TIBCO iProcess® Modeler

Getting Started

Software Release 11.3.1
July 2012

two-second advantage™

TIBCO®
The Power of Now®

Important Information

Contents

Preface

This guide explains and demonstrates how to create basic iProcess procedures using a combination of the following products:

- TIBCO iProcess® Workspace (Windows)
- TIBCO iProcess Modeler

Topics

Changes from the Previous Release of This Guide

This section itemizes the major changes from the previous release of this guide.

There are no changes from the previous release of this guide.

Related Documentation

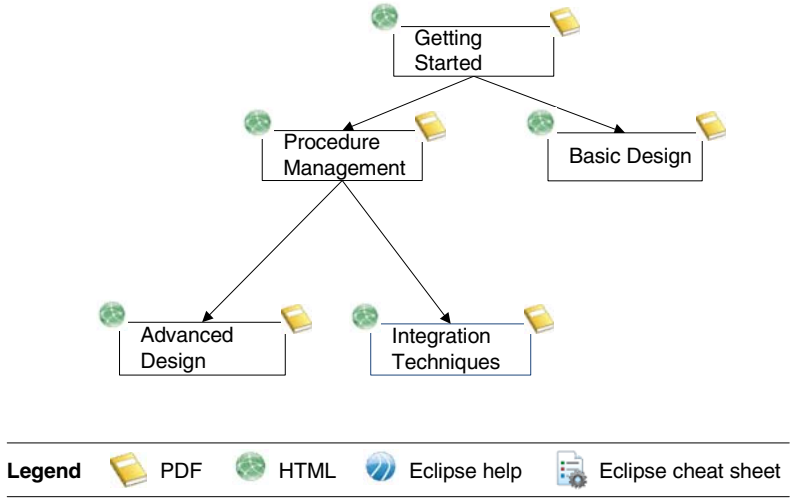
This section lists documentation resources you may find useful.

TIBCO iProcess Modeler Documentation

The following documents form the TIBCO iProcess Modeler and TIBCO iProcess Workspace (Windows) documentation set, which are supplied with the TIBCO iProcess Workspace (Windows) software:

- *TIBCO iProcess Workspace (Windows) Installation* Read this manual for instructions on site preparation and installation.
- *TIBCO iProcess Workspace (Windows) Release Notes* Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.
- TIBCO iProcess Suite Documentation This documentation set contains all the manuals for TIBCO iProcess Modeler, TIBCO iProcess Workspace (Windows), and other TIBCO products in TIBCO iProcess® Suite. The manuals for TIBCO iProcess Modeler and TIBCO iProcess Workspace (Windows) are as follows:
  - *TIBCO iProcess Workspace (Windows) User's Guide*
  - *TIBCO iProcess Modeler Getting Started*
  - *TIBCO iProcess Modeler Procedure Management*
  - *TIBCO iProcess Modeler Basic Design*
  - *TIBCO iProcess Modeler Advanced Design*
  - *TIBCO iProcess Modeler Integration Techniques*
  - *TIBCO iProcess Expressions and Functions Reference Guide*
  - *TIBCO iProcess Workspace (Windows) Manager's Guide*

If you are new to iProcess procedure development, you are advised to follow the reading path shown next. The documentation road map shows the relationships between the books and online references in this product's documentation set.



Other TIBCO Product Documentation

You may find it useful to read the documentation for the following TIBCO products:

- TIBCO ActiveMatrix BusinessWorks™
- TIBCO Business Studio™
- TIBCO Enterprise Message Service™
- TIBCO Hawk®
- TIBCO Rendezvous®

Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

Convention	Use
SWDIR	TIBCO iProcess Engine installs into a directory. This directory is referenced in documentation as *SWDIR*. The value of *SWDIR* depends on the operating system. For example, • on a Windows server (on the `C:` drive) if *SWDIR* is set to the `C:\swserver\staffw_nod1` directory, then the full path to the `swutil` command is in the `C:\swserver\staffw_nod1\bin\swutil` directory. • on a UNIX or Linux server if *SWDIR* is set to the `/swserver/staffw_nod1` directory, then the full path to the `swutil` command is in the `/swserver/staffw_nod1/bin/swutil` directory or the `$`*SWDIR*`/bin/swutil` directory. **Note**: On a UNIX or Linux system, the environment variable $*SWDIR* should be set to point to the iProcess system directory for the *root* and *swadmin* users.
`code font`	Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example: Use `MyCommand` to start the foo process.
`bold code font`	Bold code font is used in the following ways: • In procedures, to indicate what a user types. For example: Type **`admin`**. • In large code samples, to indicate the parts of the sample that are of particular interest. • In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, `MyCommand` is enabled: `MyCommand [`**`enable`**` | disable]`

Table 1 General Typographical Conventions (Cont'd)

Convention	Use
italic font	Italic font is used in the following ways: • To indicate a document title. For example: See *TIBCO ActiveMatrix BusinessWorks Concepts*. • To introduce new terms. For example: A portal page may contain several portlets. *Portlets* are mini-applications that run in a portal. • To indicate a variable in a command or code syntax that you must replace. For example: `MyCommand` *PathName*
Key combinations	Key name separated by a plus sign indicate keys pressed simultaneously. For example: Ctrl+C. Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q.
	The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.
	The tip icon indicates an idea that could be useful, for example, a way to apply the information provided in the current section to achieve a specific result.
	The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.

Table 2 Syntax Typographical Conventions

Convention	Use
`[ ]`	An optional item in a command or code syntax. For example: `MyCommand [optional_parameter] required_parameter`
`|`	A logical `OR` that separates multiple items of which only one may be chosen. For example, you can select only one of the following parameters: `MyCommand param1 | param2 | param3`

Table 2 Syntax Typographical Conventions (Cont'd)

Convention	Use

`{ }`	A logical group of items in a command. Other syntax notations may appear within each logical group.
	
	For example, the following command requires two parameters, which can be either the pair `param1` and `param2`, or the pair `param3` and `param4`.
	
	`MyCommand {param1 param2} | {param3 param4}`
	
	In the next example, the command requires two parameters. The first parameter can be either `param1` or `param2` and the second can be either `param3` or `param4`:
	
	`MyCommand {param1 | param2} {param3 | param4}`
	
	In the next example, the command can accept either two or three parameters. The first parameter must be `param1`. You can optionally include `param2` as the second parameter. And the last parameter is either `param3` or `param4`.
	
	`MyCommand param1 [param2] {param3 | param4}`

Connecting with TIBCO Resources

How to Join TIBCOmmunity

TIBCOmmunity is an online destination for TIBCO customers, partners, and resident experts. It is a place to share and access the collective experience of the TIBCO community. TIBCOmmunity offers forums, blogs, and access to a variety of resources. To register, go to http://www.tibcommunity.com.

How to Access TIBCO Documentation

You can access TIBCO documentation here:

http://docs.tibco.com

How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, contact TIBCO Support as follows:

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit this site:

  http://www.tibco.com/services/support

- If you already have a valid maintenance or support contract, visit this site:

  https://support.tibco.com

  Entry to this site requires a user name and password. If you do not have a user name, you can request one.

Chapter 1 Introduction

This chapter gives an overview of how to define procedures and explains the components of iProcess Suite you need to use.

Topics

Back to Library

TIBCO iProcess Workspace (Windows)

The TIBCO iProcess Workspace (Windows) is the single focal point for defining and managing procedures. The TIBCO iProcess Workspace (Windows) consists of:

- The Work Queue Manager
- The Procedure Manager

The Work Queue Manager displays users' queues and work items and the Procedure Manager displays all of the procedures currently available. From the Procedure Manager, you can:

- create and edit procedures. The TIBCO iProcess Modeler is started from the TIBCO iProcess Workspace (Windows).
- organize and manage your procedures as a hierarchical structure of procedure libraries, in the same way as, for example, you manage files and directories.

Back to Library

TIBCO iProcess Modeler

TIBCO iProcess Modeler has one primary objective and that is to make the analysis and documentation of business processes as simple and intuitive as possible. It is aimed at a non-technical audience and gives you a visual representation of your business process that is easy to follow and that can be enhanced or amended at any time. TIBCO iProcess Modeler builds on the familiar flowchart metaphor to show in an unambiguous manner, the flow of work for a particular business process. The rules that you define graphically are stored by the iProcess Engine node and can then be deployed across a wide ranging hardware architecture.

TIBCO iProcess Modeler is automatically started by TIBCO iProcess Workspace (Windows) when you want to create or edit procedures. From TIBCO iProcess Modeler you can access *Step Definer*, which enables you to design the forms for each step in your procedure. The forms are the part of the step seen by the person who receives the work item in their queue. The forms contain text and fields into which users can enter information for a particular case, or instance, of a procedure. Fields can also display information already provided or can hold information calculated by the procedure.

Back to Library

Defining Procedures

Procedures are at the very heart of iProcess Suite operations, and, as a procedure/process definer, your role is vital to the successful use of iProcess Suite. You are the one who will decide how to translate manual procedures and paper documents into iProcess procedures and forms. In effect, you will be using the iProcess Suite to generate custom software applications designed to meet the specific needs of your organization.

As a procedure definer, you will need to have a basic understanding of how the iProcess Suite works. Since you will be designing forms for both users and managers, you will want to know how the iProcess Suite looks from their perspectives.

In brief, the iProcess Suite works by integrating electronic mail with the management of text, forms and data. It constructs messages, memos, letters, or complex documents with text and blanks to be filled in and routes them to the appropriate individuals according to the rules that you establish.

By definition, an iProcess procedure is a series of **steps** that accomplish a task. Each step consists of five possible parts:

- An **addressee** (user, group or role)
- A **form** the addressee receives
- An **action** that takes place after the addressee completes the form
- A **deadline** by which the addressee must complete the form
- The **duration** between the step being active and released.

The flow of the procedure (from one step to the next) is controlled by a series of **conditions** and **actions** that you specify. An iProcess step could be compared to sending a blank or partially filled in form, or a memo that requests information; and then, on the basis of that information, deciding what to do next to accomplish the overall goal.

As a procedure definer, you will be working closely with the iProcess system administrator, the person who is responsible for maintaining all central iProcess data, including **users** and **groups, attributes**, **roles**, **lists** and **tables**.

You will use users, groups and roles when you define addressees. You will use attributes, lists and tables when you define the forms the addressees will receive.

Back to Library

Regardless of the type of system you use, all work that you do as a procedure definer will be stored in a special work space set aside for this purpose. You will be free to create and test procedures in a realistic environment without affecting the iProcess Suite's normal operations. When you are satisfied that your procedure accomplishes its goal, you can **release** the procedure and make it available to other iProcess users.

Back to Library

Overview of Creating a Procedure

The process of creating an iProcess procedure can be divided into five steps:

1. *Decide where the procedure should be stored in your procedure management hierarchy.* Procedures can be stored in procedure libraries which you can use to organize your procedure objects.
2. *Plan the procedure*. This involves studying the manual procedure and breaking it down into its components. These will be the steps in your iProcess procedure. This process also involves identifying addressees and designing forms.
3. *Create the steps that make up the procedure*. This includes fully defining addressees, forms, actions, and deadlines.
4. *Test and edit the procedure until it accomplishes its goals*. During the testing process, which will emulate actual cases of your procedure, you will be able to see what your procedure users will see when they access their work queues. The editing process uses exactly the same options as creating a procedure does to refine and adjust the new procedure.
5. *Release the procedure for general use*. Even after you release your procedure you will remain the procedure owner on record, with the ability to make changes if necessary and to respond to exceptional cases.

Back to Library

Chapter 2 Getting Started

This tutorial covers the basics of defining a procedure using Procedure Manager, iProcess Modeler and Step Definer. It takes approximately one hour to complete. After completing this tutorial, you will be able to perform the following tasks:

Topics

Back to Library

Displaying Procedure Manager

To display Procedure Manager in Work Queue Manager, do one of the following:

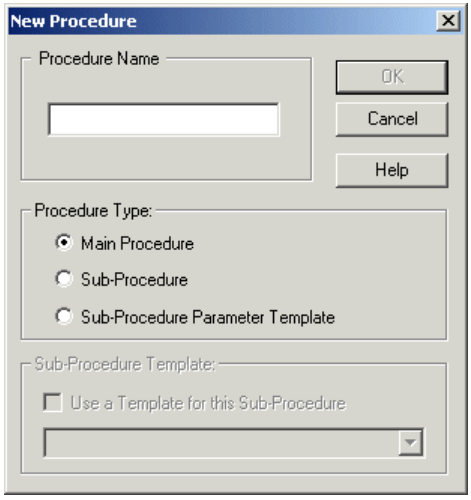
- Click **Window** > **View Procedure Manager** to display Procedure Manager in place of Work Queue Manager.
- Click **Window** > **Split Left Hand Window** to split the left hand window in Work Queue Manager to display both the work queues list and Procedure Manager.

Back to Library

Creating a Procedure

To create a new procedure, do the following:

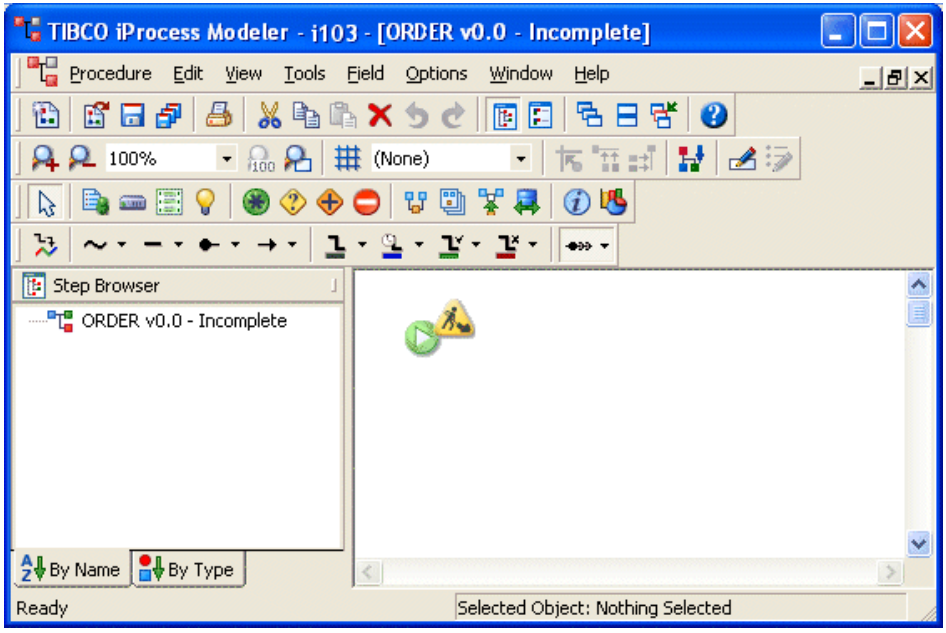
1. From the **Procedure Manager** window, double-click the **Procedure Management** library. The list of available libraries is displayed.
2. Navigate to the library where you want to create the new procedure.
3. Click **Procedure Management** > **New Procedure**. The **New Procedure** dialog is displayed.



4. Type **ORDER** in the **Procedure Name** field.
5. Make sure that **Main Procedure** is selected as the Procedure Type.

Back to Library

6. Click **OK**. A **Process Definer** window is started.



Back to Library

Defining the Procedure

This section describes how to define a simple two step procedure. It covers the following topics:

- Defining a Step
- Defining the Form
- Creating a Second Step
- Linking Steps
- Deleting a Link
- Moving Steps
- Saving the Procedure

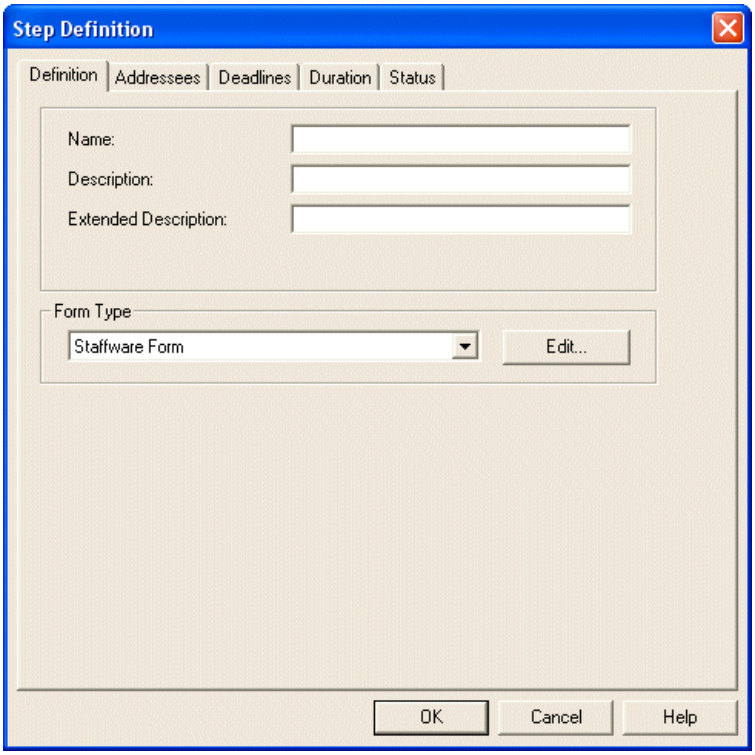
Defining a Step

This section describes how to define the first step of the procedure.

1. Click the **Step** tool  on the toolbar to select it, position the pointer where you want the step to appear (typically, just to the right of the **Start** icon) and click to place the step.

Back to Library

The **Step Definition** dialog is displayed. (If you have TIBCO iProcess™ Client (VBA) installed, the **Step Definition** dialog includes extra options).



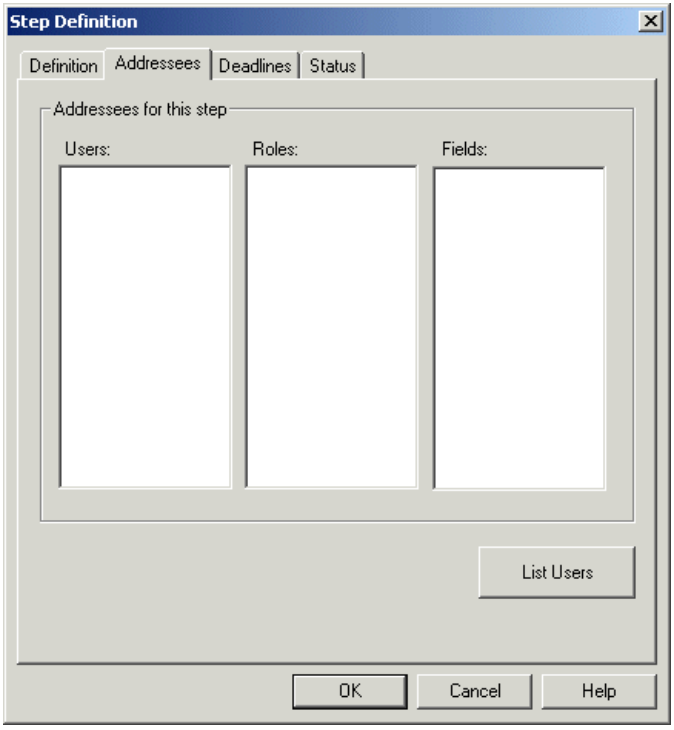
2. Enter the **Name** you want to call the step.

   In this example use the name **DETAILS**. It doesn't matter if you use lower or upper case for the name as iProcess automatically converts it to upper case. The name can be a maximum of 8 characters with no spaces.

3. Enter the **Description,** which can be up to 24 characters. Describe this step as **Receive Order Details**. The description will remain in lower or upper case just as you type it.

Back to Library

4. Click the **Addressees** tab. Every step must have an **Addressee**. An Addressee is the queue that will receive this step when a case of the procedure is run.



There are three ways to specify who the addressee(s) of a step should be - Users, Roles and Fields. These are all covered in depth in "Defining a Procedure" in the *TIBCO iProcess Modeler - Basic Design* guide but for this example you only need the **Users** column. Under the **Users** column you can enter the name of the queue (either a user or group) that will receive this step.

5. Enter your own iProcess user name and click the **Definition** tab to return to the **Step Definition** dialog.

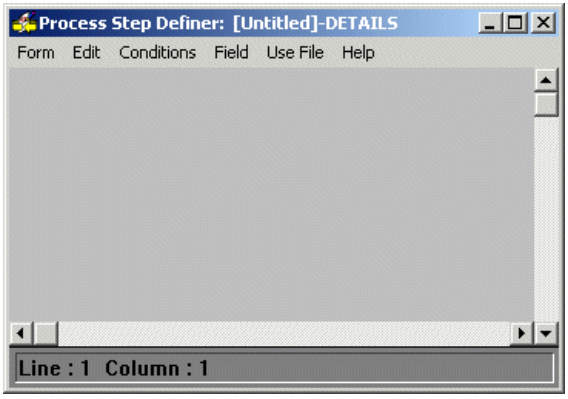
To be designated as an addressee, you must be a valid iProcess user or group. Users and groups are added in User Manager in TIBCO iProcess Administrator. See Managing Users in *TIBCO iProcess Workspace (Windows) Manager's Guide*.

Back to Library

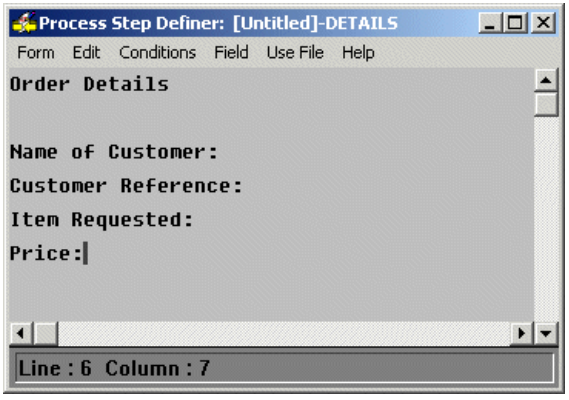
Defining the Form

The next stage is to design the form for the step. The form is what appears at run-time when the work item is selected from a queue or, if it is the first step in the procedure, when a case is started. A form can contain text and also fields where the user can enter information.

1. Click **Edit** in the iProcess Form section from the **Step Definition** dialog. The **Process Step Definer** is displayed.



2. Enter the following text on the form:

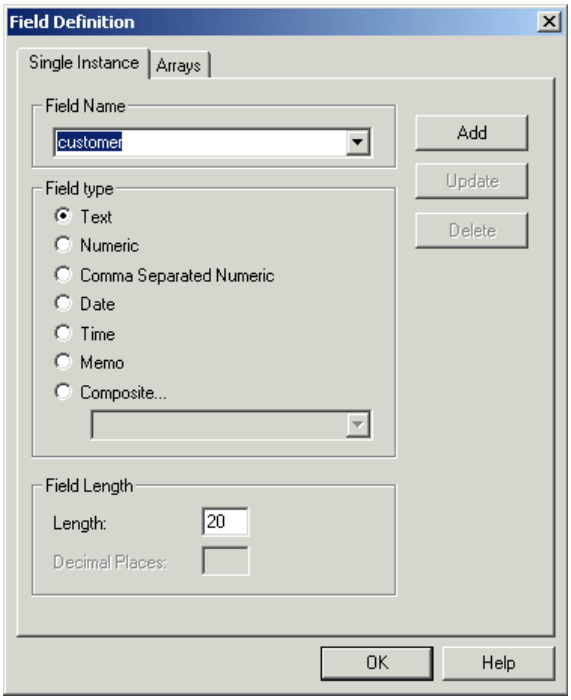


Defining Fields

As you want the user of the procedure to enter the customer name, you must define a field that they can type into when they are filling in the form. To do this:

Back to Library

1. Click **Field > Define...** The **Field Definition** dialog is displayed.



2. In the **Field Name** box, type **Customer**. This will be the name of your first field.

Field names can be up to 15 characters long and can contain letters, digits and underscore characters but they must start with a letter. Field names are converted to uppercase irrespective of how they are originally entered.

All of the different field types are discussed in detail in "Creating Fields and Forms" in the *TIBCO iProcess Modeler - Basic Design* guide.

3. For the **Customer** field, keep the default field type of **Text**.
4. To change the field length, click in the **Length** box at the bottom of the window and change the value from **20** to **30**.
5. Click **Add**. The field details you entered are saved and you can define another field.

Back to Library

6. Define three more fields:
   - A text field called **Custref** with a length of 12.
   - A text field called **Item** with a length of 25.
   - A numeric field called **Price**. The length of **Price** should be 8 with 2 decimal places.

The length of a numeric field includes the decimal point and decimal places, for example, a length of 8 gives us 00000.00.

7. Click **OK** when you have finished adding fields. You can always add more fields later if you need to.

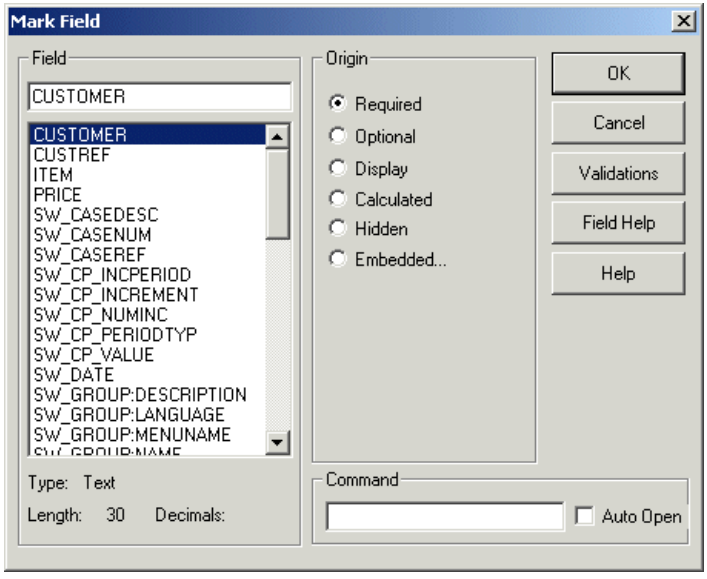
Inserting Fields into the Form

You can now place the fields you have defined on the form. To do this:

1. Place the cursor a few spaces after the text **Name of Customer**:

You can use TAB and SPACE to move the cursor to the position on the line where you want the field to start.

2. Click **Field > Insert**. The **Mark Field** dialog is displayed.



3. Click **Customer** from the **Field** list.

Back to Library

4. Click **Required** for the field's **Origin**.

   The **Origin** of a field defines how the field is going to be used in the form. The origins are:

Origin	Meaning
Required	The user must fill in this field. If a Required field is not completed the form cannot be released. In TIBCO iProcess Workspace (Windows) required fields appear on the form in red.
Optional	The user can enter information into this field but does not have to. The form can be released without an optional field being filled in. Optional fields appear in blue (white at runtime).
Display	The current value for the field is displayed in the form but it cannot be changed.
Calculated	A value for the field is calculated and displayed based on a specified calculation. See "Controlling Data Input on Forms" in the *TIBCO iProcess Modeler - Basic Design* guide for information on using field calculations.
Hidden	A value is calculated but not displayed in the form. See "Controlling Data Input on Forms" in the *TIBCO iProcess Modeler - Basic Design* guide for information on using field calculations.
Embedded	Displays the value of the field (but not any remaining spaces up to the length of the field). This origin is often used in letters as at run-time it appears as part of the form body rather than a field. See "Controlling Data Input on Forms" in the *TIBCO iProcess Modeler - Basic Design* guide for information on using embedded and ampersanded fields.

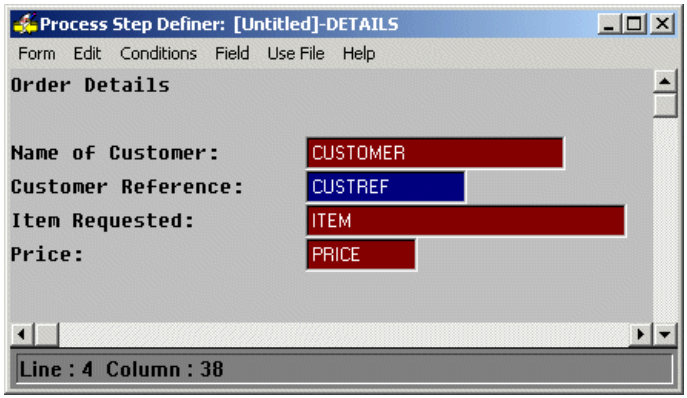
5. Click **OK**. The field is displayed at the cursor position.

Back to Library

6. Repeat this for the rest of the fields:

Field	Origin
Custref	Optional
Item	Required
Price	Required

Your form should now look like this:



If you can't see the field names, choose **Form > Setup > Show Field** names.

Saving the Form

It is important to save your form at regular intervals to ensure that you don't lose any of your work in the event of a power failure or other problem:

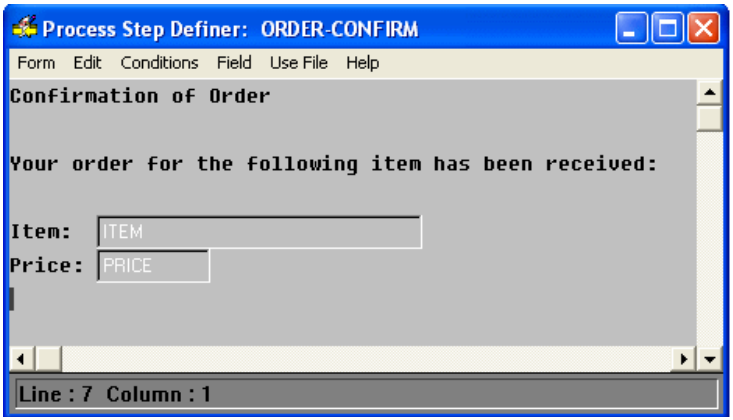
1. Click **Form** > **Save Changes**. The form is saved.
2. Click **Form** > **Exit** to close the **Step Definer** and return to the iProcess Modeler.

Creating a Second Step

Now that your first step is complete, you can create a second step.

Back to Library

1. Following the steps described in Defining a Step on page 11, create a step using a step name of **CONFIRM** and a description of **Confirmation of Order**. For the Addressee, choose the name of a colleague who is a user on the iProcess system.
2. In the form, enter the following text:



3. Insert the two fields, **Item** and **Price,** after the text Item and Price, but this time choose the origin as **Display** (see "Creating Fields and Forms" in the *TIBCO iProcess Modeler - Basic Design* guide for more information about inserting fields). **Display** can be used on this step as the value for these fields will have been entered in the previous step.
4. Save and close the form to return to the iProcess Modeler.

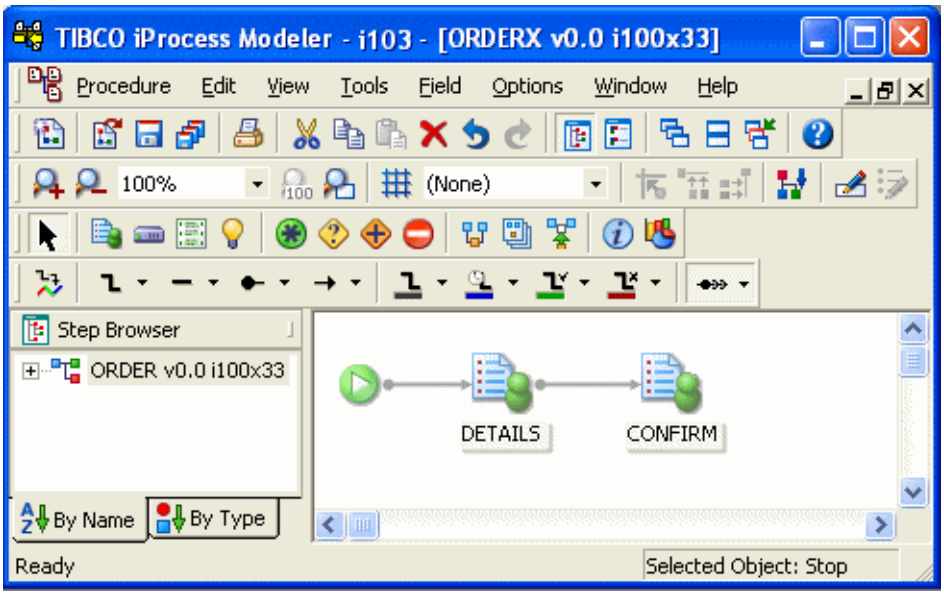
Linking Steps

Now that you have two steps in your procedure, you need to link them together to define the order in which iProcess should process them. You must also link the first step to the **Start** icon. (See "Defining a Procedure" in the *TIBCO iProcess Modeler - Basic Design* guide for more information on linking objects). The link tool is automatically selected when you move the cursor near an object. The cursor and description change when you move the cursor to the correct position to add a link.

1. The simplest way to link the two steps is to click on the right side of the **Start** icon and drag the cursor to the left side of the first step, **DETAILS**, and release.
2. Click on the right side of the first step, **DETAILS**, and drag the cursor to the left side of the second step, **CONFIRM**, and release.

Back to Library

Your procedure will now look like this.



When you link two steps in this way, the second one becomes an action that is carried out when the first step is released.

Deleting a Link

If you create a link in the wrong place, for example, connecting to the top or bottom of an icon, it can have a different meaning (see "Defining a Procedure" in the *TIBCO iProcess Modeler - Basic Design* guide for more information on linking objects). You can delete any link by selecting the line then pressing the **Delete** key.

Moving Steps

You can move the steps in your procedure by selecting the step or steps, then dragging and dropping the objects to a new position on the iProcess Modeler layout. Any links to other steps will automatically be re-drawn.

Saving the Procedure

You should now save your procedure. To do this:

1. From the iProcess Modeler, click **Procedure > Save** or click  .

Back to Library

2. If the **Save Comment** dialog is displayed, enter a comment describing the reason why the procedure has been created.

Back to Library

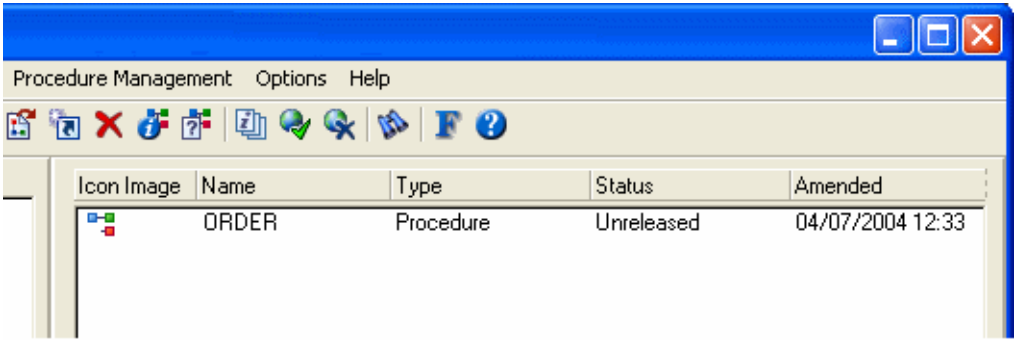
Closing TIBCO iProcess Modeler

When you are ready to close iProcess Modeler, click **Procedure > Exit**. If you have a procedure open that you have made changes to, you are asked if you want to save the procedure.

Back to Library

Checking the Procedure Status

Every procedure has a status depending on its stage of development and working life. From the **Procedure Manager**, the Procedure Details list in the right hand side of the window shows the procedure objects in the currently selected library. By default the **Status** of each object in the library is shown. If you navigate to the library that contains your procedure, you will see that the status of the procedure is **Unreleased**.



When you first define a procedure it is only available to you, the procedure definer, as it is under development. When you run a case of an unreleased procedure, all work items go to test queues belonging to you. See *Testing a Procedure* on page 24.

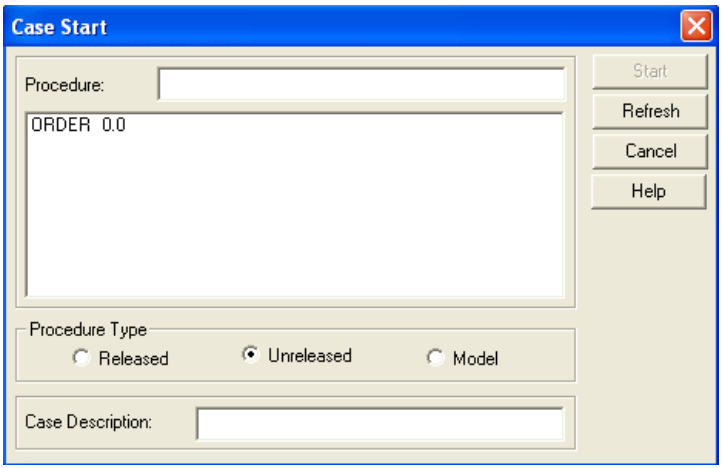
The iProcess Administrator user has access to all procedures irrespective of who owns them.

Back to Library

Testing a Procedure

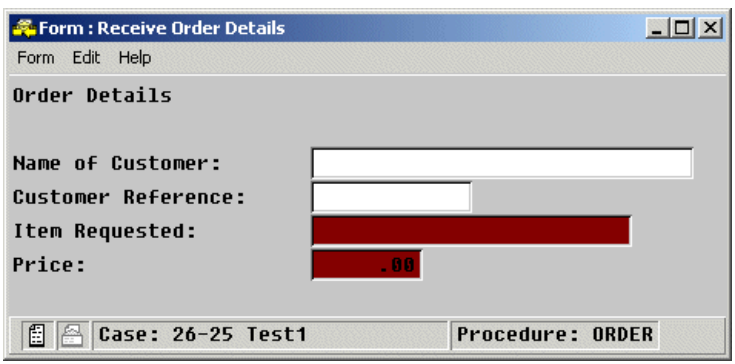
Now that you have a complete two-step procedure, you need to test it to make sure that it is correct before releasing it for other iProcess users to use.

1. From **Work Queue Manager**, click the **Case Start** button. 
2. The **Case Start** dialog is displayed. In the **Procedure Type** section of the dialog, select **Unreleased**. All available procedures of that type are listed (along with the version number).



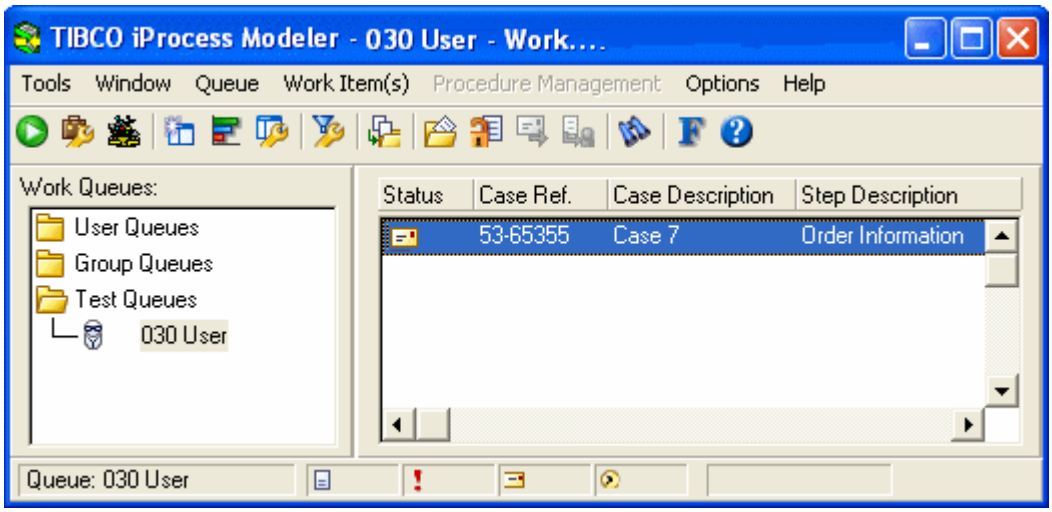
3. Click the **ORDER** procedure and enter a **Case Description**. The Case Description is used to identify the case and is usually displayed in the Work Queue Manager.
4. Click **Start**.

   The form of the first step is displayed for you to complete and release.



Back to Library

This step will also appear in your work queue in the test queues folder.



You may need to click the **Rebuild List** button, or click **Rebuild Queues List** from the **Queues** menu, to see your test queue.

Because your procedure is unreleased, all the work items related to the procedure will go to your test queues. The test queues are named after the addressee of the step.

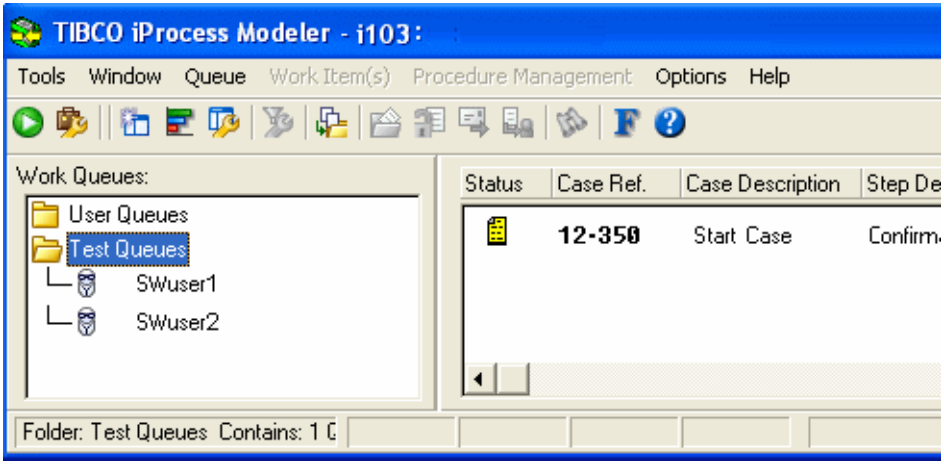
Test queues are completely separate from your own work queue containing work items from released procedures. Test queues are not visible to any other users.

5. Enter information into the fields in the first step, then release it by double clicking the release icon  in the lower left corner of the form.

The release icon is not available until you have filled in all required fields.

Back to Library

6. The **Work Queue** now shows a second test queue that contains the second step of your procedure.



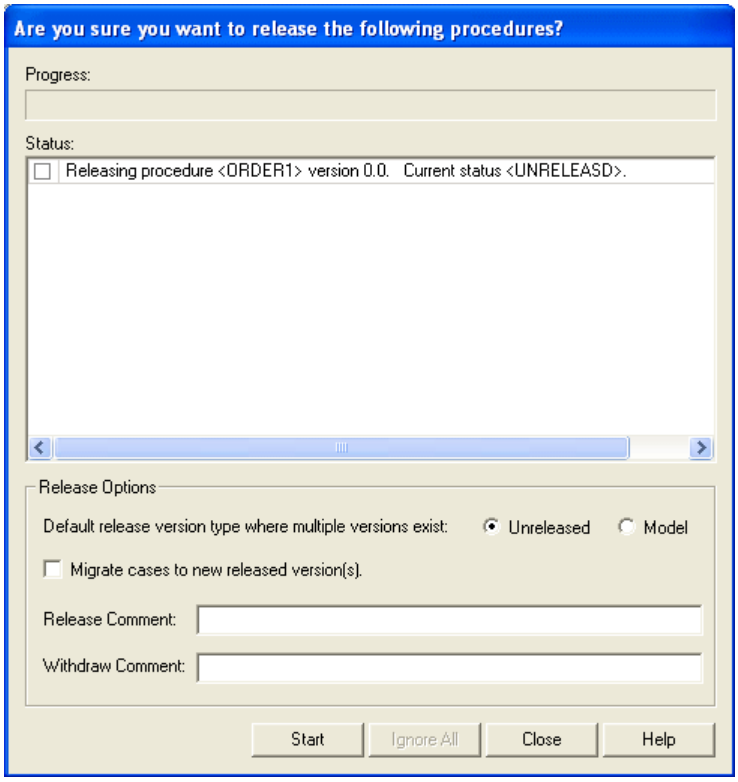
You can open the **Work Item** and release it straight away as there are no input fields on the form.

Back to Library

Releasing a Procedure

Once you have tested your procedure and have verified that it works correctly (for example, that the forms look the way you want them, that the steps go to the correct addressees, that the information being collected is shown correctly), you can return to the Procedure Manager and release the procedure.

1. In the Procedure Manager, select the **ORDER** procedure.
2. Click **Procedure Management > Release Procedure**. A dialog is displayed showing the procedures and versions that will be released and withdrawn if you continue.



3. Enter a **Release Comment** describing why the procedure is being released.
4. Click **Start**. The progress meter shows the progress of the operation, and a tick appears next to the procedure as it is released.
5. Click **Close** to close the dialog when the procedure has been released.

Your procedure is now available to all iProcess users on this iProcess installation.

Back to Library

Where To Find More Information On Defining Procedures

You can find more information about defining procedures from the following guides.

- The *TIBCO iProcess Modeler - Basic Design* guide explains basic procedure design including:
  - the types of procedure objects that can be placed in a procedure and how to link them together.
  - creating fields and forms that can be attached to a step.
  - performing calculations on fields.
  - defining a deadline in a procedure.
  - setting up a condition so that a branch in the process flow can be added.
  - defining a withdraw action so that a step can be withdrawn from the work queue if it is no longer required.
  - defining a wait in a procedure.
  - making procedures easier to follow.
  - customizing the Process Definer.
  - troubleshooting procedure definitions.
- The *TIBCO iProcess Modeler - Advanced Design* guide explains more complex procedure definition design including:
  - defining sub-procedures.
  - defining sub-procedure calls.
  - the use of array fields which are used in conjunction with dynamic sub-procedures and graft steps.
  - creating templates for use with dynamic sub-procedure calls and graft steps.
  - defining a dynamic call to multiple sub-procedures.
  - using scripts.
  - using case prediction.
  - setting priorities and escalating work items.

Back to Library

- The *TIBCO iProcess Modeler - Procedure Management* guide explains how to manage your libraries and procedures including:
  - managing procedures, libraries and shortcuts.
  - managing procedure versions.
  - setting and/or viewing properties, access controls and status information for different types of procedure objects.
- The *TIBCO iProcess Modeler - Integration Techniques* guide explains how to integrate your procedure with other external applications such as databases or custom applications including:
  - an introduction to the iProcess Suite's integration layers and the integration options available.
  - using EAI steps in your procedures to control updates to external systems and iProcess case data under transactional control.
  - how to set up graft steps so that external applications can dynamically start sub-procedures and "graft" them to the main procedure at run-time.
  - how to define public and event steps that can be published by external applications.
  - how to use an external form application in place of the iProcess Modeler's own form window for one or more steps of a procedure.
  - how to open a form window for any step of a procedure without starting or accessing a case.
  - how to set up event steps so that an action will occur when a particular event happens.
  - how you can create reports on iProcess case data and then view them using a third party data viewer application.
  - the types of command you can use to help integrate iProcess with your applications. For example, you can use a command to run an external program.

Back to Library

Back to Library

Index

S

T