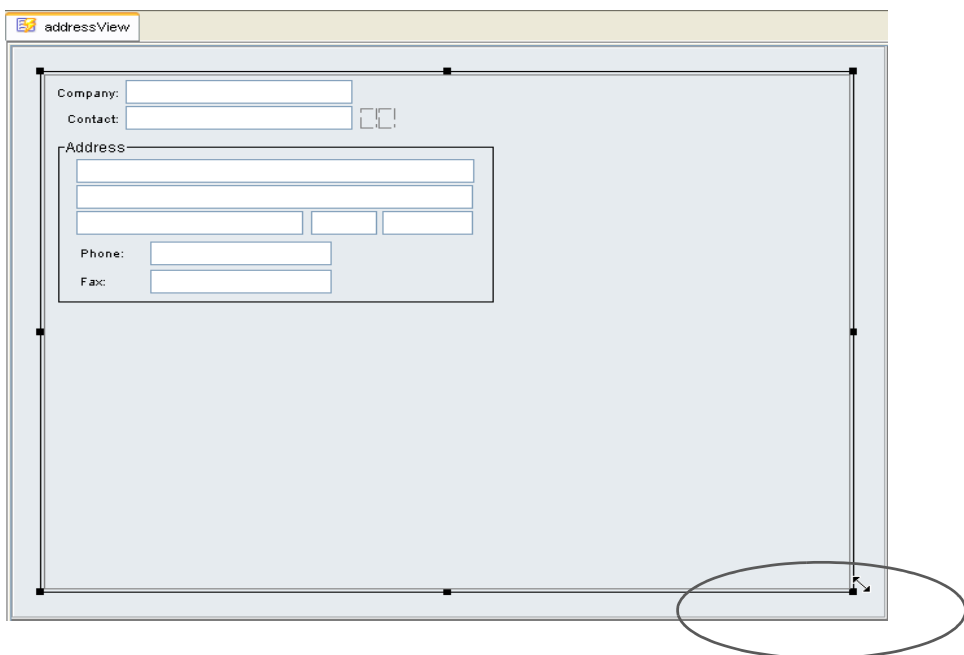
The Address field area is highlighted with a blue border, indicating it is selected or being edited.

オブジェクトをグループで選択するには、それらをマウスで範囲を指定して行うことができます。

フィールドを識別するには、そのフィールドが選択されているときにプロパティパネルの Top に表示されるその名前で行うことができます。

---

データビュー上のオブジェクトを調整して使用する面積を狭くしてから、データビュー自身を小さくサイズ変更することができます。サイズを小さくすることで、以下のタスクの作業が簡単になります。選択されたデータビューをリサイズするには、ハンドルのひとつをドラッグすることでできます。例えば、右下の角のハンドルを上へドラッグすることで、以下に見られるように垂直にも水平にもリサイズすることが可能です。



データビューの右下のハンドルを掴むためには、デザインパネルをスクロールする必要があるかもしれません。

---

**注** - ここに記述されたように、このタスク内で作成したすべてのデータビューをリサイズしてください。

---

addressView コンポーネントが完成しました。

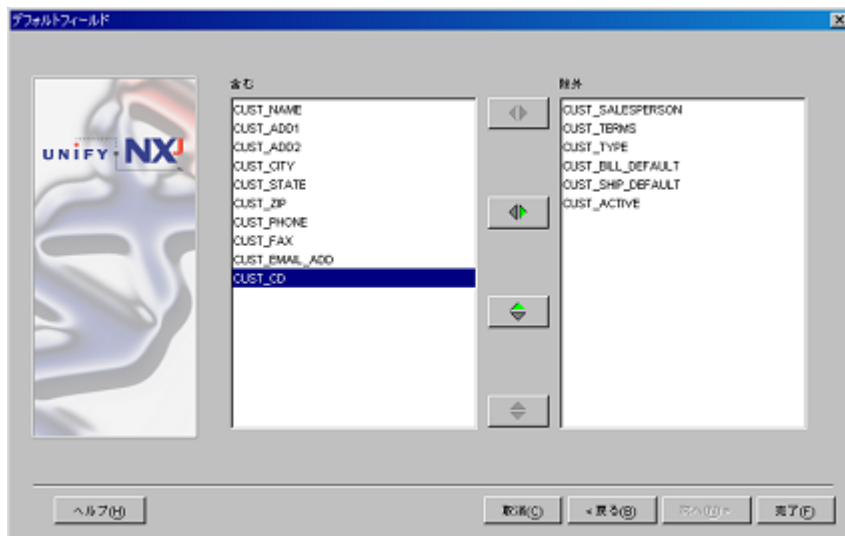
---

## サブタスク 3:

### customerInformationView コンポーネントの作成

以下の手順で customerInformationView コンポーネントを作成します。

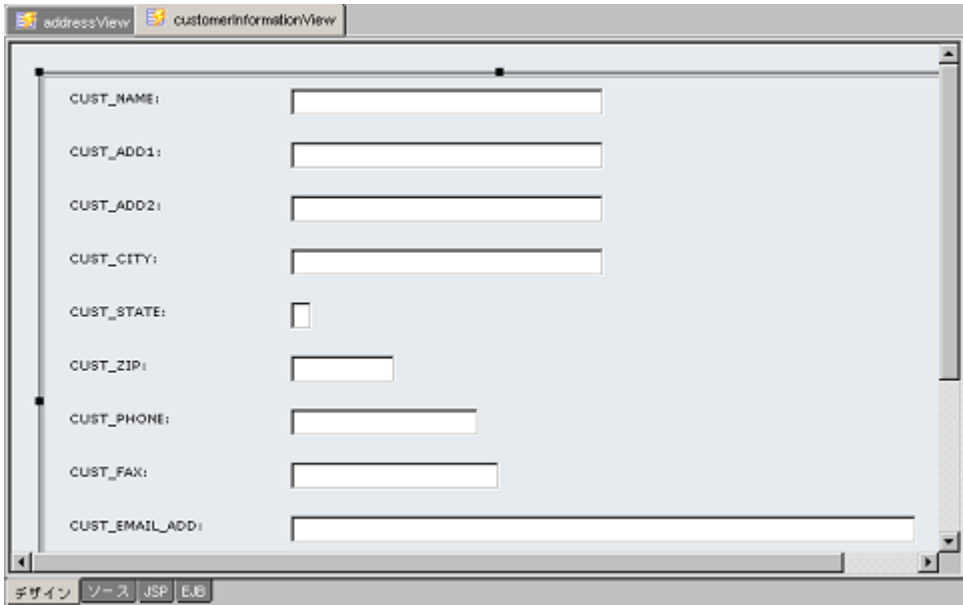
1. プロジェクトタブで Components フォルダを右クリックして、**新規作成 > DataView** を選択します。  
データビューコンテナが開かれて、デザインパネルに表示されます。
2. データビューを "customerInformationView" に名前を変更します。
3. デザインパネルのデータビューの境界内で右クリックをして、データウィザードを選択します。
4. データの関連付けパネルで、以下の値を選択します。  
コネクション : OrdMgmtDB  
テーブル : ORDINV.CUSTOMER  
スタイル : Single Occurrence Form  
絶対位置に配置 : (チェックする)  
Navigation Bar を使用 : (使用不可)
5. **次へ** ボタンをクリックします。
6. デフォルトフィールドパネルで、次に示されるようにフィールドを除外します。



フィールドを除外の列に移すには、それを選択して、右向き矢印をクリックします。

7. **完了** ボタンをクリックします。

データビューには、指定したフィールドとそのラベルが自動的に生成されます。



8. CUST\_CD フィールドを選択して、以下のプロパティを設定します。

Stop for Input: (チェックを外す)

Updateable: (チェックを外す)

Background: 透過

(透過 ボタンをクリックして、OK ボタンをクリックする)

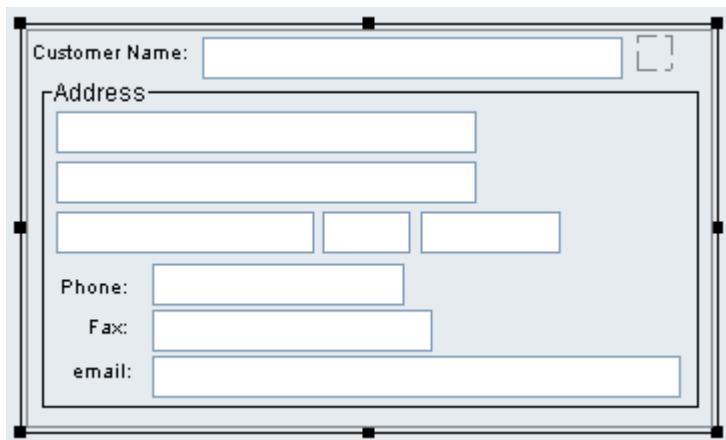
Border-Style: None

Visible: (チェックを外す)

Width: 15

9. CUST\_CD のラベルを削除します。

これらのラベルとフィールドのレイアウトを変更して、希望する Look and Feel にすることができます。以下のように調整します。



The image shows a screenshot of a form layout for customer information. The form is contained within a rectangular frame with a double border. At the top, there is a label "Customer Name:" followed by a text input field and a small square icon to its right. Below this is a section labeled "Address" which contains three stacked text input fields. Underneath the address fields are three smaller text input fields arranged horizontally. At the bottom of the form, there are four labels with corresponding text input fields: "Phone:", "Fax:", "email:", and another unlabeled field.

10. 背景をクリックしてデータビューを選択します。

プロパティパネルのトップのドロップダウンリストを見ることでデータビューが正しく選択されているかどうか確認できます。"<component>"が表示されます。

11. Auto Find チェックボックスをチェックします。

Database	
Auto Find	<input checked="" type="checkbox"/>
Clear After Add	<input type="checkbox"/>

Auto Find プロパティは、データビューが実行時に指定されたマスタ / 詳細関係、あるいは Query パラメータに基づいて検索を行うことを指定します。このプロパティが設定されていない場合、ユーザがコマンドに割り当てられたボタンをクリックするように、FIND コマンドがトリガされたときに検索を行います。

customerInformationView コンポーネントは完成です。

---

## サブタスク 4:

### customerView コンポーネントの作成

以下の手順で customerView コンポーネントを作成します。

1. この前のタスクで行ったのと同様に、components フォルダに新しく "customerView" という名前のデータビューを作成します。
2. デザインパネルでデータビューの境界内で右クリックをして、データウィザードを選択します。
3. データの関連付けパネルで以下の値を選択します。

コネクション : OrdMgmtDB  
テーブル : ORDINV.LOGIN  
スタイル : Single Occurrence Form  
絶対位置に配置 : (チェックを外す)  
Navigation Bar を使用 : (使用不可)

4. 完了 ボタンをクリックします。

絶対位置に配置のオプションはチェックされていないので、ラベルとフィールドは HTML テーブルオブジェクトの中に配置されます。

必要に応じて、テーブル、関連する列をリサイズできます。テーブルを選択して、その Cell Padding プロパティに 2 を設定することをお勧めします。これにより、実行時のレイアウトの外観が良くなります。

5. データビューを選択して、その Auto Find プロパティのチェックボックスをチェックします。

customerView コンポーネントは完成です。

## サブタスク 5:

### productChooser コンポーネントの作成

以下の手順で、productChooser コンポーネントを作成します。

1. プロジェクトタブで、components フォルダを右クリックして、**新規作成 > Dropdown list** を選択します。
2. Dropdown1 エントリの名前を "productChooser" に変更します。

- productChooser コンポーネントの Query プロパティに以下の値を設定して、OK ボタンをクリックします。



- productChooser コンポーネントに以下のプロパティを設定します。

Width: 350

Auto Accept: (チェックする)

Immediate: (チェックする)

Auto Accept と Immediate プロパティは、ユーザがリストから製品を選択するとすぐに、価格が自動的に更新されてレコードが追加されることを示しています。

これで、productChooser コンポーネントは完成です。

## サブタスク 6:

### orderDetailView コンポーネントの作成

以下の手順で orderDetailView コンポーネントを作成します。

- 前のタスクで行ったと同様に、components フォルダ内に "orderDetailView" という名前の新規のデータビューを作成します。
- デザインパネルでデータビューの境界内で右クリックをしてデータウィザードを選択します。

データの関連付けウィザードで、グリッドにフィールドを作成します。ここまでの2つのデータビューのように、一件のレコードではなく、複数のレコードを表示する場合にグリッドは使われます。

- データの関連付けパネルで以下の値を選択します。

コネクション : OrdMgmtDB  
テーブル : ORDINV.ORDER\_DETAIL  
スタイル : Multi Occurrence Form with Grid  
絶対位置に配置 : ( 利用不可 )  
Navigation Bar を使用 : ( チェックを外す )

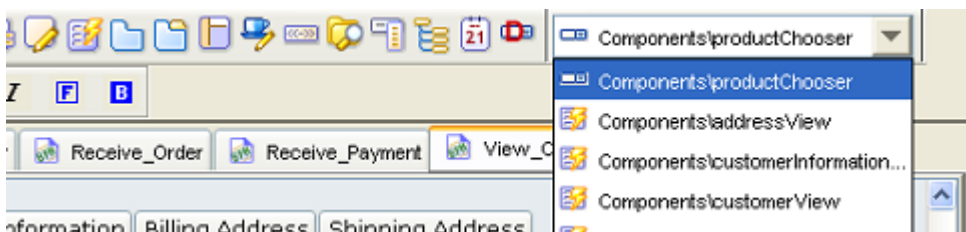
- 次へ ボタンをクリックします。
- ORDER\_NUMBER フィールドを除外します。
- 完了 ボタンをクリックします。

データビューには、設定通りにフィールドがグリッドで生成されます。グリッド内の PRODUCT\_NUMBER フィールドは、注文された製品を示します。このフィールド内容を置き換えるには、productChooser コンポーネントを使用します。レイアウト上にコンポーネントを挿入するには、ボックスのようなオブジェクトを挿入したのとは異なる方法で行ないます。

- グリッド内の PRODUCT\_NUMBER 列を削除します。

グリッド内の列を削除するには、グリッドの 1 行目にある該当のフィールドをクリックして選択してから、**Delete** キーを押下します。

- グリッドが選択されている状態で、下図のようにコンポーネントドロップダウンリストから productChooser コンポーネントを選択します。



- グリッド内でクリックをしてコンポーネントを配置します。

productChooser コンポーネント ( 技術的にはそのインスタンス ) がグリッドに追加されます。カーソルの正確な位置によって、新しい列が PRODUCT\_PRICE 列の右か左に挿入されます。

- グリッド内の productChooser コンポーネントが選択された状態で、以下のプロパティを設定します。

Target Field: PRODUCT\_NUMBER

---

コンポーネントは、再利用でき、毎回異なるターゲット列に接続するよう設計されているので、このプロパティは、コンポーネントがフォームに配置された後に設定する必要があります。

11. グリッドのヘッダテキストを、"Column1" から "Description" に、"PRODUCT\_PRICE" から "Price" に変更します。

ヘッダテキストを変更するには、そこをダブルクリックして選択して、上書きします。

12. 必要に応じて、2つの列をグリッド内で順序を入れ替えれば、Description が Price の前にきます。

グリッド内で列を移動するには、ヘッダの列をクリックして、他の列の上にドラッグします。

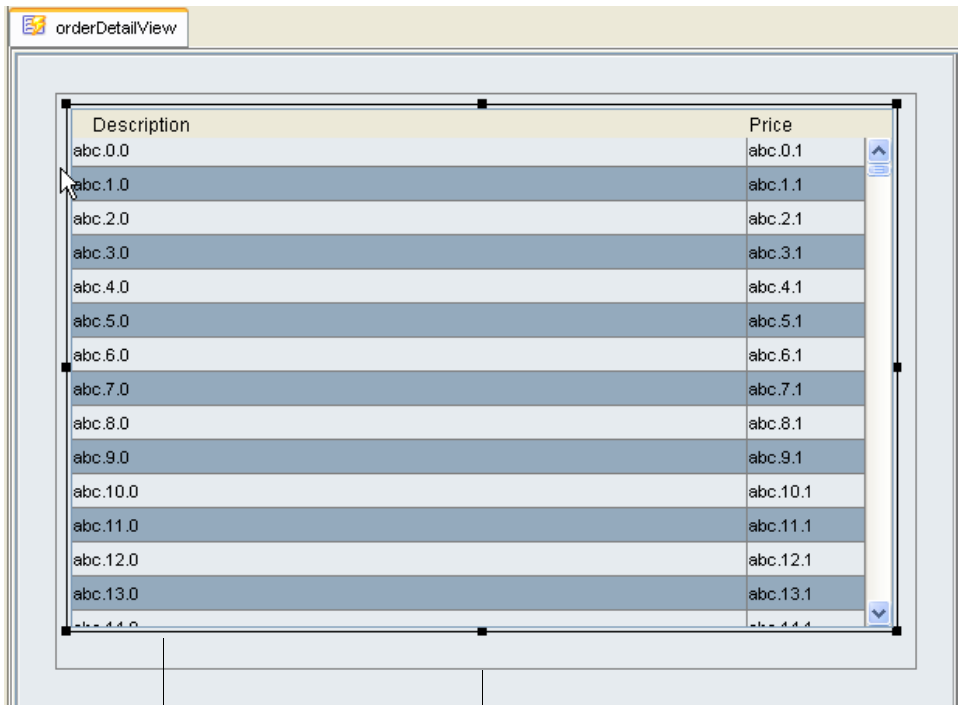
13. データビューを選択して、Clear After Add プロパティをチェックします。

データビューを選択するには、グリッドの外側のデータビューの背景をクリックします。

Clear After Add プロパティは、新規のレコードが追加される準備にデータビューを初期化することを示すプロパティです。

14. グリッドを小さくリサイズして、この後のタスクの作業を簡単にします。

6 行ほど表示されるように、垂直方向にリサイズします。データビューをリサイズしたのと同様に、下部のハンドルを上へドラッグすればグリッドもリサイズされます。データビューではなくグリッドを選択するようにしてください。



グリッドの境界  
(選択)

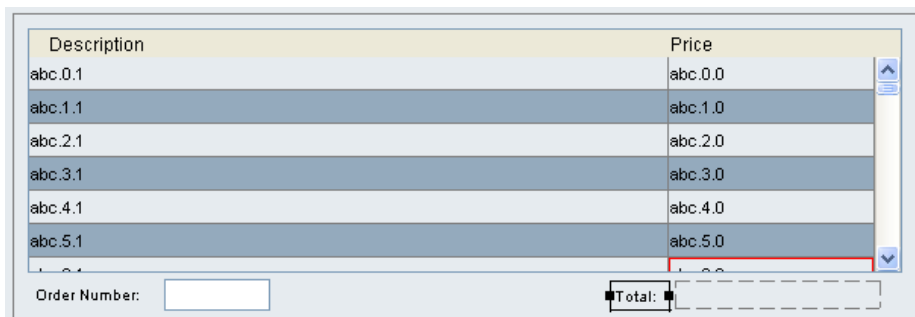
データビューの境界  
(非選択)

セルの1つを右クリックして、繰り返し領域の選択を選択することでも、グリッドを選択することができます。

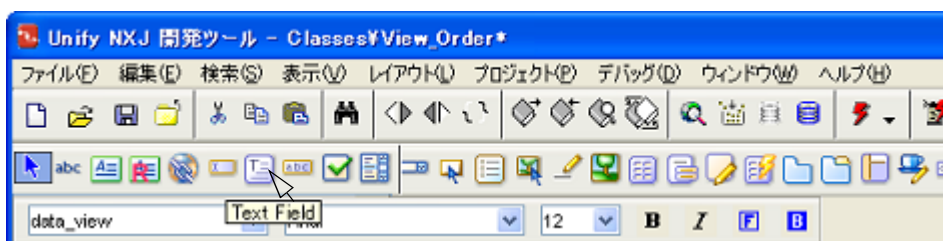
15. Price 列の Updatable プロパティのチェックを外します。

Updatable プロパティは、プロパティの Interactive グループにあります。

16. 下図のように、グリッドの下に、2つのラベル、1つのフィールド、1つのダイナミックテキストを配置します。



データビューレイアウトに、これらの新規のオブジェクトを配置するには、Unify NXJ 開発ツールのツールバーから適切なボタンをクリックしてから、レイアウト上で該当するオブジェクトを配置する場所でクリックします。ツールバーボタンがどのオブジェクトを表しているかを知るには、そのボタンの上にカーソルを近づけるとツールチップが表示されます。



オブジェクトにはデフォルトの名前が割り当てられていますが、Name プロパティを設定することで変更することができます。サンプルアプリケーションでは、指示がない限りはデフォルトの名前を変更しないでください。

17. 新規オブジェクトのプロパティを以下の通りに変更します。

新規フィールド：

Target Field: ORDER\_NUMBER

新規のダイナミックテキスト：

Expression: =orderTotal

ダイナミック式とは、静的なテキストではなく変数を基にフィールドの値を決めるものです。変数は、フォームのソースコードで設定されています。

- 
18. ソースタブをクリックして、スクリプトのトップにある以下の構文を探します。

```
DATA VIEW orderDetailView
{
```

19. 左括弧のすぐ後に、以下の宣言文を入れます。

```
NullableAmountVariable orderTotal = 0.0;
```

20. この宣言文のすぐ下に、以下のイベントセクションを入れます。

```
BEFORE FIND
{
orderTotal = 0.0;
}

ON FIND
{
orderTotal = orderTotal + PRODUCT_PRICE;
}
```

BEFORE FIND イベントセクションには、FIND 操作を行う前に実行するロジックを指定します。ON FIND イベントセクションは、検索結果の各レコードごとに実行されます。この場合、orderTotal 変数は、最初に初期化されて、PRODUCT\_PRICE の値を集計するために使用されます。

21. 下方向にスクロールして以下の構文を探します。

```
FIELD PRODUCT_NUMBER extends Components.productChooser
{
```

22. 左括弧のすぐ後に、以下のイベントセクションを入れます。

```
ON DATA ACCEPT
{
EXEC SQL USING CONNECTION OrdMgmtDB
    SELECT PRODUCT_LIST FROM ORDINV.PRODUCT
    WHERE PRODUCT_ID = :PRODUCT_NUMBER
    INTO PRODUCT_PRICE;
orderTotal = orderTotal + PRODUCT_PRICE;
session.queueCommand("ADD_UPDATE");
}
```

---

ON DATA ACCEPT セクションは、ユーザによってフィールドの値が変更されたタイミングで自動的に実行されます。検証や、ビジネスルールの起動に使用されます。

これは、ActiveWeb 技術の重要な事例です。

23. プロジェクト > 変更分のみ Make を選択します。

24. ファイル > すべて保存 を選択します。

## サブタスク 7: orderView コンポーネントの作成

1. 前のタスクで行ったと同様に、components フォルダ内に orderView という名前の新規のデータビューを作成します。
2. デザインパネルで、データビューの境界内で右クリックをしてデータウィザードを選択します。
3. データの関連付けパネルで以下の値を選択します。

コネクション：OrdMgmtDB

テーブル：ORDINV.ORDERS

スタイル：Multi Occurrence Form with Grid

絶対位置に配置：（利用不可）

Navigation Bar を使用：（チェックを外す）

4. 次へ ボタンをクリックします。

5. 以下のフィールドを除外します。

CUSTOMER\_ID

TRACKING\_NUMBER

SHIPPING\_COST

SHIP\_DATE

6. 完了 ボタンをクリックします。

指定したフィールドのグリッドがデータビュー内に生成されます。グリッドをリサイズして、7 行程度が表示されるようにします。

7. グリッドの右に、4 行 2 列のテーブルを配置します。テキストフィールドと同様に、テーブルを配置するには、ツールバーで選択してから、レイアウト上でクリックします。

レイアウトにテーブルを配置したときに、列数と行数を指定するプロンプトが表示されますので、4 行 2 列と入力します。



Table  
ボタン

---

最初の列にラベル、2列目にフィールドを追加します。Table グループの cell padding に 2 を設定して、テーブルを小さくします。

Customer ID	<input type="text"/>
Shipping Cost	<input type="text"/>
Tracking Number	<input type="text"/>
Shipping Date	<input type="text"/>

フィールドは、次のデータベース列に紐付けられなければなりません。(それぞれの Target field プロパティに設定)

CUSTOMER\_ID  
SHIPPING\_COST  
TRACKING\_NUMBER  
SHIP\_DATE

- Shipping Cost フィールドの Data Type プロパティに、Amount を設定します。

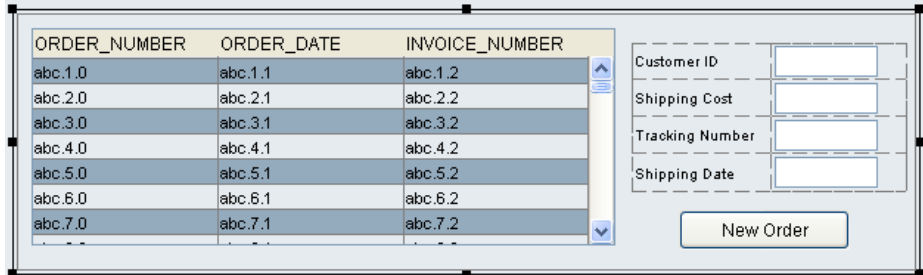
デフォルトで、データタイプは、ターゲット列のデータタイプに準拠して Float が設定されます。このフィールドは、金額を扱うので、Amount データタイプが適切です。

注：フィールドに、プロパティの General グループにある固有の Data Type プロパティを必ず変更してください。Db Type プロパティは、Data Binding グループにあります。

- データビュー内の shipping date の下に新規のボタンを追加して、以下のようプロパティを設定します。

Command: CLEAR TO ADD  
Command Context: parent  
Title: New Order

余分なスペースをなくすために、必要に応じてデータビューをリサイズすると、下図のようなレイアウトになります。



10. データビューの背景をクリックして選択します。
11. ソースタブをクリックして、スクリプトのトップに以下の構文を探します。

```
DATA VIEW orderView
{
```

データビューは、各注文に割り当てられるユニークな注文番号を必要とし、番号はデータベーステーブルで管理されます。NXJプログラミング言語で書かれた簡単な手法で注文番号は管理されます。

12. 左括弧の後の行にカーソルを合わせて、以下のメソッド定義を追加します。

```
public int getNextNum(String colName)
throws Exception
{
    int tmpNextValue = 0;
    EXEC SQL USING CONNECTION OrdMgmtDB
    SELECT NEXTNO_VALUE FROM ORDINV.NEXTNO
    WHERE NEXTNO_FIELD = :colName
    INTO tmpNextValue;

    EXEC SQL USING CONNECTION OrdMgmtDB
    UPDATE ORDINV.NEXTNO SET NEXTNO_VALUE = :tmpNext-
tValue+1
    WHERE NEXTNO_FIELD = :colName;

    return tmpNextValue;
}
```

---

このように、標準的なメソッド定義に NXJ の埋込み型 SQL が利用できることで、開発者は、Java の構文ではなく標準的な SQL 構文を使用して書くことができます。埋込み型 SQL は、標準の Java 構文に変換されて、コンパイル時に同じトランザクション、データベース接続にまとめられます。アプリケーションがコンパイルされた後、この EJB のソースは EJB タブで見ることができます。

このフォームを使って実行される主な操作の一つは、新規注文の登録です。New Order ボタンが押下されたときには、ボタンに関連付けられたコマンドが実行されます。この場合のコマンドは、NXJ が事前に提供している CLEAR TO ADD コマンドです。このコマンドは、INSERT 操作のためにターゲットテーブルを準備します。INSERT は、ユーザが product リストから product を選択したときに実行されます。

データベースのテーブルに新規注文が登録される前に、いくつかの初期化作業が必要です。この初期化に最も適しているのは、名付けられたイベント（この場合、Clear To Add コマンド）が実行される時だけに実行されるセクションである ON CLEAR TO ADD イベントセクションです。

13. getNextNum() メソッドの右括弧のすぐ下に、以下のロジックを入れます。

```
ON CLEAR TO ADD
{
    ORDER_NUMBER = getNextNum("ORD_NO");
    ORDER_DATE = session.currentDate;
    INVOICE_NUMBER = getNextNum("INV_NO");
    TRACKING_NUMBER = 0;
    session.queueCommand("ADD_UPDATE");
}
```

14. プロジェクト > 変更分のみ Make を選択します。

15. ファイル > すべて保存 を選択します。

## サブタスク 8:masterView コンポーネントの作成

以下の手順で、masterView コンポーネントを作成します。

1. 前のタスクで行ったのと同様に、components フォルダに "masterView" という名前の新規データビューを作成します。

2. データビューに大きな **Tab Set** を追加します。

タブセットはコンテナオブジェクトです。データビューにタブセットを配置するときに、ドラッグでサイズを設定できます。また、配置後にリサイズすることもできます。タブセットは、タブページだけしか持つことができませんが、タブページにはどんなオブジェクトも含むことができます。



Tab Set

3. 最初に、**Tab** ボタンをクリックして、タブセットコンテナ内でクリックする方法で、タブセットの中に4つのタブを追加します。



Tab ボタン

4. 4つのタブのそれぞれの Title プロパティの設定を以下の順で行います。

Orders  
Customer Information  
Billing Address  
Shipping Address

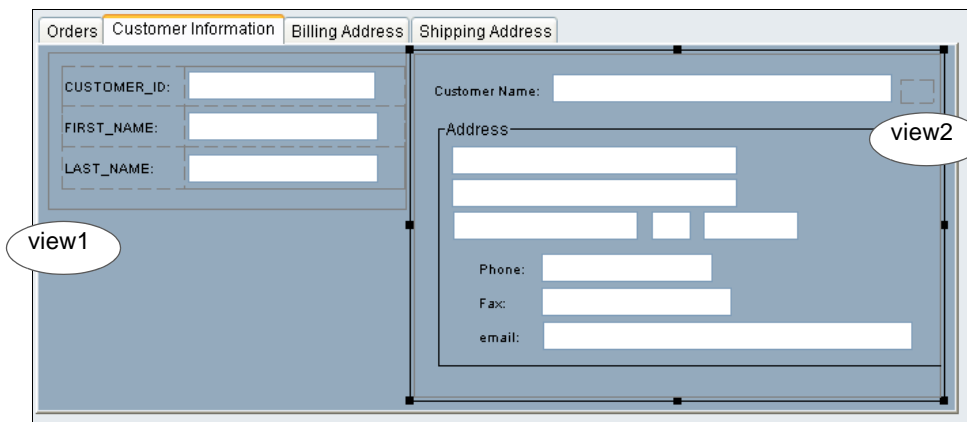
Title プロパティは、Tab グループにあります。

5. "Customer Information" タブを選択して、以下の順でコンポーネントを追加します。

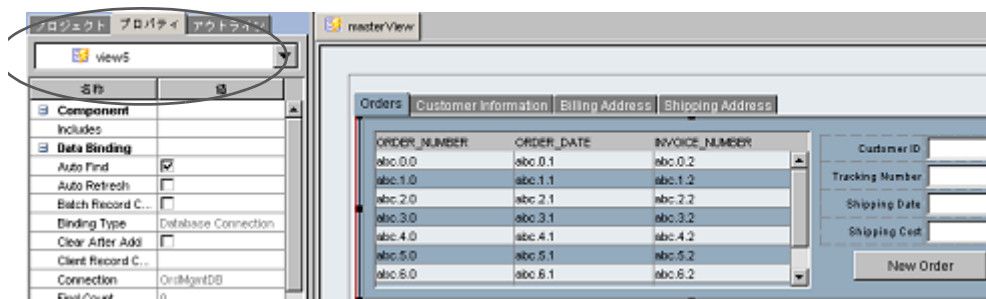
customerView  
customerInformationView

productChooser コンポーネントを前のタスクで追加したのと同じ方法でドロップダウンリストからコンポーネントを挿入します。

完成したタブセットは、以下のようになります。



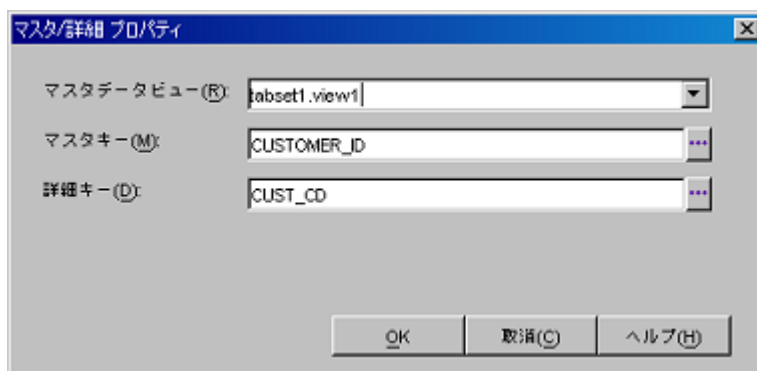
コンポーネントインスタンスに割り当てられるデフォルト名は、配置された順序に依存するので、コンポーネントを配置する順序は大変重要です。この後、これらの名前でもコンポーネントインスタンスを参照するコードを使用します。コンポーネントに割り当てられた名前は、そのコンポーネントを選択すると、名前がプロパティタブに下図のように表示されます。



新規に配置されたコンポーネントは、マスタ / 詳細関係を持ちます。マスタ / 詳細関係は、自動的に2つのデータ構造を結合する親 / 子供の関係を定義します。親レコードが変更されると、子供レコードは新しい親を反映するために自動的に更新されます。ページ全体のリフレッシュをすることなく動的にこの表示変更ができるのも、ActiveWeb 技術の特徴です。フォーム全体ではなく、データのみがリフレッシュされます。

6. "customerInformationView" コンポーネントを選択して、Master Keys プロパティの値の列をクリックしてマスタ / 詳細プロパティダイアログを表示します。

下図のように、マスタ / 詳細プロパティを設定します。



---

値の設定は、選択ボタンを使っても直接値を入力してもどちらでも構いません。**OK** ボタンをクリックし、変更を保存して、ダイアログを終了します。

---

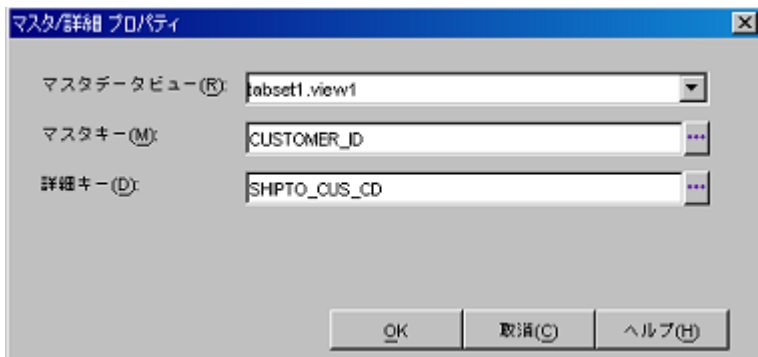
注 - このダイアログでは、Master Keys, Detail Keys, Master Data View の3つのプロパティを設定します。これらのプロパティのどの値の列をクリックしても、ダイアログを開くことができます。

---

7. "Billing Address" タブで、addressView コンポーネントを追加します。

The image shows a software dialog box with four tabs: "Orders", "Customer Information", "Billing Address", and "Shipping Address". The "Billing Address" tab is selected and highlighted. The dialog contains several input fields: "Company:", "Contact:", "Address" (with three stacked lines), "Phone:", and "Fax:". A "view3" button is circled in red.

- コンポーネントを選択して、以下のようにマスタ / 詳細プロパティを設定します。

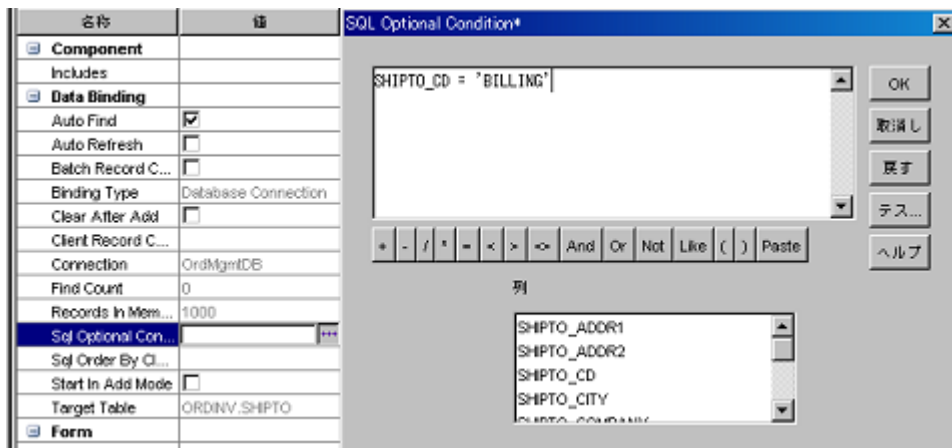


上で行ったように、Master Keys プロパティの値の列をクリックして、マスタ / 詳細プロパティダイアログを開きます。図のように値を指定して、OK ボタンをクリックしてダイアログを終了します。

- コンポーネントが選択されたままの状態では、「Sql Optional Condition」プロパティを設定して請求住所を検索します。

このプロパティを設定するには、値の列をクリックして、次に**選択** ボタンをクリックして表示したダイアログに以下のクエリを入力します。

SHIPTO\_CD = 'BILLING'



OK ボタンをクリックし、変更を保存して、ダイアログを終了します。

SqlOptionalCondition プロパティは、デフォルトの検索のパラメータになります。この例では、同じデータビューが請求先と出荷先の住所に使用されています。動作を識別するために、ここでは請求先住所にこのプロパティを使います。後程、データビューを再利用しますが、そのときは出荷先住所のためにこのプロパティを使用します。これは、再利用をどのように利用するか  
の事例です。

10. “Shipping Address” タブを選択して、addressView コンポーネントを追加します。
11. “addressView” コンポーネントが選択された状態のままで、下図のようにマスタ / 詳細プロパティを設定します。

マスタ/詳細 プロパティ

マスタデータビュー(R): tabset1.view1

マスタキー(M): CUSTOMER\_ID

詳細キー(D): SHIPTO\_CUS\_CD

OK 取消(C) ヘルプ(H)

12. コンポーネントが選択された状態のままで、“Sql Optional Condition” プロパティに SHIPTO\_CD = 'SHIPPING' を設定します。
13. "Orders" タブを選択して、orderView コンポーネントを追加します。

Orders Customer Information Billing Address Shipping Address

ORDER_NUMBER	ORDER_DATE	INVOICE_NUMBER
abc.0.0	abc.0.1	abc.0.2
abc.1.0	abc.1.1	abc.1.2
view5	abc.2.1	abc.2.2
	abc.3.1	abc.3.2
abc.4.0	abc.4.1	abc.4.2
abc.5.0	abc.5.1	abc.5.2
abc.6.0	abc.6.1	abc.6.2

Customer ID

Tracking Number

Shipping Date

Shipping Cost

New Order

14. “orderView” コンポーネントが選択された状態のままで、下図のようにマスタ / 詳細プロパティを設定します。

マスタ/詳細 プロパティ

マスターテーブル (R): tabset1.view1

マスターキー (M): CUSTOMER\_ID

詳細キー (D): CUSTOMER\_ID

OK 取消 (C) ヘルプ (H)

15. 同じ "Order" タブで、以下のように orderDetailView コンポーネントを追加します。

Orders Customer Information Billing Address Shipping Address

ORDER_NUMBER	ORDER_DATE	INVOICE_NUMBER
abc.0.0	abc.0.1	abc.0.2
abc.1.0	abc.1.1	abc.1.2
abc.2.1	abc.2.1	abc.2.2
abc.3.1	abc.3.1	abc.3.2
abc.4.0	abc.4.1	abc.4.2
abc.5.0	abc.5.1	abc.5.2
abc.6.0	abc.6.1	abc.6.2

view5

Customer ID

Tracking Number

Shipping Date

Shipping Cost

New Order

Description	Price
abc.0.0	abc.0.1
abc.1.0	abc.1.1
abc.2.1	abc.2.1
abc.3.1	abc.3.1
abc.4.1	abc.4.1
abc.5.1	abc.5.1
abc.6.1	abc.6.1

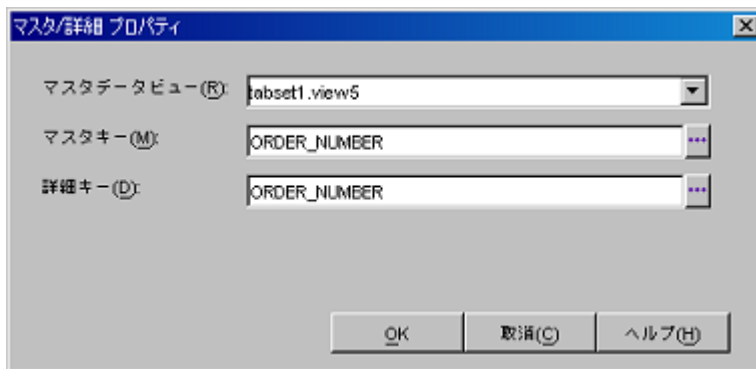
view6

Order Number

Total:

このタブにコンポーネントを追加したら、タブをリサイズしてコンポーネントすべてが表示されるようにします。

16. コンポーネントが選択された状態のままで、下図のようにマスタ / 詳細プロパティを設定します。



新しい注文が登録される時、このデータビューが追加モードに変わるようにソースコードを追加する必要があります。

17. orderView コンポーネント ( view5 ) を選択します。
18. ソースタブをクリックして、"DATA VIEW view5 extends Components.orderView" のコードを探します。その左括弧のすぐ下に次のイベントセクションを追加します。

```
AFTER ADD
{
    session.queueCommand("CLEAR_TO_ADD", "view6");
}
```

このイベントセクションは、ADD 操作が完了した後に実行されます。この例では、CLEAR TO ADD コマンドの orderDetail ビューに対する実行が処理待ちキューに入れられました。キューコマンドは、開発者がプログラムでコマンドの実行を管理するためのものです。キューコマンドは、入れられた順番で処理がされます。コマンドが完了すると、アプリケーションはエンドユーザーにフォーカスを戻します。

19. プロジェクト > 変更分のみ Make を選択します。

これで、アクティビティフォームを組み立てる準備が整いました。

# タスク 6: ActiveWeb コンポーネントをアクティ ビティフォームに配置

8

各アクティビティフォームは、masterView コンポーネントを使用します。

コンポーネントを使用すると NXJ アプリケーションのメンテナンスの生産性が飛躍的に向上します。一旦、フォームにコンポーネントのインスタンスが配置されると、それは、その親クラスの定義と関連付けられます。定義を変更する場合、その変更は自動的に各インスタンスに適用されます。

このタスクは、次のサブタスクからなります。

- サブタスク 1: Start\_Order フォームの完成
- サブタスク 2: View\_Order フォームの完成
- サブタスク 3: Receive\_Order フォームの完成
- サブタスク 4: Fill\_Order フォームの完成
- サブタスク 5: Receive\_Payment フォームの完成

この PDF ファイルからこのタスクで使用するコードを直接コピーできます。

## サブタスク 1: Start\_Order フォームの完成

以下の手順で、Start\_Order フォームにコンポーネントを追加します。

1. Unify NXJ 開発ツールのプロジェクトタブで、Start\_Order フォームをクリックして、開きます。
2. ツールバーのコンポーネントのドロップダウンリストから、masterView を選択します。

3. デザインパネルで、下図のように既存のフィールドの下に masterView コンポーネントを配置します。

The screenshot shows a software application window titled "Start\_Order". It features a "Finish Order" section with "Order Complete" and "Cancel" buttons, a "Process Information" section with input fields for "CUST\_CD", "Order\_Number", "Process\_Invoker", and "Process\_Title", and a "PayType" dropdown menu. Below these is the "Orders" section, which includes tabs for "Customer Information", "Billing Address", and "Shipping Address". The "Customer Information" tab is active, displaying a table with columns "ORDER\_NUMBER", "ORDER\_DATE", and "INVOICE\_NUMBER". To the right of this table are input fields for "Customer ID", "Tracking Number", "Shipping Date", and "Shipping Cost", along with a "New Order" button. At the bottom, there is another table with columns "Description" and "Price", an "Order Number" input field, and a "Total" field.

このフォームには、プロセスオペランドを更新してビジネスプロセスを起動する新規のコマンドが必要です。

4. ソースタブをクリックします。
5. スクリプトの最初の左括弧のすぐ下に以下のコマンド定義を入れます。

```
COMMAND startProcess
{
    box1.Order_Number =
view1.tabset1.view5.ORDER_NUMBER;
    box1.Process_Title = "Order # " +
```

---

```
box1.Order_Number;
    box1.CUST_CD = view1.tabset1.view1.CUSTOMER_ID;
    box1.Process_Invoker =
view1.tabset1.view1.FIRST_NAME;
    session.queueCommand("Start_Process");
}
```

このコマンドでは、プロセス変数（オペランド）の初期化をします。一旦、初期化されると、プロセスをスタートするためのコマンドがキューに入ります。

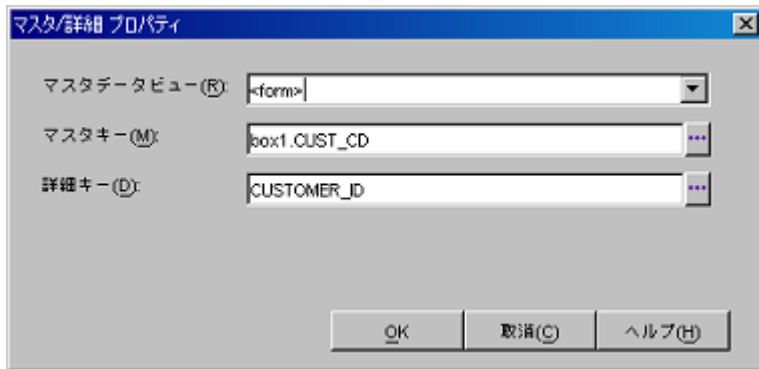
6. デザインパネルで、フォームの上部にある Order Complete ボタンを選択します。
7. その Command プロパティに "startProcess" を設定します。

デフォルトで、このボタンの Command プロパティには、"Start Process" システムコマンドが設定されていますが、このアプリケーションで使用するコマンドではありません。

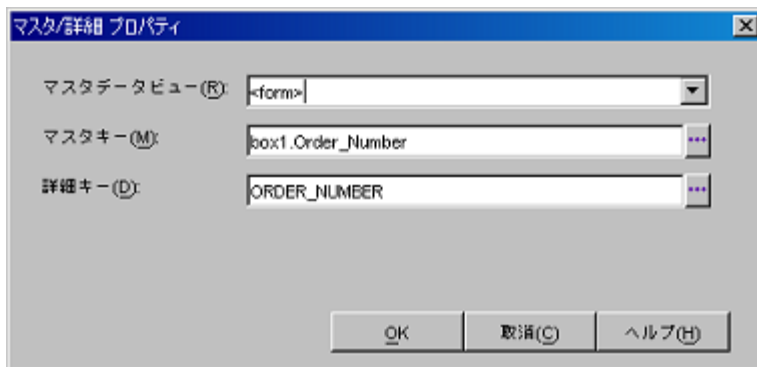
## サブタスク 2: View\_Order フォームの完成

以下の手順で View\_Order フォームにコンポーネントを追加します。

1. プロジェクトタブで、View\_Order フォームをクリックして開きます。
2. 前のタスクと同じ手順で、フォームに masterView コンポーネントを配置します。
3. "Customer Information" タブをクリックして、左の方にある view1 という名前のデータビュー (customerView) を選択します。
4. 下図のように、このデータビューのマスタ / 詳細 プロパティを設定します。



- Orders タブをクリックして、上部にある view5(orderView) を選択します。
- 下図のようにマスタ / 詳細プロパティを設定します。



最新の注文だけが、データビューの生成に使われます。

- デザインパネルで、**Orders タブ**をクリックして、New Order ボタンを選択します。
- "Visible" プロパティのチェックを外します。
- プロジェクト > **変更分のみ Make** を選択します。
- ファイル > **すべて保存** を選択します。

---

## サブタスク 3: Receive\_Order フォームの完成

サブタスク 2 と同じ内容を Receive\_Order フォームに対して繰り返します。

## サブタスク 4: Fill\_Order フォームの完成

サブタスク 2 と同じ内容を Fill\_Order フォームに対して繰り返します。

## サブタスク 5: Receive\_Payment フォームの完成

サブタスク 2 と同じ内容を Receive\_Payment フォームに対して繰り返します。

新規プロセスを起動する準備が整いましたので、すべてのステップを始めて新しい機能を確認してください。

# タスク7: アプリケーション の配備と実行

9

アクティビティフォームが完成したので、実行環境でアプリケーションを見てみます。アプリケーションを一通り行うには、タスク3で行ったのと同様にユーザーコミュニティの様々なロールでログインをする必要があります。

---

**注** – このタスクを実行するために、ordmgmt プロパティプロジェクトを使用している場合、最初にタスク3のサブタスク2のステップ7に説明されているように、OrdMgmt プロセスをアップロードする必要があります。

---

このタスクは以下のサブタスクからなります。

サブタスク 1: アプリケーションの配備

サブタスク 2: OrdMgmt プロセスを起動して Star\_Order フォームから  
注文登録

サブタスク 3: Receive\_Order フォームを使用している注文を承認

サブタスク 4: Fill\_Order フォームを使用して注文の出荷状況を示す

サブタスク 5: Receive\_Payment フォームを使用して注文の代金回収状況を示す

## サブタスク 1: アプリケーションの配備

以下の手順でアプリケーションを配備し、実行します。

1. Unify NXJ 開発ツールのメニューから、**プロジェクト > すべて Make** を選択します。

数秒後に、メッセージパネルのコンパイルタブに "Make 終了" のメッセージが表示されます。

2. Unify NXJ 開発ツールのメニューから、**プロジェクト > 配備** を選択します。

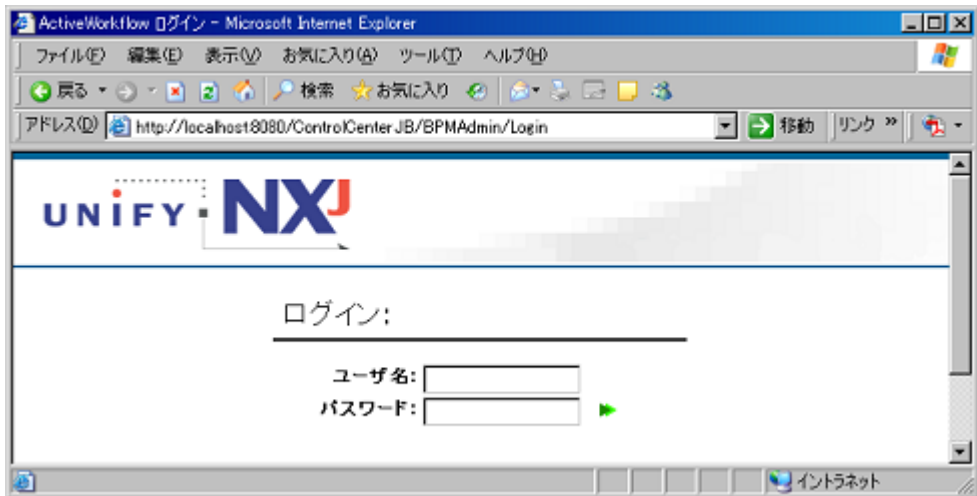
数秒後に、アプリケーションが構築されて、アプリケーションサーバに配備されます。

データソースの作成ダイアログが表示されたら、はい ボタンをクリックします。

3. Web ブラウザから次の URL を呼び出します。

**http://localhost:8080/ControlCenter/BPAdmin/Login**

ActiveWorkflow 管理者のための ActiveWorkflow ログインページが表示されます。



## サブタスク 2: OrdMgmt プロセスを起動して Start\_Order フォームから注文を登録

Start\_Order フォームは、OrdMgmt プロセスのインスタンスを作成する Start Process コマンドをトリガします。ユーザ Dave には、プロセスを作成する権限を与えています。

1. ActiveWorkflow ログインページで、次の内容を入力して、ログインボタンをクリックします。

ユーザ名 : dave  
パスワード : dave

ActiveWorkflow 管理画面が表示されます。

2. 個人ワークフローリストをクリックします。

Dave のプロセスリストが表示されます。これは、Dave が起動することのできるリストです。OrdMgmt プロセスが表示されていない場合、タスク 3 のサブタスク 4 のステップ 1 から 9 に従って追加する必要があります。

3. 個人ワークフローリストの **MyBusinessProcess/OrdMgmt** リンクをクリックします。

OrdMgmt プロセスは、インスタンスが作られて、プロセスのこのステップに関連付けられた Start\_Order フォームが表示されます。

Start\_Order - Microsoft Internet Explorer

ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)

アドレス(A) http://localhost:8080/ControlCenter.JB/packages/MyBusinessProcess/Sta 移動 リンク

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
400071	07/07/2004	300069
400000	07/07/2004	300000
400001	07/07/2004	300001
400002	07/07/2004	300002
400005	07/07/2004	300005
400006	07/07/2004	300006

Customer ID: 104120

Tracking Number: 214749364

Shipping Date: 12/31/1999

Shipping Cost: ¥15.75

New Order

Description	Price
-------------	-------

Order Number:

Total: 70.00

- 
4. Orders タブで、**New Order** ボタンをクリックします。

ORDER\_NUMBER、ORDER\_DATE、INVOICE\_NUMBER が自動的に付与された新規注文が Orders タブのデータビューに作成されます。この注文番号はデータビューの下の方にも表示されます。プロセスの後半で注文を識別するために必要になるので、注文番号は控えておきます。(タスク 3 の結果としてアクティビティリストにまだ注文がある場合)

5. product Description ドロップダウンリストから少なくとも一つの項目を選んで注文をサービス作成します。

グリッドに選択された項目が追加されるとすぐに新規の空の行が次の項目のために作成されて、選択されていることに注目してください。また、Total フィールドには、項目の価格が反映されています。

必要に応じて、繰り返し行ってください。

6. フォームの左上のところにある **Order Complete** ボタンをクリックします。

Start\_Order フォームのウィンドウは閉じられます。Dave が入力した注文は登録されて、プロセスの次のアクティビティ Receive Order に移動します。

## サブタスク 3: Receive\_Order フォームを使用して注文を承認

1. ActiveWorkflow 管理者をログアウトして、営業グループのメンバの Alice でログインします。ActiveWorkflow ログインページで、次の入力をしてログインボタンをクリックします。

ユーザ名 : alice  
パスワード : alice

2. アクティビティリストタブをクリックします。

ReceiveOrder アクティビティを含むアクティビティリストが表示されます。

3. 名前列にある **MyBusinessProcess/Receive Order** リンクをクリックします。

Receive\_Order フォームが表示されます。注文情報は Orders タブに表示されます。

Process Information

CUST\_CD: 104120  
Order\_Number: 400024  
Process\_Invoker: Dave  
Process\_Title: Order # 400024

PayType: PO

Cancel Approved

Orders Customer Information Billing Address Shipping Address

ORDER_NUMBER	ORDER_DATE	INVOICE_NUMBER
400024	11/26/2007	300024

Customer ID: 104120  
Tracking Number: 0  
Shipping Date:   
Shipping Cost: ¥ 0.00

Description	Price
UNIFY NXJ Design Center - Enterprise Edition	200

Order Number: 400024 Total: 7200.00

powered by UNIFY NXJ  
イントラネット

4. Receive\_Order フォームで、**Approved** ボタンをクリックします。

Receive\_Order フォームウィンドウは閉じられます。Alice が承認した注文は、次のアクティビティ Fill Order に移動します。

- 
5. アクティビティリストページをリフレッシュします。

注文は、Alice のリストから削除されます。

## サブタスク 4: Fill\_Order フォームを使用して 注文の出荷状況を示す

1. ActiveWorkflow 管理者をログアウトして、出荷グループのメンバーの Bob でログインします。ActiveWorkflow ログインページで、次の入力をしてログインボタンをクリックします。

ユーザ名 : bob

パスワード : bob

2. アクティビティリストタブをクリックします。

Fill Order アクティビティを含むアクティビティリストが表示されます。

3. 名前列の **MyBusinessProcess/Fill Order** リンクをクリックします。

Fill\_Order フォームが表示されます。注文情報は、Order タブに表示され  
ます。

Fill\_Order - Microsoft Internet Explorer

ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)

アドレス(D) http://localhost:8080/ControlCenter JB/packages/MyBusinessProcess/Fil 移動 リンク

Process Information

CUST\_CD: 104120  
Order\_Number: 400024  
Process\_Invoker: Days  
Process\_Title: Order # 400024

PayType: Purchase Order

Cancel Filled

Orders Customer Information Billing Address Shipping Address

ORDER_NUMBER	ORDER_DATE	INVOICE_NUMBER
400024	11/26/2007	300024

Customer ID: 104120  
Tracking Number: 0  
Shipping Date:  
Shipping Cost: ¥ 0.00

Description	Price
UNIFY NXJ Design Center - Enterprise Edition	200

Order Number: 400024 Total: ?200.00

powered by UNIFY N  
ページが表示されました イン트라ネット

4. Fill\_Order フォームで、**Filled** ボタンをクリックします。

Fill\_Order フォームウィンドウは閉じられて、プロセスは Receive Payment アクティビティに移動します。

5. アクティビティリストページをリフレッシュします。

注文は Bob のリストから削除されます。

---

## サブタスク 5: Receive\_Payment フォームで注文の 代金回収状況を示す

1. ActiveWorkflow 管理者をログアウトして、経理グループのメンバーの Charlie でログインします。ActiveWorkflow ログインページで、次の入力をしてログインボタンをクリックします。

ユーザ名 : charlie  
パスワード : charlie

2. アクティビティリストタブをクリックします。

Receive Payment アクティビティを含むアクティビティリストが表示されます。

3. **MyBusinessProcess/Receive Payment** リンクをクリックします。

Receive\_Payment フォームが表示されます。注文情報は Order タブに表示されます。

The screenshot shows a web browser window titled "Receive\_Payment - Microsoft Internet Explorer". The address bar shows "http://localhost:8080/ControlCenterJB/packages/MyBusinessProcess/Re". The form contains the following sections:

- Process Information:** A box containing "cust\_cd: 104120", "Order\_Number: 400024", "Process\_Invoker: Dave", and "Process\_Title: Order # 400024".
- PayType:** A dropdown menu set to "Purchase Order" with "Cancel" and "Done" buttons below it.
- Orders:** A tabbed interface with "Orders" selected. It contains a table with columns "ORDER\_NUMBER", "ORDER\_DATE", and "INVOICE\_NUMBER".
- Customer Information:** Fields for "Customer ID" (104120), "Tracking Number" (0), "Shipping Date", and "Shipping Cost" (¥ 0.00).
- Item List:** A table with columns "Description" and "Price".
- Summary:** "Order Number" 400024 and "Total: 7200.00".

ORDER_NUMBER	ORDER_DATE	INVOICE_NUMBER
400024	11/26/2007	300024

Description	Price
UNIFY NXJ Design Center - Enterprise Edition	200

Order Number: 400024      Total: 7200.00

4. Receive\_Payment フォームで、**Done** ボタンをクリックします。  
Receive\_Payment フォームウィンドウは閉じられます。注文処理は完了します。
  5. アクティビティリストページをリフレッシュします。  
注文は Charlie のリストからは削除されます。
- これで、ActiveWorkflow チュートリアルを終了します。