WebSphere® Partner Agreement Manager

# User's Guide

*Version 2 Release 2*

**Note:** Before using this information and the product it supports, read the information in *Notices* on page 235.

# TABLE OF CONTENTS

# Welcome to the User's Guide

This guide describes WebSphere® Partner Agreement Manager and explains how to start a Partner Agreement Manager session, design public and private processes, define element definition sets, create business objects, and manage process distribution.

**To use Partner Agreement Manager, follow these general steps:**

- Read *Introducing Partner Agreement Manager* on page 1 before you begin.
- Become familiar with the Process Manager window, which enables you to communicate with your business partners to exchange information. See *Partner Agreement Manager basics* on page 7.
- Analyze existing business processes and identify the workflow between partners. See *Designing processes* on page 15.
- Create element definition sets and business object types, which identify the type of information to be exchanged with your partners. See *Defining business object types* on page 27.
- Create maps that use information from another source to populate a business object type, or to move information from one business object type to another. See *Mapping business objects* on page 59.
- Design the public process, which determines the flow of messages and information between partners. See *Defining public processes* on page 93.
- Distribute the public process to your partners for approval. See *Managing processes* on page 203.

- Define private processes, to determine the actions that each partner takes. See *Defining private processes* on page 139.
- Install the public process and start running it. See *Managing processes* on page 203.

## WHO SHOULD USE THIS INFORMATION

This information is for those who need to design and implement processes between business partners.

## RELATED INFORMATION

For additional information see the following:

- The readme.htm file. This file contains information that became available after this book was published. Before installation, the readme.htm file is located in the root directory of the product CD-ROM. After installation, the readme.htm file is located in the root directory of the Partner Agreement Manager installation.
- The StartHere.htm file. This file contains links to the Partner Agreement Manager readme.htm file and the *Partner Agreement Manager Installation Guide*. Before installation, the StartHere.htm file is located in the root directory of the product CD-ROM. After installation, the StartHere.htm file is located in the root directory of the Partner Agreement Manager installation.
- The *Partner Agreement Manager Installation Guide*, form number GC34-5964-02, which describes how to install Partner Agreement Manager.
- The *Partner Agreement Manager Administrator's Guide*, form number BIAAAB02, which describes how to set up, configure, and administer Partner Agreement Manager after you install it.
- The *Partner Agreement Manager Adapter Developer's Guide*, form number BIAAAD02, which describes how to develop and administer adapters using the Partner Agreement Manager Adapter Development Environment.
- The *Partner Agreement Manager Script Developer's Guide*, form number BIAAAE02, which describes how to write scripts used in Partner Agreement Manager private processes and elsewhere.

- The *Partner Agreement Manager External API Guide*, form number BIAAAF02, which describes principles behind the Partner Agreement Manager External API. See also the Javadoc for the External API, which is installed in the Partner Agreement Manager Docs folder.

- The *Partner Agreement Manager Adapters for MQSeries User's Guide*, form part number BIAAAG02, which describes how to install, configure, and run the Partner Agreement Manager Adapters for MQSeries.

- The *Partner Agreement View User's Guide*, form number GC34-5965-02, which describes how to install, configure, and use Partner Agreement View.

# Summary of changes

This edition includes these changes since the previous, first, edition:

- *External APIs. Partner Agreement Manager* 2.2 provides added flexibility to external applications through additional APIs. These APIs allow third-party applications to take advantage of the Partner Agreement Manager partner management and process engine through programmatic access. The API is distributed as a set of Java classes that the external application can import. Communication between the API classes and the Process Server is through RMI, but in the future can be swapped out for HTTP or SOAP. Specifically, APIs have been added to the following functional areas:

  - Session Service API
  - Admin Service API
  - Document Service API
  - Partner Service API
  - Adapter Service API
  - Process Service API

- *LDAP Support. Partner Agreement Manager* 2.2 provides centralized user authentication and administration through an LDAP directory. Partner Agreement Manager can retrieve user information—such as name, e-mail address, phone, and fax—stored in an LDAP directory. Updating this information is done in a single place, through the LDAP management tool. Users are authenticated through the same directory, giving them single-sign-on capabilities across enterprise applications.

- *Double-byte character sets (DBCS) and National Language Support (NLS).* Double-byte character sets are now supported in Partner Agreement Manager 2.2. Double-byte and multibyte data can be transferred and operated on in business objects and adapters. NLS lets Partner Agreement Manager display user interface text in other languages.

- *Improved XML Support.* The Partner Agreement Manager 2.2 engine fundamentally changes the way it interacts with business objects by replacing proprietary parsers with a third-party parser. This simplifies support of DTD 1.0 and the support of XML Schemas when the standard is finalized.

  The Business Object and Script API have been extended with new classes and methods. The new classes and methods let you work with business objects as W3C Documents.

- *Adapter Asynchronous Callback.* An additional Adapter API allows adapters to be more efficient with long-running adapter operations. The Asynchronous Callback method tells the Adapter Server that an operation will be long-running, that system resources should be freed while the adapter waits for a response from the end system, and that another method will be called when the response arrives. The Asynchronous Callback method frees the adapter developer from using the request-retry method that makes the Adapter Server responsible for polling the end system for the response.

- *Script API Changes.* The script API now provides access to the PartnerGroupContext and the Public and Private Process Contexts. Through these contexts, you can get information such as partner group binding, a reference to the process, inputs to the process (which contain a reference to the sender, the ID of the sending node, and the variable name), and unique node and loop IDs.

- *Certificate Support. Partner Agreement Manager* 2.2 is able to request and import certificates from certificate authorities like VeriSign. This lets organizations use their existing certificate, or request a new one if their partners do not accept self-signed certificates. Partner Agreement Manager 1.1 supported only self-signed certificates.

- *Outbound Proxy Support. Partner Agreement Manager* 2.2 channels that use HTTP communication can work with outbound proxies that use authentication. Outbound proxy authentication is used within *internal* networks to ensure that only people and applications that are authenticated may communicate with an *external* network. Authentication in the outbound proxy is done with a standard user name and password combination. You can turn on the outbound proxy feature after installation. Thereafter, all outbound HTTP communication will use the same user name and password combination for the proxy.

**NOTE:** Note that this feature is only used by channels using HTTP communication; it does not apply to channels that use the built-in Partner Agreement Manager proxy.

# 1

# INTRODUCING PARTNER AGREEMENT MANAGER

Welcome to WebSphere Partner Agreement Manager 2.2, the leading B2B software platform. Partner Agreement Manager is designed to create a seamless web of electronic communications and commerce between businesses over the Internet—helping you to work more closely and effectively with your partners, suppliers and customers.

This introductory chapter includes these sections:

- *Partner Agreement Manager and the extended enterprise* on page 2.
- *About Partner Agreement Manager* on page 3.
- *Working with Partner Agreement Manager* on page 4.

# Partner Agreement Manager and the extended enterprise

Keeping pace with today's rapidly changing business environment requires innovative software solutions that enable highly collaborative business relationships. WebSphere Partner Agreement Manager was designed to meet the challenges of companies that rely on partners to perform critical activities. With its seamless coordination, Partner Agreement Manager allows companies that work together as partners—such as suppliers, resellers, manufacturers, distributors, and customers—to function as efficiently as a single organization.
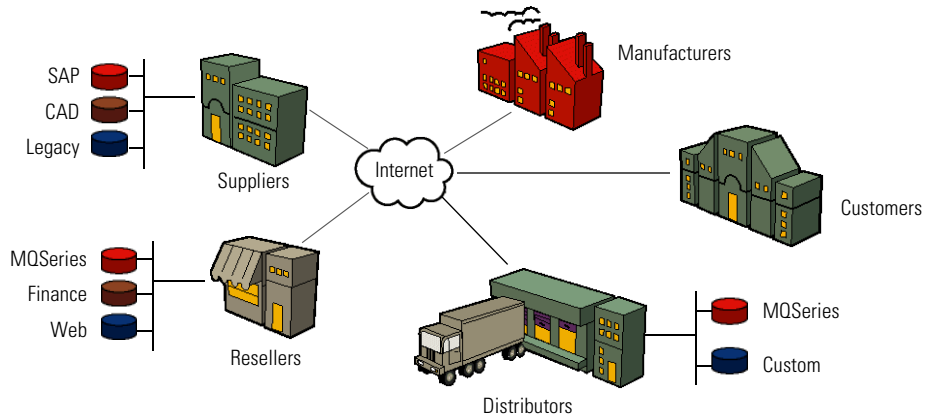


Partner Agreement Manager offers a complete and flexible software platform that is easy to manage and deploy. These are some of the functions that Partner Agreement Manager performs:

- Integration with existing applications.

- Management of communications and security.

- Support for your entire internal and B2B process life cycle, including modeling, execution, and management.

- Administration for partner management, including audit analysis and reporting.

With Partner Agreement Manager, you can continue to interact with existing business systems and business partners around the world. Data exchange is transparent over the Internet or other open networks.

# About Partner Agreement Manager

Partner Agreement Manager lets enterprises communicate with their business partners using fully integrated, real-time connections. This facilitates the flow of business information between partners in a controlled, secure environment. You and your trading partners can work together effectively, yet maintain organizational independence when you need it.
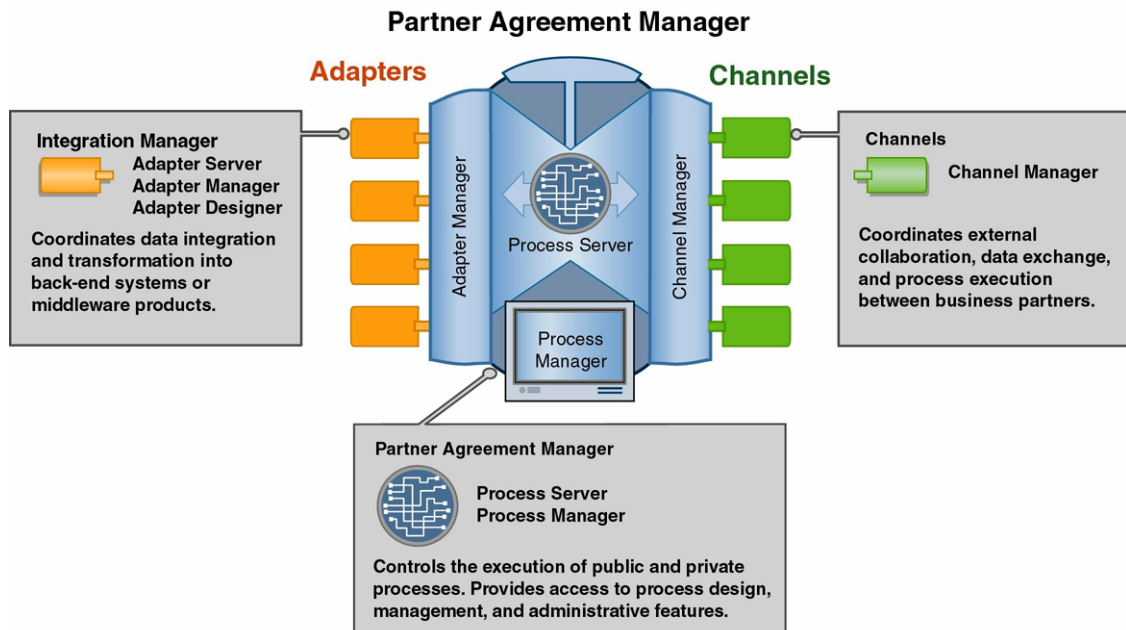


The key components of Partner Agreement Manager are:

- Partner Agreement Manager (core) handles the execution of public and private processes. This is where you design processes and create business objects that handle the transfer of information. Within Partner Agreement Manager:

  - The Process Server is the engine that powers Partner Agreement Manager.

- You use the Process Manager to design public and private processes, design formats for sharing business information, and manage processes. You use Public Process and Private Process windows to build public and private processes. Public processes define the flow of actions and information between partners. Private processes determine the actions that each partner takes for its step in the public process.

  **NOTE:** In earlier releases of Partner Agreement Manager, the Process Manager window was called the Partner Agreement Manager window, or the Explorer window.

- The Integration Manager provides the connection and takes care of data transformation when you need to exchange information with back-end or other systems within your organization. Within Integration Manager:

  - The Adapter Server is the engine that powers integration.
  - You use the Adapter Manager to add, remove, start, and stop adapters.
  - You use the Adapter Designer to create or modify adapter types that provide connectivity to end systems. Depending on the end system, you may also be able to use integration wizards to create adapters.
  - Individual adapters provide for integration with specific back-end systems. Examples are the adapters for MQSeries.

- The Channel Manager handles the configuration and management of the channels Partner Agreement Manager uses to connect you to your partners. Each channel has its own characteristics and capabilities. An example is Channel for RosettaNet.

# WORKING WITH PARTNER AGREEMENT MANAGER

Here's a brief overview of the steps involved in working with Partner Agreement Manager and where to locate information about each step.

**STEP 1** Install the Partner Agreement Manager Process Server and Partner Agreement Manager Process Manager. (For more information, see the *Partner Agreement Manager Installation Guide*.)

**STEP 2** Set up your Partner Agreement Manager profile, your Partner Agreement Manager users, and your partners. (For more information, see the *Partner Agreement Manager Administrator's Guide*.)

**STEP 3** Analyze existing processes and identify the workflow between partners. (See *Designing processes* on page 15.)

**STEP 4** Design and create the business objects you'll use to exchange information. (See *Defining business object types* on page 27.)

**STEP 5** Create maps that populate business object types from another source. (See *Mapping business objects* on page 59.)

**STEP 6** Design the public processes. (See *Defining public processes* on page 93.)

**STEP 7** Distribute the public processes to your partners for approval. (See *Managing processes* on page 203.)

**STEP 8** Design your private processes plus any private business objects you'll use to transfer information internally within a private process. (See *Defining private processes* on page 139.)

As part of a private process, you may need to include one or more adapters. (For more information about adapters in general and the Adapter Development Environment, see the *Partner Agreement Manager Adapter Developer's Guide*. For information about specific adapters, see the guides that accompany any adapters you may have already installed.)

The actions in your private processes will also include scripts. (For more information, see the *Partner Agreement Manager Script Developer's Guide*.)

**STEP 9** Install the process and start running it. (See *Managing processes* on page 203.)

**NOTE:** For information on finding an instance of a process, monitoring a running process, auditing a completed process, and viewing system messages, see the *Partner Agreement Manager Administrator's Guide*.

# 2

# PARTNER AGREEMENT MANAGER BASICS

Read this chapter for information about starting and ending a Partner Agreement Manager session and using the Process Manager window.

This chapter includes these sections:

- *Starting a Process Manager session* on page 8.
- *About the Process Manager window* on page 10.
- *Ending the Process Manager session* on page 13.

# Starting a Process Manager session

When you are working with processes, business objects, and partners, the two key components in Partner Agreement Manager are the Process Server and Process Manager. The Process Server is the engine that powers process execution. You can run the Process Server as either a Window NT service or as an application.

The Process Manager is your access point for process design, process management, process auditing, and a host of other tasks. the Process Manager is a client that runs against the Process Server. Therefore, 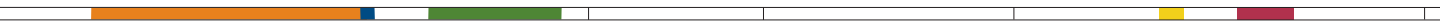the Process Server must be running before you start the Process Manager. In previous releases of Partner Agreement Manager, the Process Manager was called the Partner Agreement Manager window or the Explorer window.

**Partner Agreement Manager**



The Process Server is the engine that powers Partner Agreement Manager.

The Process Manager is a client that runs against the Process Server.
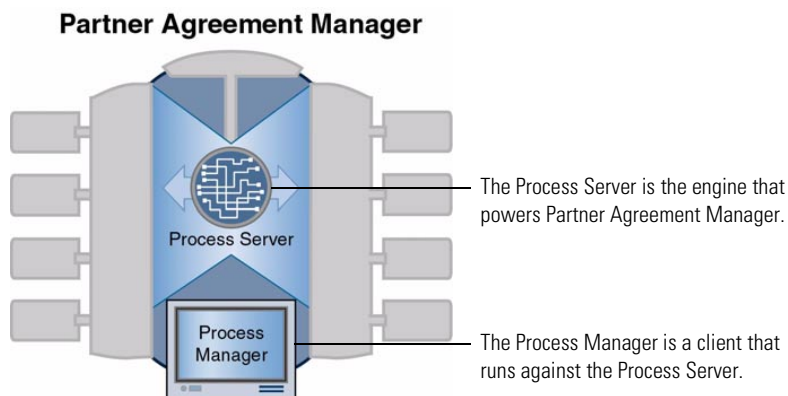
After the Process Server is running, under Windows NT you can start the Process Manager (the Partner Agreement Manager client). The Process Manager runs only on Windows NT and is not supported on UNIX. If you have the Process Server installed on a UNIX system, you must install Process Manager on a Windows NT system. Check with the Partner Agreement Manager administrator to make sure that the Process Server is running before starting the Process Manager.

## Running the Process Manager

You can run the Process Manager the same way you run any other application program. You will need a Partner Agreement Manager user name and a password to log in. Check with your Partner Agreement Manager administrator for your user name and password.

**To run the Process Manager:**

**1** On the Windows desktop, click Start>Programs>IBM WebSphere Business Integrator>Process Manager (*partner name*).

The Process Manager Login dialog box appears.

Type your user name and password.
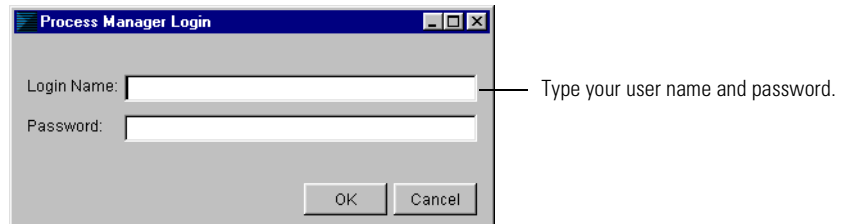
**2** Type your Partner Agreement Manager user name (login name) and password and click OK.

User names are not case-sensitive, but passwords are. The Process Manager window appears. (See *About the Process Manager window* on page 10.)

# About the Process Manager window

When Partner Agreement Manager opens, it displays the Process Manager window. This window is designed to resemble and behave like the Windows NT or Windows 95 Explorer windows. Like other explorer-type windows, the Process Manager window is divided into two panels. The left panel contains a hierarchical list of the Partner Agreement Manager module folders: Administration, Partners, Business Objects, Processes, and Auditor. Each top-level folder contains one or more subfolders that organize its contents. For example, the Administration folder contains subfolders for Certificates, Channels, and Users.

Toolbar buttons give you quick access to important Process Manager window commands.

Top-level folders represent Partner Agreement Manager modules.

Choosing a folder in the left panel displays its contents in the right panel.

The status bar shows the current user and the company name.
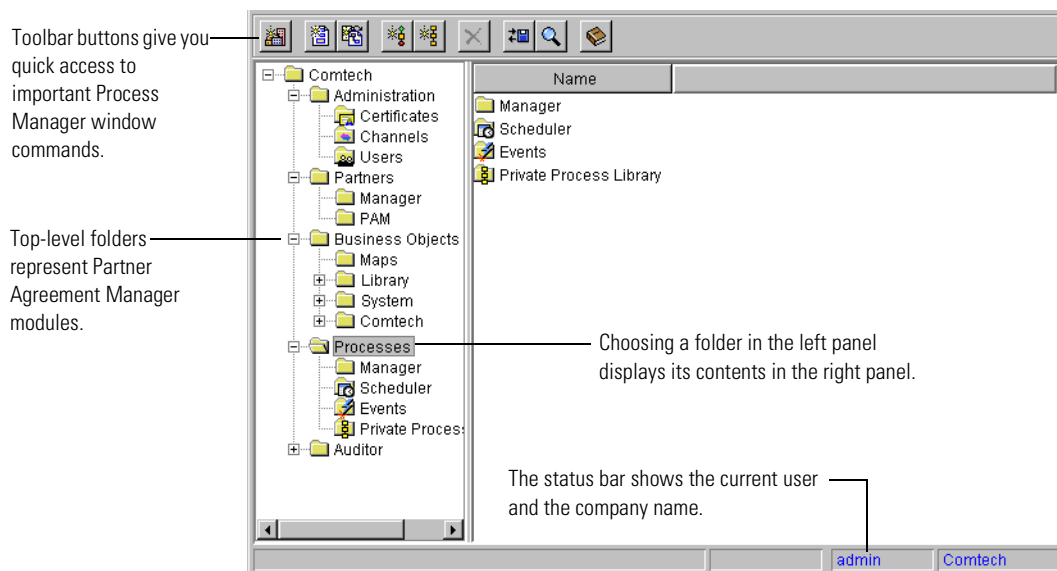
The two panels in the Process Manager window work together to display Partner Agreement Manager modules. For example, when you choose the Administration folder in the left panel, the right panel lists the Certificates folder, the Channels folder, and the Users folder. Similarly, choosing the Users folder in the left panel displays a list of Partner Agreement Manager users in the right panel.

**NOTE:** If you don't have access to a module, its folder appears but doesn't contain subfolders.

You can open the folders in the Process Manager window and resize its panels the same way you do in any explorer window.

| To | Do this |
|---|---|
| Open a folder | Click the plus sign (+) to the left of a folder. |
| Close a folder | Click the minus sign (-) to the left of the folder. |
| Display the contents of a folder | Click the folder in the left panel. |
| Open a folder **and** display its contents | Double-click the folder in either panel. |

## About the Command toolbar

With the Command toolbar at the top of the Process Manager window, you can you set up new partners; create new business objects, public processes, private processes, and external events; delete items in the Process Manager window; import and export public processes; or find an instance of a process.
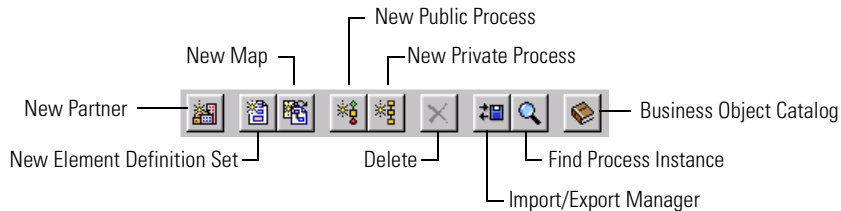


Other Partner Agreement Manager windows (Public Process, Private Process, and Business Objects) also have toolbars that give you access to the most important functions in their respective windows.

## About the status bar

The status bar at the bottom of the Process Manager window displays the names of the current user and company. Other Partner Agreement Manager windows (Public Process, Private Process, and Business Objects) also have status bars that give you information about the item currently displayed in the window.

## Resizing Process Manager window elements

You can resize the panels in the Process Manager window, or change the width of any column in the right panel. The column widths remain the same, regardless of the module you're viewing. Any changes you make remain in effect only for the current Partner Agreement Manager session: Partner Agreement Manager reverts to its preset panel and column widths at the start of each Partner Agreement Manager session.



Drag the right edge of a column heading to resize the column.

Drag the bar between panels to resize the panels.
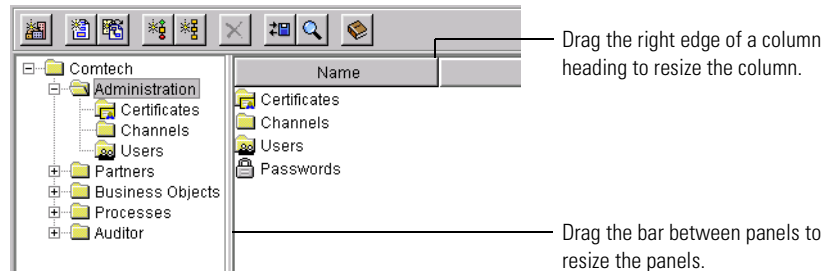
## Sorting Process Manager window contents

You can sort the items in the Process Manager window based on the contents of any column. The order of items in the left panel remains fixed, but you can freely sort the contents of the right panel.

**To sort Process Manager window contents:**

1 Right-click any item in the column that you want to use as the basis for sorting.

For example, if you want to sort by Contact Name, right-click any contact name.

2 Choose Sort by *<column name>* from the menu that appears, and choose Ascending or Descending.

Partner Agreement Manager sorts the items in the right panel based on the column you choose. The columns remain sorted until you change the sorting or open another folder.

**TIP:** You can also sort items by clicking a column header. Partner Agreement Manager initially sorts items in ascending order. If you click the column header again, Partner Agreement Manager sorts in descending order.

## Moving Process Manager window columns

You can rearrange the columns in the right panel of the Process Manager window. You might, for example, want to move columns that you're interested in viewing to the left, and those you're less interested in to the right. You can then resize the Process Manager window and still see the most important columns.

### To move Process Manager window columns:

▶ Drag the column header to the position you want.

Drag the column header to the desired location.
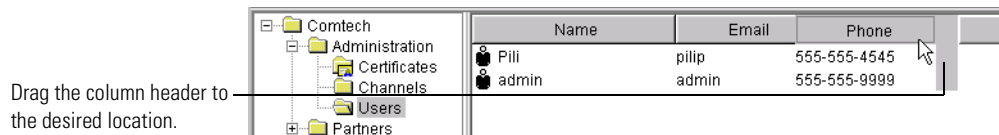


Partner Agreement Manager repositions the column, and shifts all other columns to the right.

Partner Agreement Manager inserts the column and shifts remaining columns to the right.
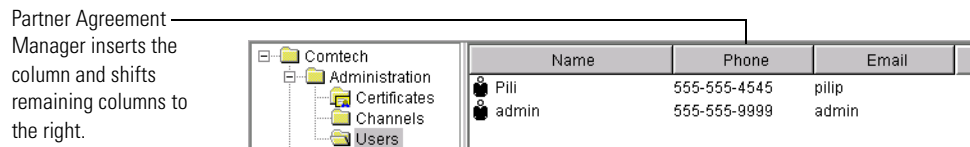


# Ending the Process Manager session

At the end of the Process Manager session, close the Process Manager window. After you have installed processes, you must keep the Process Server running so that the processes can be executed.

# Designing processes

Read this chapter for background information to help you design processes.

This chapter includes these sections:

# About processes

A process is the flow of actions and the exchange of business information between partners in an extended enterprise. The nature and complexity of the processes you develop depend entirely on your business needs. For example, a purchasing process might be as straightforward as the exchange of purchase orders and acknowledgments between two partners.

| | |
|---|---|
| ☐ | Action |
| ⬡ | Information |

**Partner1** — **Fax or Mail** — **Partner2**

**1** Sends purchase order. → *Purchase Order* → **2** Receives, logs, and processes purchase order.

**3** Receives reply. ← *Reply* ← Prepares and sends reply.
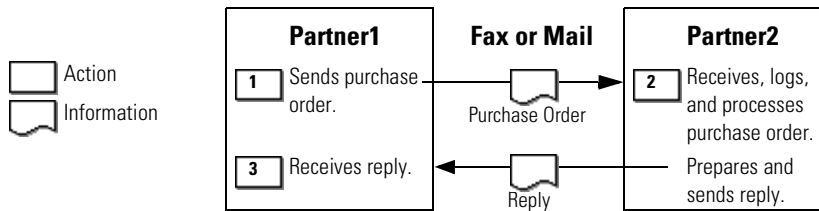
Or it can include the exchange of configuration or specification drawings, requests for clarifications, acknowledgments of information received, requests for additional information from third parties, and much more.

**Partner1** — **Fax or Mail** — **Partner2** — **Fabricator**

**1** Sends design drawings. → *Design Drawings* → **2** Receives, logs, and processes design drawings. If necessary, sends a request for clarification.

**3** Receives request. ← *Request* ←

Prepares and sends clarification drawing. → *Clarification* → **4** Processes clarification. Sends fabrication order.

**Fax or Mail**

*Fabrication Order* → **5** Processes fabrication order. Sends reply.

**7** Receives notice of shipping date. ← *Shipping Notice* ← **6** Receives reply and sends notice of shipping date. ← *Reply* ←

Using the building blocks that Partner Agreement Manager provides, you can model whatever type of processes your extended enterprise requires—from outsourcing logistics operations, to vendor-managed inventory, integrated supply chain management, or contract manufacturing.

# About public and private processes

Partner Agreement Manager's two-level process model is designed to automate transactions and the exchange of information between partners in an extended enterprise. This model promotes cooperation between partners, yet maintains total security and flexibility for each partner's internal activities.

The first level is the *public process*, which defines the step-by-step flow of information and actions between two or more partners in an extended enterprise. One partner develops the public process, and all partners involved review and accept the process before it is implemented. The partner who designs a process is its owner.

The second level is the *private process*, which defines each partner's internal sequence of actions for its steps in the public process. Although all partners in a public process see and agree to its flow, each partner's private process is just that—private. The partner who develops a private process is the only one who can ever see it.



In the public process, Partner1 sends a purchase order to Partner2. Partner2 processes it and replies to Partner1.

Partner2's private process determines exactly how it processes Partner1's purchase order.
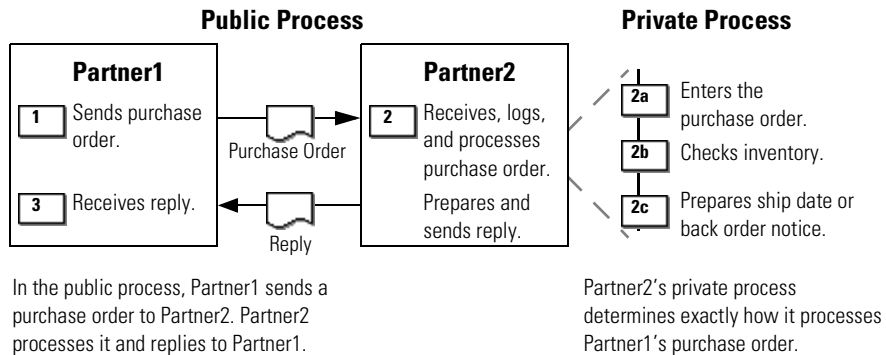
With the private process, you can vary the flow of a process based on whatever criteria you set. It also gives you the flexibility to interact with your enterprise system and with other server-based or desktop applications. You can notify people within your organization via e-mail messages, for example, and get approvals on automatically generated Web-based forms.

# About designing processes

These are the steps involved in designing a Partner Agreement Manager process:

**Step 1** Analyze the existing process and identify the workflow between partners.

This step takes place before you start using Partner Agreement Manager. You begin by looking at how the process currently works. What is its purpose? What steps are involved? What information is exchanged? Where does the information come from? Where does it go? How does it get there? Who are the partners involved? Can the same process work with more than one partner? What are potential problem areas? The answers to these questions will help you decide what actions and information to include in the Partner Agreement Manager process, and in what order. They will also help you design effective error-handling logic that will keep your business processes running smoothly.

**Step 2** Set up your partners (if you haven't already done so).

All companies involved in the extended enterprise must define their Partner Agreement Manager profiles before they can set up partners. (See the configuration information in the *Partner Agreement Manager Administrator's Guide*.)

**Step 3** Define the element definition sets and create the business object types to be exchanged.

Business object types represent the information you want to exchange—forms, documents, and so forth. With the Element Definition Set Editor, you can lay out the structure and define the data fields you need. Fields can include any type of data—standard text or numeric, or binary files, for example. (See *Defining business object types* on page 27.)

**Step 4** Lay out the public process.

Next, you can begin to lay out the public process. The Process window tools make it easy to model the flow of actions and messages between partners. Using a few simple component tools, you can add and remove actions and business object types, and create whatever types of branching and looping paths you need between them. Partner Agreement Manager also gives you the flexibility to design processes that can include any one of a group of interchangeable partners. (See *Defining public processes* on page 93.)

**Step 5**  Distribute the public process to partners for approval.

When you're satisfied with the first draft of a public process, you can distribute it to your partners for their feedback. Partner Agreement Manager's version controls make it easy to track revisions to a process, so you can safely continue to work on a different version of a public process while partners review it. (See *Distributing a process* on page 208.)

**Step 6**  Revise the public process as necessary.

Based on the feedback from your partners, you might need to adjust the flow of the public process—adding or removing steps, adding, removing, or revising business objects, or changing the order of events. When all of the partners involved agree on a public process, it becomes available for implementation. (See *Distributing a process* on page 208.)

**Step 7**  Create private processes and local business objects to implement actions in the public process. Include error-handling logic based on the process analysis you did in Step 1.

While the public process undergoes review and revision, partners can begin creating their private processes. Partners can also create any business objects they need locally to implement their business rules. (See *Defining private processes* on page 139.)

When the public process is approved by everyone, you and your partners can activate your private processes. (See *Activating a different version of a private process* on page 198.)

**Step 8**  Install and activate the process.

After you have verified the public process, and activated private processes for each public process step that you own, you let your partners know that you are ready to test the process. When testing is completed, you and your partners can begin using the process within your extended enterprise. (See *Installing a process for testing* on page 212 and *Activating a process* on page 214.)

# Translating an existing process into Partner Agreement Manager

The first step in designing a process is to identify the existing workflow between partners and translate that flow into a series of Partner Agreement Manager actions. The first example below shows a hypothetical purchasing process that currently relies on faxed documents, phone calls, and manual data entry. To help you see the relationship between an existing workflow and a Partner Agreement Manager process, the second example shows how to implement a more efficient version of the same process in Partner Agreement Manager.

The existing process:

1 When Partner1's manufacturing system signals that it needs communications components, Partner1 prints and faxes a purchase order to Partner2.

2 Partner2 processes the order.

Processing the order includes running it through the configuration analysis software, notifying order administration, and entering the order into the order entry system software.

3 If the configuration is valid, Partner2 faxes a sales order acknowledgment to Partner1. If the configuration has errors, Partner1 and Partner2 exchange faxes and telephone calls until the configuration is correct.

4 When Partner1 receives the sales order acknowledgment, it schedules receipt against the purchase order.

The existing process flow looks like this. Most of the action takes place at Partner2 in response to the faxed purchase order it receives from Partner1.

**Partner1's** manufacturing system generates a purchase order. **Partner1** prints and faxes the purchase order to **Partner2**.

**Partner2** receives the faxed purchase order.

**Partner2** manually enters the purchase order in its configuration analysis system.

Is the purchase order for a valid configuration?

Yes    No

**Partner2's** order administration group phones **Partner1**.

**Partner1** and **Partner2** exchange phone calls and faxes.

**Partner2** manually enters the order into its order entry system.

Yes    No

Is the configuration valid?

**Partner2** prints and faxes a sales order acknowledgment to **Partner1**.

**Partner1** receives the acknowledgment, schedules receipt against the purchase order, and the process ends.

The automated Partner Agreement Manager process:

Now, here's how the same process can be implemented in Partner Agreement Manager. The left side of the illustration (connected with bold lines) shows the public process: the flow of business between Partner1 (who owns the process) and Partner2. The boxed diagram on the right shows the private process that governs the flow of tasks within Partner2.



**Partner1's** manufacturing system triggers Partner Agreement Manager to generate a purchase order and transmit it to **Partner2**.

**Partner2** processes the purchase order.

**Private Process**

The order is entered in **Partner2's** configuration analysis software.

**Partner2's** order administration system is notified of the results.

If the configuration is not valid, a request for clarification is prepared.

If the configuration is valid, the order is entered in **Partner2's** order entry system and a sales order acknowledgment is prepared.

Sales order acknowledgment

Request for clarification

**Partner1** transmits clarification to **Partner2** identifying problem area.

Clarification

New request for clarification

**Partner2** processes the clarification and repeats the request for more information until the configuration is valid.

Sales order acknowledgment

**Partner1** schedules receipt against the purchase order and the process ends.

Dividing the existing process into public and private processes yields tremendous benefits for both companies. By instituting standardized business objects and a clear succession of steps, the two partners can turn what was a labor-intensive process into an efficient and flexible process that can be managed with minimal intervention by busy staff members.

# Process design style guidelines

These style guidelines can help you design consistent, clear processes. You can use these suggestions as a basis for developing your own guidelines.

## Public processes

### Process name and description

- Use nouns instead of verbs. For example, use Order Fulfillment instead of Fulfill Order.
- Use the description field to describe the business objective of the public process. For example, "This process initiates a request for quote from one or more business partners."
- Add user notes in the description field if necessary.

### Partner Node

- Use the label field for the activity description.
- Start the activity description with a verb.
- Start the verb with a capital letter.
- Use the imperative form of the verb. For example, "Prepare joint business plan."
- Use the description field for a more elaborate description of the activity.
- Add user notes in the description field if necessary.

### Branch Label

- For each branch, describe the option that can be chosen.
- Use nouns or verbs.
- Use the imperative form of a verb.
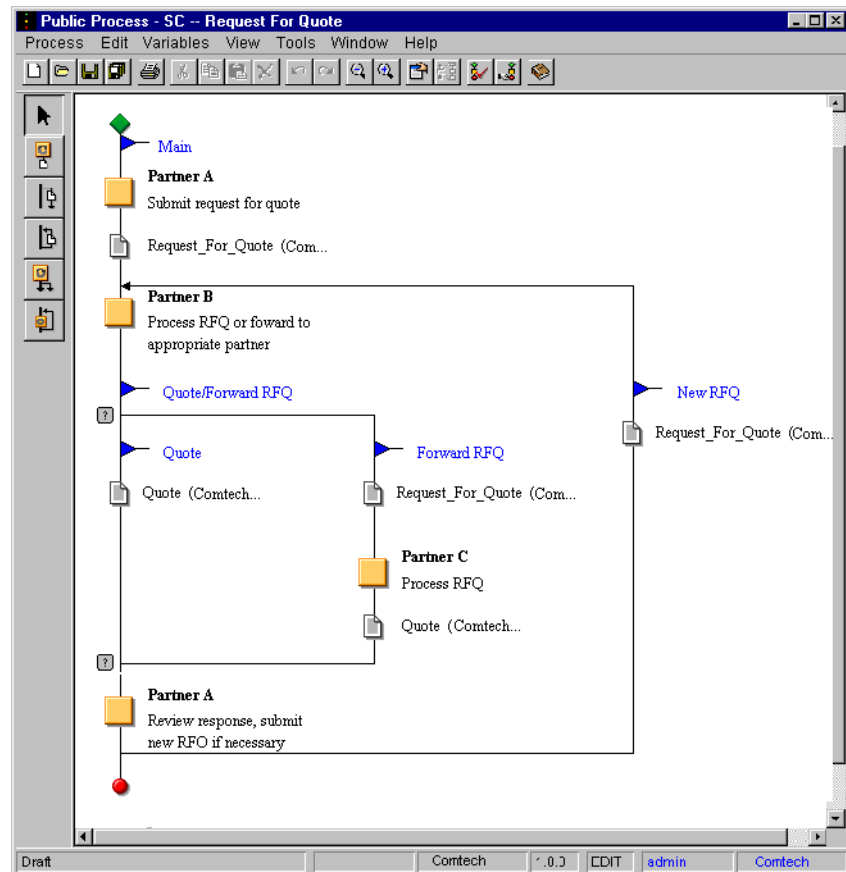- Capitalize every word, and separate words with underscore marks.

For example: Send_Order_Acknowledgment / Send_Change_Order

## Loop Label

- Describe the activity to be executed.
- Start the activity description with a verb.
- Only use the imperative form of the verb.
- Capitalize every word.

For example: Send Revised Order

## Example of a well-designed process

# Element definition sets and business object types

### Element definition set name

- Use nouns.
- Capitalize every word.
- Use underscores between words.
- Use the description field to explain the objective of the Element Definition Set.
- Add notes in the description field if necessary.

Example: Purchase_Order_Acknowledgment

### Top-level element for business object type

- Create one element group that has exactly the same name as the Element Definition Set.
- Choose the element group that has the same name as the Element Definition Set as the top-level element for the Business Object Type.
- Capitalize each word and separate words with underscore marks.

Example: Purchase_Order_Acknowledgment

### Other element groups and fields

- Don't capitalize words.
- Separate words with underscore marks. For example, order_number.
- Use the description field to explain the meaning of the element.
- Add user notes in the description field if necessary.

# 4

# DEFINING BUSINESS OBJECT TYPES

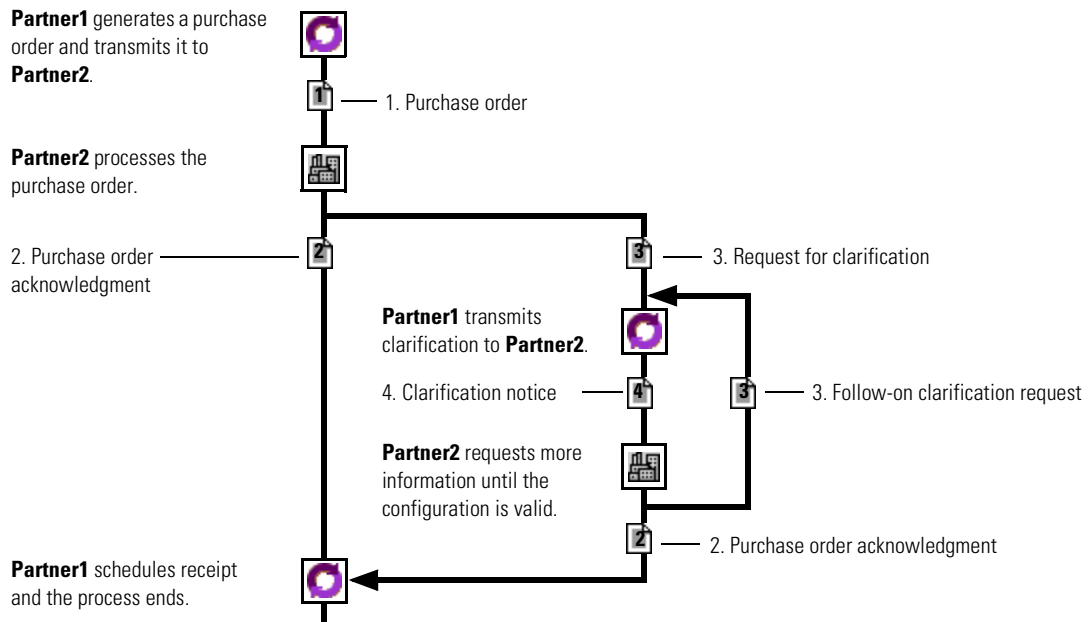Read this chapter for information about creating element definition sets and business object types.

This chapter includes these sections:

# About messages and business object types

When you run a Partner Agreement Manager process, a key component is the *message* that passes from one partner to the next. The message serves two purposes: to transfer information and to pass control of the process to the next partner. Upon receipt of a message, the action in a process passes to the partner who receives it.

In this sample purchasing process, the two partners use messages containing four different kinds of data to transfer information and control within the process: a purchase order, a purchase order acknowledgment (used twice), a request for clarification (also used twice), and a clarification notice.

**Partner1** generates a purchase order and transmits it to **Partner2**.

1. Purchase order

**Partner2** processes the purchase order.

2. Purchase order acknowledgment

3. Request for clarification

**Partner1** transmits clarification to **Partner2**.

4. Clarification notice

3. Follow-on clarification request

**Partner2** requests more information until the configuration is valid.

2. Purchase order acknowledgment

**Partner1** schedules receipt and the process ends.

As they design a process, the partners must agree to the type of information each message will contain and the way in which it will be transmitted. The content of the message is determined by its *business object type*; the way in which it is transmitted (synchronously or asynchronously) is determined by the message's communication service type. You set the message properties—business object type and service type—when you build a public process. (See *Assigning message properties* on page 112.)



The message in a public process is defined by its...

**Message**

**Business object type**

... business object type and...

**Other message properties**

... its message communications properties.
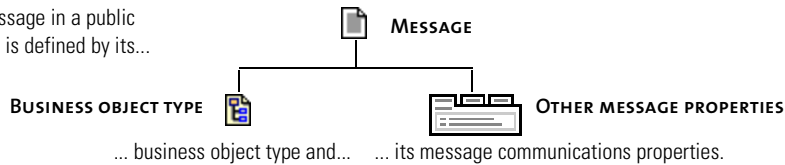
The business object type, in turn, is defined by its top-level element, its key field, and whether it returns audit information. You set these properties when you create the business object type. (See *Creating a business object type* on page 51.)



**Message**

A business object type is defined by a...

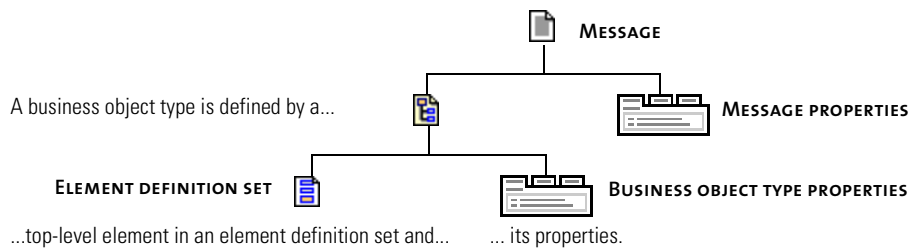**Message properties**

**Element definition set**

...top-level element in an element definition set and...

**Business object type properties**

... its properties.

An element definition set consists of fields used to transfer information. You can arrange fields in groups, and nest groups within other groups, to create a hierarchical structure.



**Message**

**Business object type**

**Message properties**

An element definition set consists of...

**Business object type properties**

...elements that define the structure and meaning of a business object.
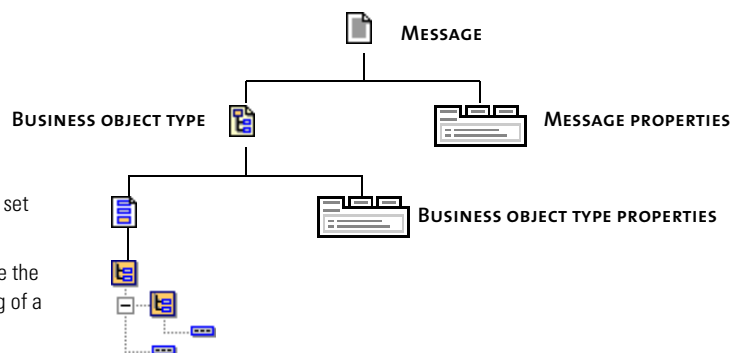
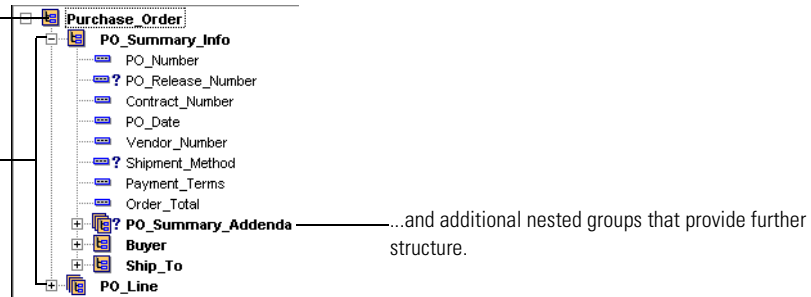Group icon ——— 🔳

Field icon ——— ▭

An element definition set describes the structure and meaning of a business object type. It is essentially a list of elements—fields or groups—that can include any type of data: standard text or numeric data, drawings, or binary files, for example. Depending on the nature of the business object type you want to create from it, you can add any fields you need to the element definition set, or you can add fields and organize them into groups. A purchase order, for example, might contain two groups of fields: one group that identifies the purchase order (vendor number, purchase order number, and so forth) and another group that contains the items to be ordered. Those groups can contain additional groups that provide further structure to the purchase order.

This element definition set contains a top-level group...

...two groups at the next level down...



🔳 Purchase_Order
　🔳 PO_Summary_Info
　　▭ PO_Number
　　▭? PO_Release_Number
　　▭ Contract_Number
　　▭ PO_Date
　　▭ Vendor_Number
　　▭? Shipment_Method
　　▭ Payment_Terms
　　▭ Order_Total
　　🔳? PO_Summary_Addenda
　　🔳 Buyer
　　🔳 Ship_To
　🔳 PO_Line

...and additional nested groups that provide further structure.

In addition to their role in public process messages, business object types can be used with variables and events. For example, Partner Agreement Manager stores the output from a private process in a variable based on the appropriate business object type.

Partner Agreement Manager organizes your business object types and element definition sets in the Business Objects folder in the Process Manager window. The Business Objects folder also contains business object types and element definition sets that your partners have created and distributed to you, a library of element definition sets that you can use as the basis for creating new ones (the Library), and the System folder, which contains business object types and element definition sets that Partner Agreement Manager uses.



This folder contains default Partner Agreement Manager business object types.

This folder contains partner Comtech's business objects.

This folder contains a library of business objects that you can use as a starting point for creating new ones.

This folder contains partner Comtech's element definition sets.
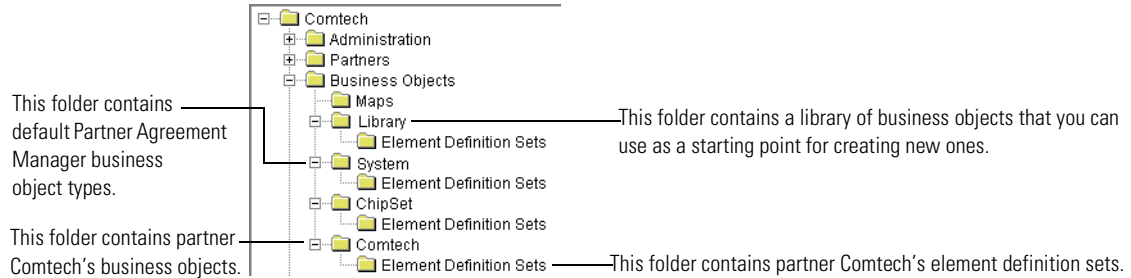
The Business Objects folder also contains a folder that holds any maps you have created. With maps, you can use information from another source to populate a business object type. For more information, see *About mapping business objects* on page 60.
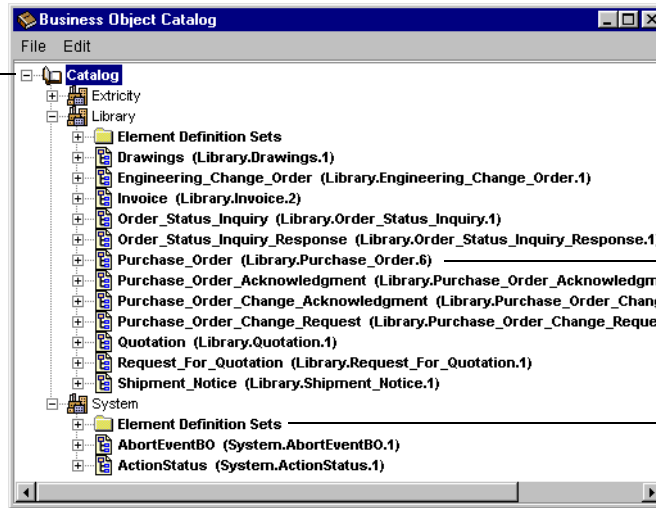
## About the Business Object Catalog

Partner Agreement Manager uses a Business Object Catalog to list all of your business object types and element definition sets. In addition, it contains:

- Library (see *About the Library*, next).
- System business object types and element definition sets.
- Partner Agreement Manager business object types and element definition sets, which are used in the utility adapters.
- Any business object types and element definition sets you receive from partners.

You can keep the Business Object Catalog open as you work so that you can copy and paste business object types or elements from the catalog into an element definition set. You can also use it to view the structure of a business object type used in a process. For DTDs you import, you can see only the root level element in the Business Object Catalog.

The Business Object Catalog contains all of your business object types and element definition sets in your system, including those you created, those created by your partners, and those supplied by Partner Agreement Manager (Library and System business objects).



Business object types can contain field or group elements.

Element definition sets can contain field or group elements.

## About the Library

Partner Agreement Manager provides a Library of generic element definition sets and business object types you might find useful when you create your own element definition sets. For example, one element definition set in the library—Common_Groups (Version: 3)—contains elements you might typically include in many different types of business objects.
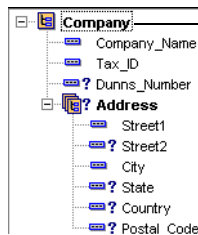


This element definition set contains fields typically used to identify companies.

There are two basic ways to use the business objects in the Library. You can open an element definition set from the library, save it as a new element definition set, and modify it as needed. Or you can copy and paste or drag library elements from the Catalog into your own element definition set.
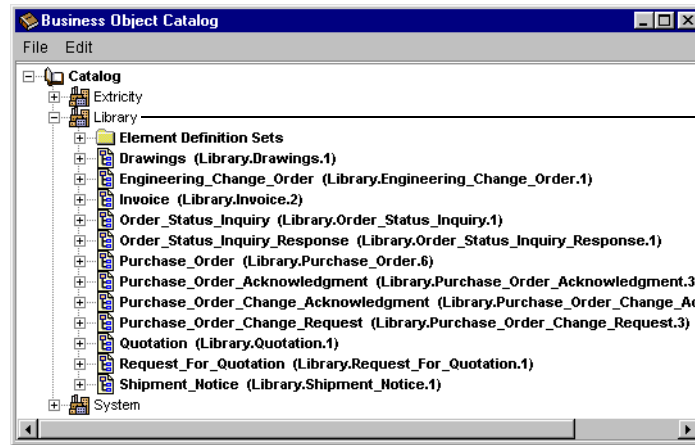


The Library contains a variety of generic business objects.

## ABOUT CREATING ELEMENT DEFINITION SETS

When you create a new element definition set, you name it and add elements. After you save the new element definition set, you can create a business object type from it.

### ABOUT THE ELEMENT DEFINITION SET EDITOR

You use the Element Definition Set Editor to create element definition sets. The Element Definition Set Editor is divided into two panels: the Elements panel and Element Tree panel. The Elements panel shows the elements that you add. Each element is listed individually. You can open a group element to see what elements it contains.

The Element Tree panel shows the hierarchy of elements from any starting point you choose. The lower sections of each panel display the description of an element chosen in the corresponding upper portion.

You can resize the panels by dragging their borders.



Toolbar buttons give you quick access to common Element Definition Set Editor commands.

This panel shows the elements that have been added to the element definition set.

Drag a panel's border to resize the panel.

This panel shows the hierarchy of elements contained in the element you choose.

This area shows the attribute and value for the chosen element.

The status bar shows information about the element definition set.

Partner Agreement Manager is preset to display both panels, with the Element panel on the left and the Element Tree panel on the right. You can choose to display only one panel at a time.

**To change the panel display:**

▶ Open the View menu and choose Elements Only, Element Tree Only, or Elements and Element Tree.

The Element Definition Set Editor uses a variety of icons and graphical cues to provide information about elements in an element definition set.

These are the font cues in the Elements panel.



**Bold** text indicates a group. — Regular text indicates a field element.

*Italics* indicates that an element is included in a group.
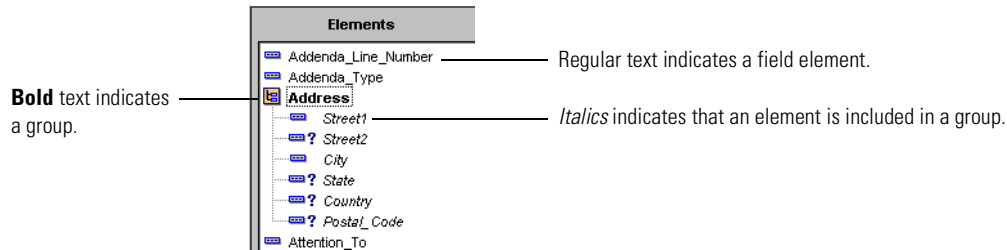
In the Element Tree panel, fields and groups that are members of other groups do not appear in italics.

This table summarizes the icons that can appear in the Element Definition Set Editor.

| This icon | Represents | This icon | Represents |
|---|---|---|---|
| ▭ | A required field | 🖻 | A required group |
| ▭? | An optional field | 🖻? | An optional group |
| ▤ | A required field sequence | 🖼 | A required group sequence |
| ▤? | An optional field sequence | 🖼? | An optional group sequence |

**NOTE:** Unless specifically designated as optional, all elements are required. This means that any element without a question mark is required in its group and must contain data for an instance of the group to be considered valid.

**To change the view in the Element Tree:**

▶ Drag an element from the Element panel into the Element Tree panel, or choose an element from the Element Tree list.



Choose an element to view its contents.
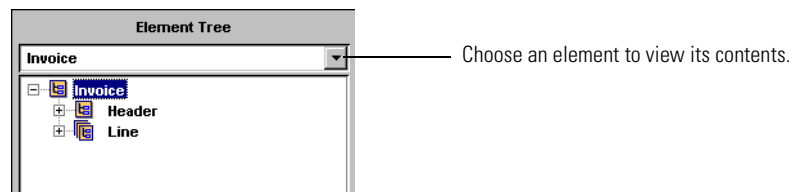
**To view the contents of a group:**

▶ Double-click any group in the Element Tree or a top-level group in the Elements list.

**To view an element's properties:**

- Double-click a field element in either panel.
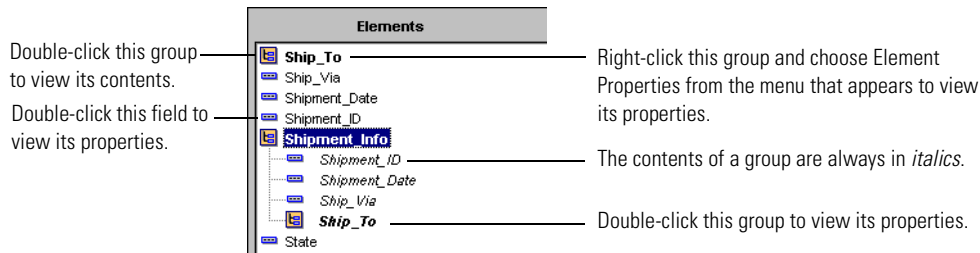- Double-click a nested group element in the Elements panel.
- Select an element and choose Element Properties from the Edit menu.
- Right-click an element and choose Element Properties from the menu that appears.

Double-click this group to view its contents.

Double-click this field to view its properties.

Right-click this group and choose Element Properties from the menu that appears to view its properties.

The contents of a group are always in *italics*.

Double-click this group to view its properties.

In the Element Tree panel, double-clicking the name of a nested group opens or closes it.

Double-click this group to view its contents.
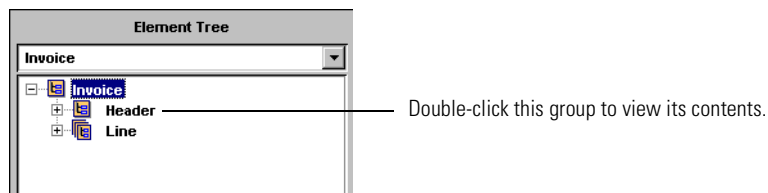
## About the Command toolbar

The Command toolbar at the top of the Element Definition Set Editor gives you access to common File and Edit menu commands, lets you add groups or fields to a business object, and displays the Business Object Catalog.

Insert Element

Insert Child Element

Save Element Definition Set

Element Properties

Delete

New Element Definition Set

View Catalog

Open Element Definition Set

Cut, Copy, Paste

Move Element Up or Down
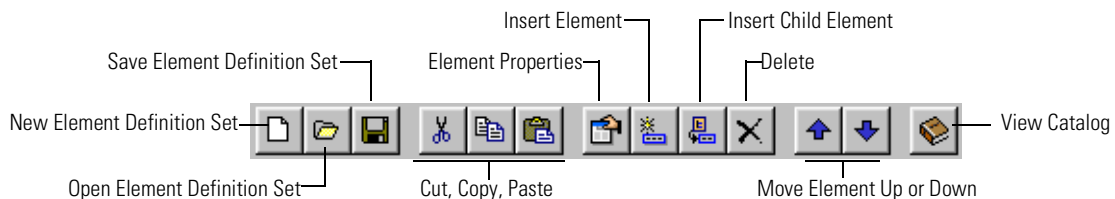
## About the status bar

The status bar at the bottom of the Element Definition Set Editor shows information about the current user and the state of the element definition set.
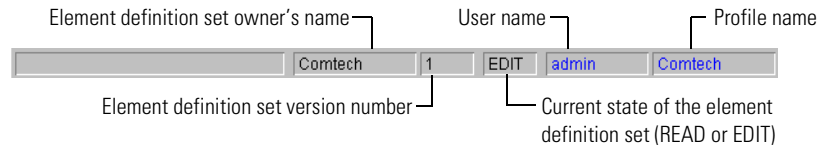


## About the menu bar

The Element Definition Set Editor menu bar appears at the top of the window.

- The File menu contains commands for creating, opening, saving, closing, and freezing element definition sets, as well as for creating a business object type.
- The Edit menu contains commands for cutting, copying, pasting, and deleting elements, for inserting and moving elements, and for setting element properties.
- The View menu contains commands for hiding or showing the panels in the Element Definition Set Editor.

# Creating an element definition set

You can create an element definition set from scratch, or you can base it on an existing element definition set.

When you create an element definition set, think about the structure you need: not only which fields you will need, but also how you want to arrange them. For example, when you create a business object type from the element definition set, you will choose an element to be the top-level element (typically this is a group element that contains all of the other elements nested within it). Partner Agreement Manager assigns the business object type the same name as the top-level element, so you will want to make sure you add an appropriate top-level element and arrange the other fields and groups accordingly.

Partner Agreement Manager lets you add elements in any order and rearrange them as you want. You can change the order of elements in a group using arrows that move them up and down. You can convert an element into a group by dragging another element or group into it. You can turn a group into a field element by deleting its contents. Deleting the contents of a group simply removes them from the group, not from the element definition set.

## CREATING A NEW ELEMENT DEFINITION SET

You start creating an element definition set by opening the Element Definition Set Editor. You can then add, remove, or rearrange elements or groups as necessary.

**NOTE:** To create an element definition set with attributes and constructs other than optional and mandatory groups and fields, import an externally created DTD in the Element Definition Set Editor. From the File menu, choose Import DTD.

**To create a new element definition set:**

1  Click the New Element Definition Set button in the Process Manager window's Command toolbar.

Or you can choose New from the Actions menu and then choose Element Definition Set from the menu that appears.

The Element Definition Set Editor appears.



Drag the borders of a panel to resize it.
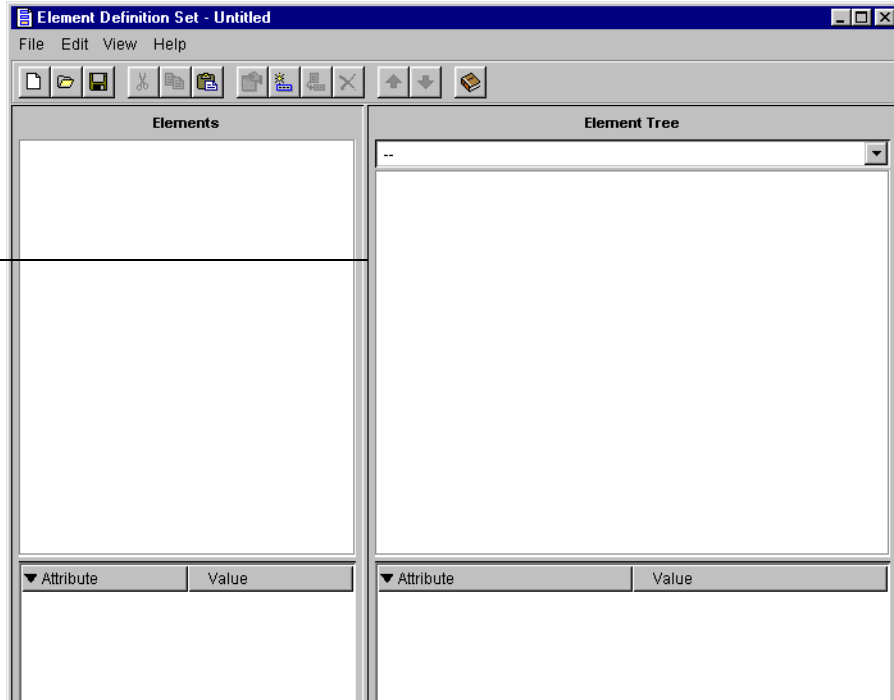
**2** To add an element, click the New Element button in the Element Definition Set Editor's Command toolbar. To add an element to an existing group, or to turn a field element into a group, choose the element or group and click the New Child Element button in the Command toolbar.

You can also choose New Element from the Edit menu. The New Element dialog box appears.



| | |
|---|---|
| | Type a name for the element. |
| | For a field element, choose the content data type. |
| | Content length is generally used for text fields. |

**3** Type a name for the element.

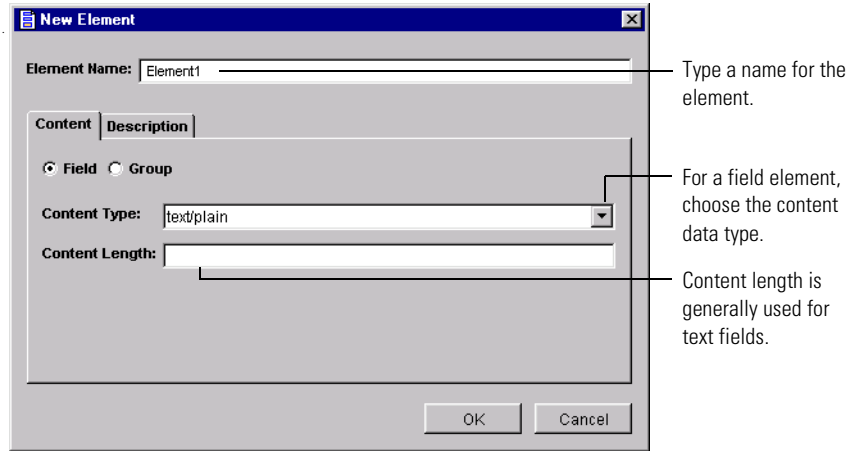**TIP:** Make element names as concise as possible, and use names that are readily understandable by business analysts—for example, Purchase_Order_Type. Use existing business process terms where possible—for example, if a process is called Global Currency Code, name the element Global_Currency_Code.

Often, a data element is described in terms of an attribute such as pounds, inches, or dollars. In these cases, combining the data element with the attribute helps make the element name clear and descriptive—for example, Length_In_Inches. If it is too difficult to encapsulate all the information in a name concisely, use the Description tab to further describe the element.

**4** To add a field, choose the content type.

The content type can be text, images, audio, video, or one of three application types: octet-stream, PostScript, or active. The content length is an optional description that is generally used for text fields.

**5** To add a group, click Group.

The Content tab shows the group properties.



Click to add a field to the group.

**6** Click Add to add a field to the group.

A group must contain at least one element (either a group or a field). You can add a new element, or choose an existing one.



Type a name for the element, or choose an existing element.

Turn this setting on for elements that don't require data.

Turn this setting on for elements that can be repeated.

**7** Specify whether the element is optional and whether it can occur more than once.

An element is only optional within the context of a group. This means that if you use an element in two different groups, you can designate it as optional in one group and required in another.

When you are deciding whether to make an element optional, think about the information the element will contain. A field that contains information that will uniquely identify an instance of a business object type in a process must always be required— for example, the purchase order number on a purchase order. Other fields can be set to optional, such as the second line in an address.

Required elements must be present for a business object to be valid. If the business object can be valid even if the element is not present, make the element optional—for example, the second line in an address or the extension of a telephone number.

**TIP:** When deciding whether to make a field required or optional, consider what impact a null value might have. If null or "" (the empty string, a string of length 0) is a possible value for a field, make the field optional. If null or "" is *not* acceptable, make the field mandatory.

An element is only repeatable within the context of a group. When you are deciding whether to make an element repeatable, consider the information the element will contain. A field that contains information that will uniquely identify an instance of a business object type in a process must occur only once (and must also be required— for example, the purchase order number or the buyer name in a purchase order). Other elements might be repeated, such as line items in a purchase order.

**8** Continue adding elements as necessary.

You can add fields, or you can create groups.

**9** Click the Description tab and add an optional description for the chosen element.

You might use the description to outline the element's intended purpose. Because you can't change the description of a frozen business object, use the description to document only static information.

**TIP:** To make the business object easy to read and work with, document the element with information such as the element definition, expected values (if a code is expected), definition of any codes used, or expected format. For example, date formats can often be confusing—is the format MMDDYYYY or DDMMYYYY? Adding a comment to define the date format can eliminate the confusion.

| Content | Description |

Type a description for the chosen element.
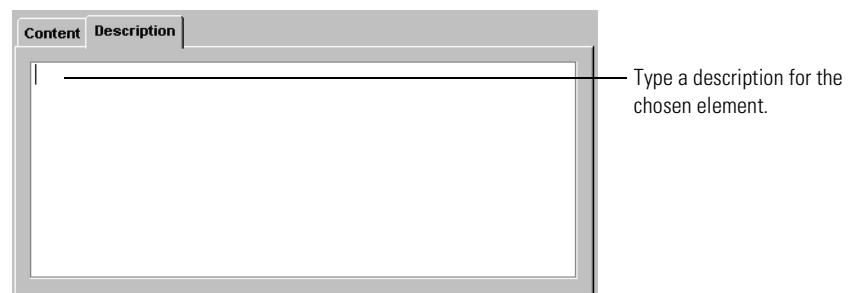
**10** Click OK.

## Adding elements from the catalog

The Business Object Catalog contains all element definition sets and business object types in your system. They are organized by element definition set and by the top-level element of all business object types. You can copy and paste field elements from the Business Object Catalog to an element definition set. You can also copy and paste (or drag and drop) groups of elements, or even entire top-level elements, from a business object type.

**NOTE:** Adding a business object type adds the elements and structure from the corresponding element definition set.
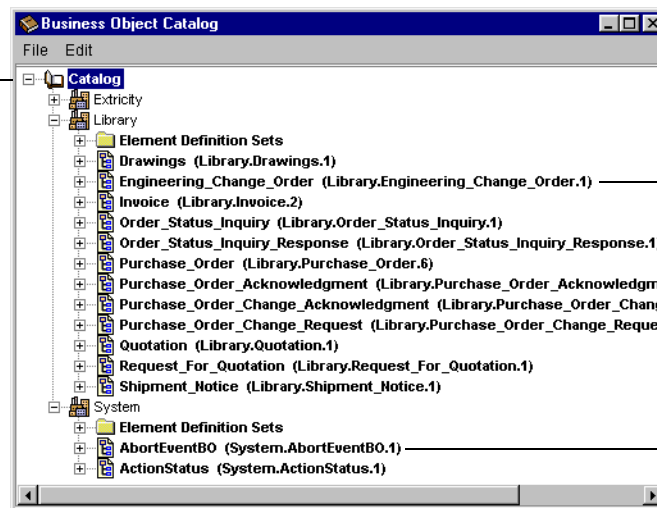
### To add elements from the Business Object Catalog:

1 Click the Business Object Catalog button in the Command toolbar to open the Business Object Catalog.

The Business Object Catalog appears.

The Business Object Catalog contains all business object types and element definition sets in your system.

Choose the top-level element in a business object type.

Choose a field or group element from an element definition set.

2 Navigate to the element you want to add.

3 Drag the element from the Business Object Catalog to the Element Definition Set Editor.
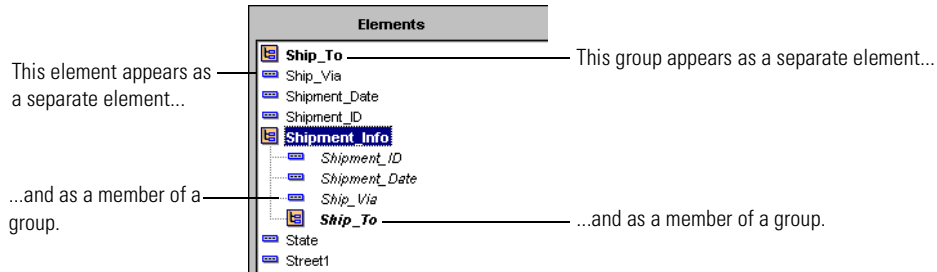
You can also select an element, and then choose Copy from the Business Object Catalog's Edit menu and Paste from the Element Definition Set Editor's Edit menu.

The elements you copy retain their properties and relationships to other elements.

## ADDING CHILD ELEMENTS

A child element is an element that is part of a group. You can choose an existing group and add a child element to it, or you can choose a field element and turn it into a group by adding a child element.

When you add a new child element, Partner Agreement Manager adds it to the element list and displays it as a member of a group, so that the child element appears twice in the Elements list.

This element appears as a separate element...

...and as a member of a group.

This group appears as a separate element...

...and as a member of a group.

**TIP:** You can also convert an element into a group by dragging an existing element onto it.

**To add a child element:**

**1** Choose the element or group and click the Insert a New Child Element button in the Command toolbar.

You can also choose Insert New Element from the Edit menu. The Insert New Element dialog box appears.

**2** Type a name for the element and set its properties.

If the child element is a group, you can add more elements to the child element group at the same time.

**3** Click OK.

## Adding or arranging elements in a group

As you develop an element definition set, you can add existing elements to a group or change the order in which they appear. You can add fields or other groups to a group.

### To add an existing element to a group:

▶ Drag the element to the group.

You can drag a field or a group, and you can drag within the Elements panel or from the Elements panel to the Element Tree.

You can drag a field...

...or a group.

You can drag a field or group within this panel...

...or from this panel to the Element Tree.
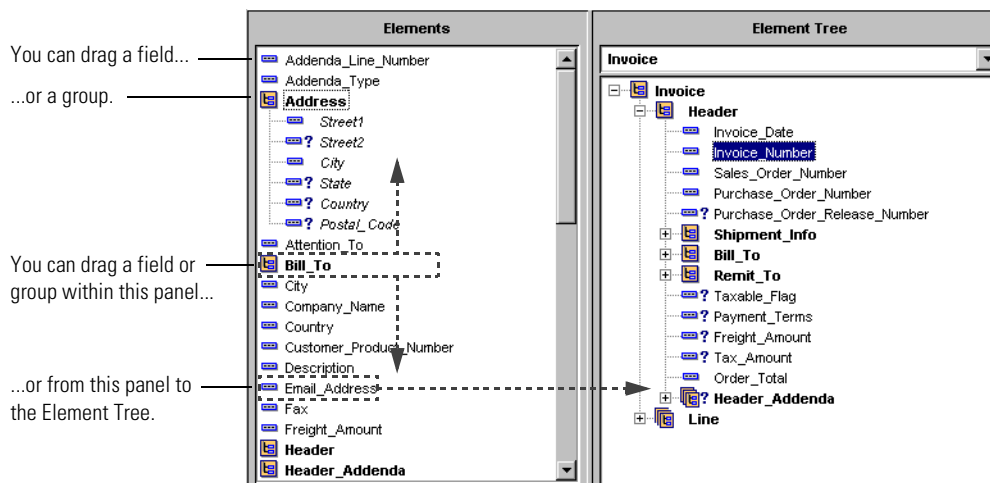
**Tip:** You can also use this method to convert a field element into a group.

### To move an element to a different position within a group:

▶ Choose the element and use the Move Up or Move Down button in the Command toolbar.

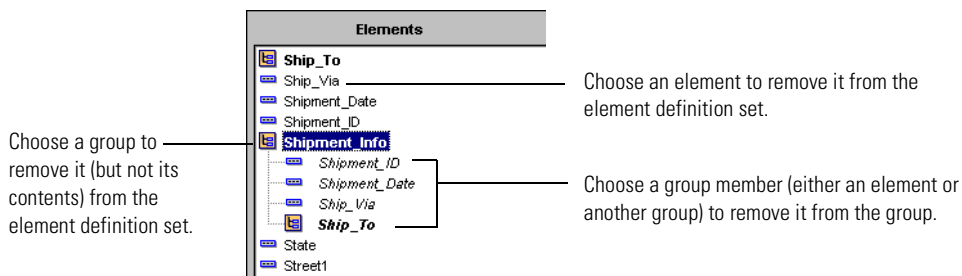You can also choose Move Element Up or Move Element Down from the Edit menu.
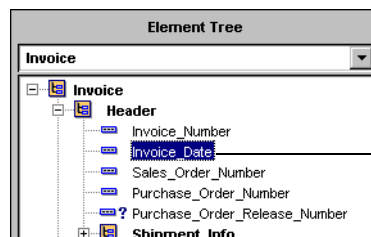
Choose an element and use the Up and Down arrows to move it to another position.

**TIP:** Avoid complex, deeply nested structures when grouping elements within a definition set. Create a group only if there is more than one element that logically belongs within the group, to keep the structure simple and understandable. For example:



This complex, nested element structure...

...can be simplified to make it more understandable.
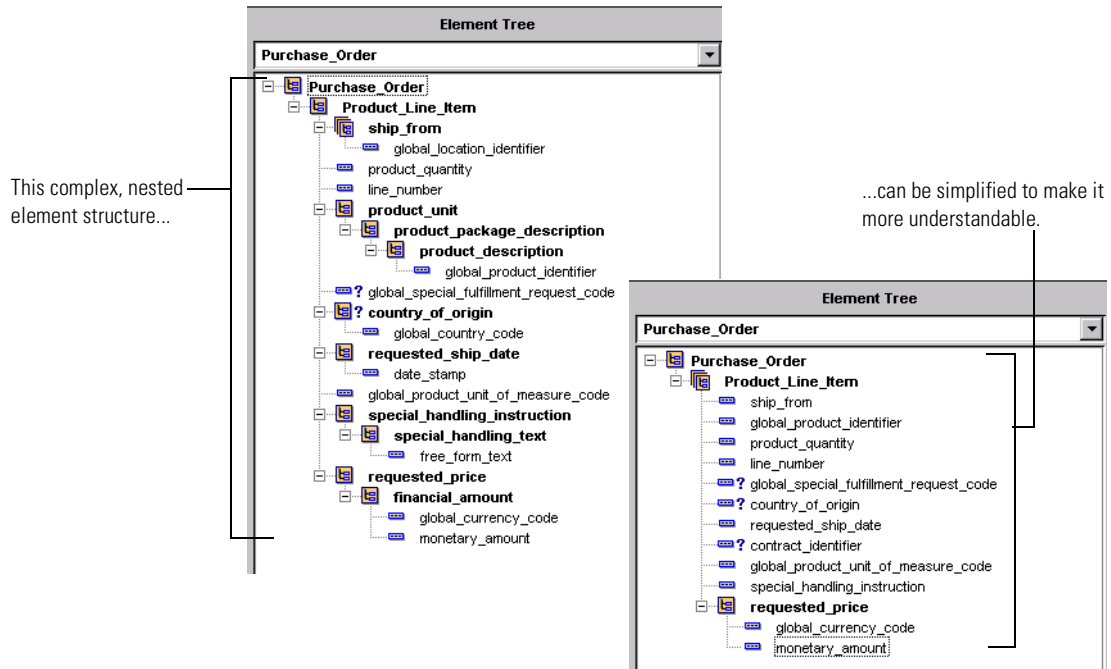
## REMOVING ELEMENTS

Depending on where you choose an element, you can remove it from a group or from the element definition set.



Choose a group to remove it (but not its contents) from the element definition set.

Choose an element to remove it from the element definition set.

Choose a group member (either an element or another group) to remove it from the group.

You can choose items in either the Elements list or the Element Tree.

- Deleting an element from a group removes it from the group, but does not remove it from the element definition set.

- Deleting a group from the Elements list removes just the group, but does not remove its contents from the Elements list.

- Within the Elements list, deleting an item that is not in italics removes it entirely from the element definition set.

**To remove an element from a group:**

▶ Choose the element within the group in either the Elements list or the Element Tree and click the Delete button on the Command toolbar.

You can also right-click the element and choose Delete from the menu that appears, or choose the element and either press the Delete key or choose Delete from the Edit menu. Partner Agreement Manager removes the element from the group, but not from the element definition set.

**To remove an element from the element definition set:**

▶ Choose the top-level element in either the Elements list or the Element Tree and click the Delete button on the Command toolbar.

Partner Agreement Manager removes the element from the element definition set.
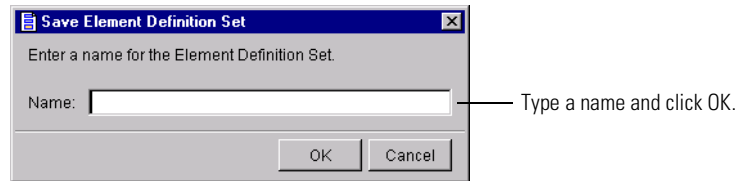
## Saving an element definition set

As you develop an element definition set, it's a good idea to save your work periodically. The first time you save an element definition set you give it a name, and Partner Agreement Manager lists it in your Element Definition Set folder. Thereafter, you can use the Save As command to save the element definition set using a new name. You can also use the Save As New Version command to create a new version of an element definition set.

**Tip:** You can use the Save As New Version command to create an editable version of a frozen element definition set that you own.

**To save an element definition set:**

**1** Choose Save from the File menu.

The Save Element Definition Set dialog box appears.



Type a name and click OK.

**2** Type a name and click OK.

The element definition set appears in your Element Definition Set folder in the Process Manager window.

**NOTE:** The name you type must not contain an exclamation point (!).
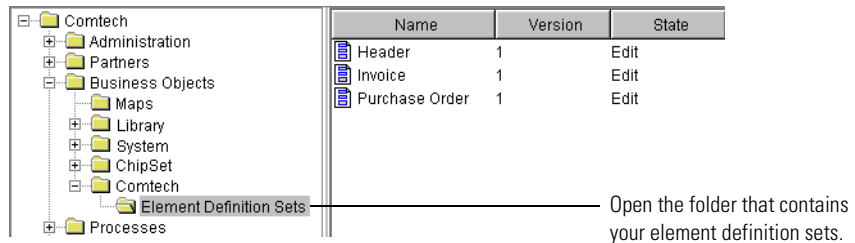
## FREEZING AN ELEMENT DEFINITION SET

As you define element definition sets, you might want to make changes. With Partner Agreement Manager you can revise element definition sets freely before committing to version control.

When you reach a point where you're ready to distribute a public process that includes a business object type based on a particular element definition set, you freeze the element definition set to institute version control. Its state changes from Edit to Read, which means that no further changes can be made to the current version. Although you can refer to a business object type in a process before its element definition set is frozen, you cannot distribute the process for implementation.

From that point on, you must save an element definition set as a new version when you want to make changes. If you must make changes to a frozen element definition set, use the Save As New Version command to create a new version of the element definition set. You can then revise the element definition set, create a new business object type, and replace the previous business object type with the new one in any processes that refer to it.

**To freeze an element definition set:**

**1** Open the Business Objects folder in the Process Manager window, and open the folder that contains your element definition sets.

| Name | Version | State |
|------|---------|-------|
| Header | 1 | Edit |
| Invoice | 1 | Edit |
| Purchase Order | 1 | Edit |

Open the folder that contains your element definition sets.

**2** Right-click the element definition set you want to freeze and choose Freeze from the menu that appears.

You can also open the element definition set in the Element Definition Set Editor and choose Freeze Definition from the File menu.

The state in the Process Manager window changes from Edit to Read.

| Name | Version | State |
|------|---------|-------|
| Header | 1 | Edit |
| Invoice | 1 | Edit |
| Purchase Order | 1 | Read |

Read state means that the element definition set is frozen.

## Creating an element definition set based on another

You can base an element definition set on any existing element definition set, regardless of who created it. For example, you can base an element definition set on one from the Library, on one that was created by a partner, or on one that was installed with Partner Agreement Manager.
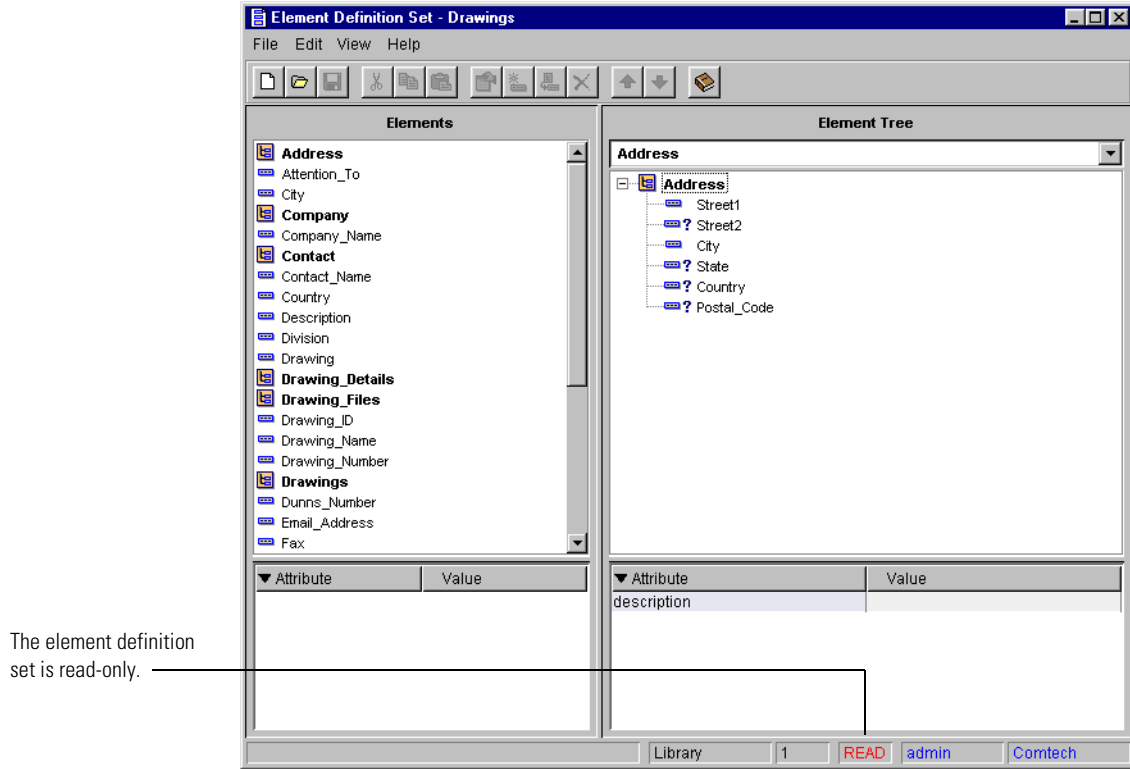
**To create an element definition set based on another:**

**1** Open the Business Objects folder in the Process Manager window, and navigate to one of the Element Definition Sets folders.

The Library, System, and Extricity folders all contain Element Definition Sets folders.

**2** Double-click an element definition set.

The Element Definition Set Editor appears. It displays the elements in the element definition set you chose. Notice that the element definition set is read-only.



The element definition set is read-only.

**3** Choose Save As from the File menu to save a version that you can edit.

The Save Element Definition Set dialog box appears.



Type a name and click OK.
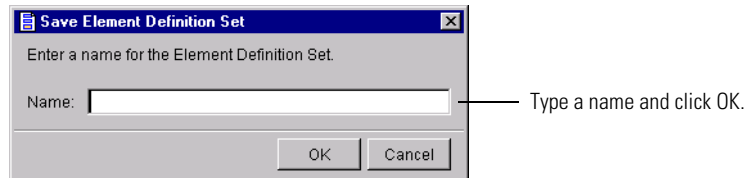
**4** Type a name and click OK.

The element definition set appears in your Business Objects folder in the Process Manager window and you can now edit it.

# CREATING A BUSINESS OBJECT TYPE

A business object type determines the structure and meaning of information used in processes. When you create a business object type, you choose an element definition set, and specify the top-level element for the business object type. You can also specify a key field and whether instances of this business object type must be audited for non-repudiation purposes.

Currently, you can have only one business object type for each element definition set. This feature has been designed so that in future Partner Agreement Manager releases you will be able to create several business object types from a single element definition set.

You can create a business object type either before or after an element definition set is frozen. The name for a business object type is the same as the element you choose as the top-level element.

**To create a business object type:**

1 Open the element definition set from which you want to create the business object type, open the File menu, and choose Create Business Object Type.

You can also right-click the element definition set in the Process Manager window, and choose Create Business Object Type from the menu that appears.
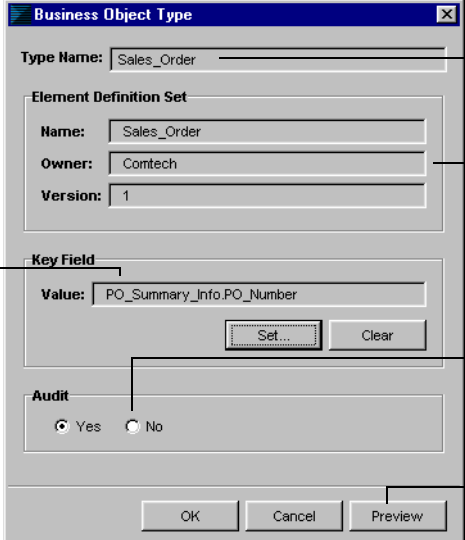
The Create Business Object Type dialog box appears.



Choose the element for the top level of this business object type.
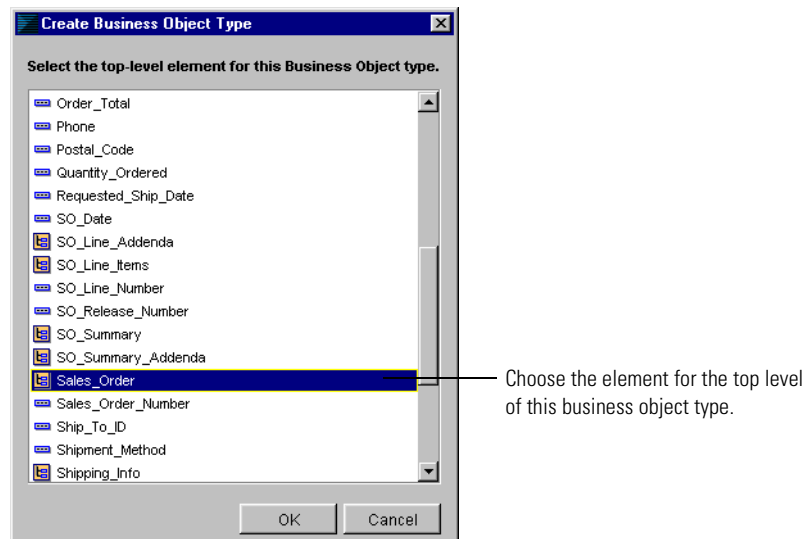
**2** Choose the element that you want as the top level for the business object type.

The name of the top-level element will become the name of the business object type.

**3** Click OK.

The Business Object Type dialog box appears.



The business object type has the same name as the top-level element.

Information about the element definition set appears here.

Choose a field that will have a unique value each time a process is run. The name that Partner Agreement Manager displays includes both the group and field names.

Turn Auditing off if you do not need to be able to extract the contents of messages you send or receive that use this business object type.

Click to see a preview of the structure of the business object type.

**4** To specify a key field, click Set in the Key Field area.

The Set Key Field dialog box appears. The key field can be any field that has a unique value each time you run a process. Partner Agreement Manager can then search on the key field value to locate a specific instance of the business object type. For example, each instance of a purchase order business object type will have a unique PO number.
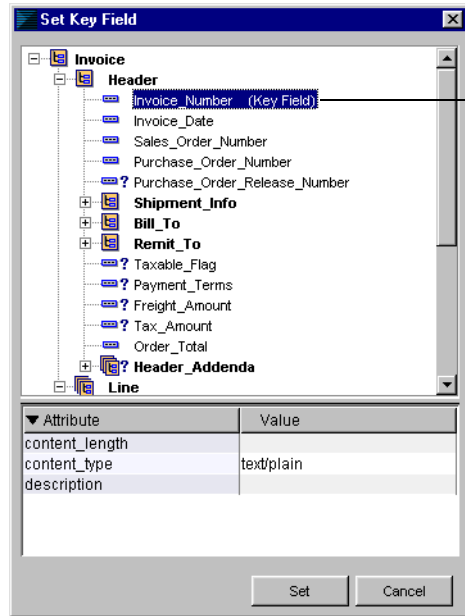


Choose a field to act as the key field for locating instances of a business object type.

**5** Choose a field and click Set.

You return to the Business Object Type dialog box.

**6** If necessary, change the Audit setting from Yes to No.

Partner Agreement Manager is preset to audit all business object types. When auditing is enabled, you can extract the contents of each message you send or receive in instances of a process that uses this business object type.

You might want to disable auditing for certain types of business objects that don't require it. For example, you might want to turn auditing off for business object types that contain large CAD drawings.

---

**IMPORTANT:** If you disable auditing for a business object, non-repudiation support is also disabled for all messages that contain this type of business object.

---

**7** Click OK.

Partner Agreement Manager alerts you to the security implications of disabling auditing.
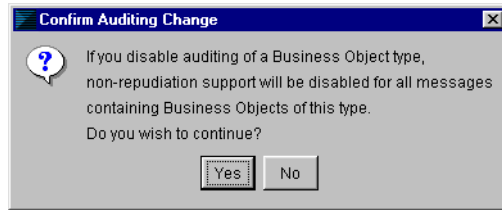


**8** Click Yes to continue. Click No to return to the Business Object Type dialog box and change the setting.

## IMPORTING AND EXPORTING BUSINESS OBJECTS

With Partner Agreement Manager's import and export features, you can import and export both element definition sets and business object types. While you're working in the Element Definition Set Editor, you can export single element definition sets or import individual Document Type Definition (DTD) files. From the Process Manager window, you can start the Import/Export Manager and choose the business object types you want to export, or choose XML files that contain business object types you want to import.

You might, for example, want to build an element definition set or business object type during a Partner Agreement Manager training session and want to move it to your own Partner Agreement Manager installation. You can also use import and export to move an element definition set or business object type back and forth between installations of Partner Agreement Manager running on a desktop computer and laptop so that you can continue working during business trips.

### EXPORTING AN ELEMENT DEFINITION SET

Exporting an element definition set creates a DTD file that contains a complete description of the element definition set.

**To export an element definition set:**

**1** Open the element definition set to export.

**2** In the Element Definition Set Editor, choose Export from the File menu.

**3** Type a name and choose the location where you want to save the export file.

**4** Click Save.

Partner Agreement Manager saves the element definition set to a DTD file using the name and location you specified.

## Importing an element definition set

When you import an element definition set, you can choose any DTD file, regardless of who created it. Partner Agreement Manager examines the DTD to determine if the element definition set has been frozen by another Partner Agreement Manager installation, or if it has the same name and ID as an existing element definition set.

If the element definition set has been frozen by another Partner Agreement Manager installation (by your partner, for example), Partner Agreement Manager lets you import it in either a frozen or unfrozen form.

If the element definition set has the same name and version as an existing one, Partner Agreement Manager permits you to overwrite an existing unfrozen element definition set. If the element definition set is frozen, you cannot import a replacement with the same name and ID.

**To import an element definition set:**

**1** In the Element Definition Set Editor, choose Import from the File menu.

The Select DTD dialog box appears. You can choose any DTD file.

**NOTE:** You must use the File>Import command in the Element Definition Set Editor to import the DTD file.

**2** Choose the file you want to import and click Open.

If the element definition set has been frozen by another Partner Agreement Manager installation, or if it is a duplicate of an existing element definition set, Partner Agreement Manager prompts you to choose what action to take.

**3** The Import dialog box papers.

Choose an Owner for importing the DTD:

- Local—for the local partner
- Partner—for importing a DTD that will be owned by a remote partner
- Standard—for importing a DTD that is owned by a standards body. You must specify the standards body that owns the DTD.

After importing the DTD, you can view its source. You must set an external ID, freeze the element definition set, and create a business object type separately.

**NOTE:** Imported DTDs are not usable in the mapper, the business object catalog, and the instance editor in this release.

## EXPORTING BUSINESS OBJECT TYPES

Exporting business object types creates an XML file that contains a complete description of each business object type you choose and the element definition set it's based on.

**To export business object types:**

**1** Click the Import/Export Manager button in the Command toolbar.

The Import/Export Manager appears.
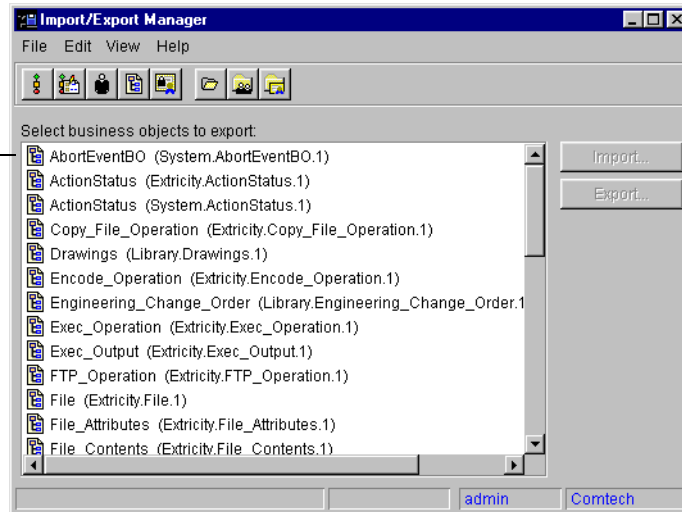
Click to choose a process to be exported.

This area shows a list of the items in a pending import or export action.
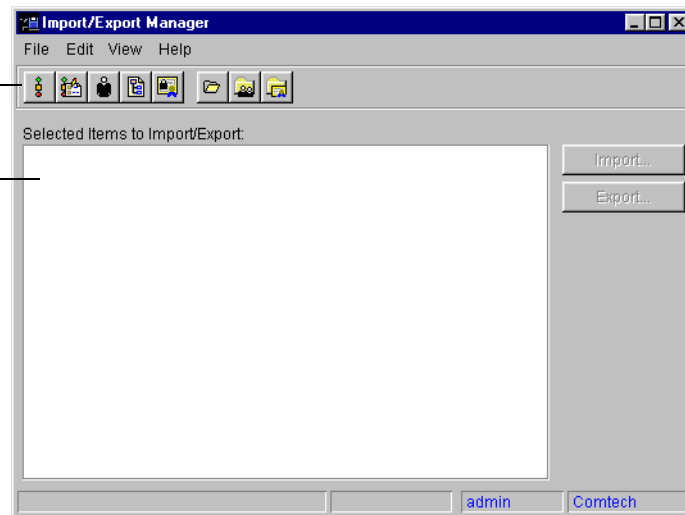
**2** Click the Select Business Objects to Export button in the Command toolbar, or choose Select for Export from the File menu, and then choose Business Object from the menu that appears.

The Import/Export Manager displays a list of all available business objects.

These are all the business objects in your system.

```
Import/Export Manager                              _ □ ×
File  Edit  View  Help

 ⅜ 🏠 👤 📋 🖼️   📂 🖼️ 🗂️

Select business objects to export:
📄 AbortEventBO  (System.AbortEventBO.1)                    ▲    Import...
📄 ActionStatus  (Extricity.ActionStatus.1)
📄 ActionStatus  (System.ActionStatus.1)                         Export...
📄 Copy_File_Operation  (Extricity.Copy_File_Operation.1)
📄 Drawings  (Library.Drawings.1)
📄 Encode_Operation  (Extricity.Encode_Operation.1)
📄 Engineering_Change_Order  (Library.Engineering_Change_Order.1
📄 Exec_Operation  (Extricity.Exec_Operation.1)
📄 Exec_Output  (Extricity.Exec_Output.1)
📄 FTP_Operation  (Extricity.FTP_Operation.1)
📄 File  (Extricity.File.1)
📄 File_Attributes  (Extricity.File_Attributes.1)
📄 File_Contents  (Extricity.File_Contents.1)                 ▼
◄                                                    ►
                                          admin      Comtech
```

**3** Choose one or more business object types, and click Export.

You can also choose Export from the File menu. The Export to File dialog box appears.

**4** Type a name and choose the location where you want to save the export file.

Partner Agreement Manager exports the items you chose and alerts you when the export is complete.

**5** Click Save.

**6** Close the Import/Export Manager.

## IMPORTING BUSINESS OBJECT TYPES

When you import business object types, you can choose any exported XML file, regardless of who in your organization created it. Partner Agreement Manager examines the XML file to determine if the element definition set has been locked by another Partner Agreement Manager installation, or if it has the same name and ID as an existing element definition set.

**To import business object types:**

**1** Click the Import/Export Manager button in the Command toolbar.

The Import/Export Manager appears.

**2** Choose Open for Import from the File menu, then choose XML File.

The Open Exported File dialog box appears. You can choose any export (XML) file.

**3** Choose the file you want to import and click Open.

The Import/Export Manager shows all available components in the file.

This list shows all available components you can import.
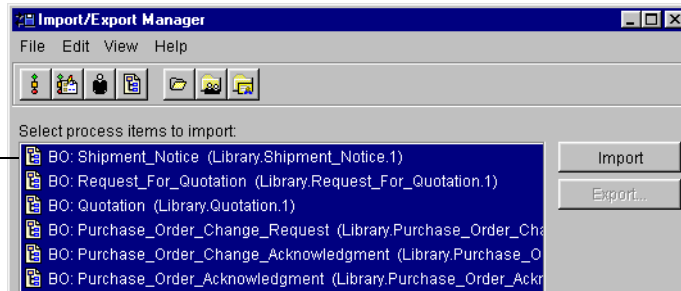


**4** Choose the components you want to import, and choose Import from the File menu or click Import.

Partner Agreement Manager imports the items you chose and alerts you when the import is complete.

**5** Close the Import/Export Manager.

# 5

# MAPPING BUSINESS OBJECTS

Read this chapter for information about creating maps that use information from another source to populate a business object type.
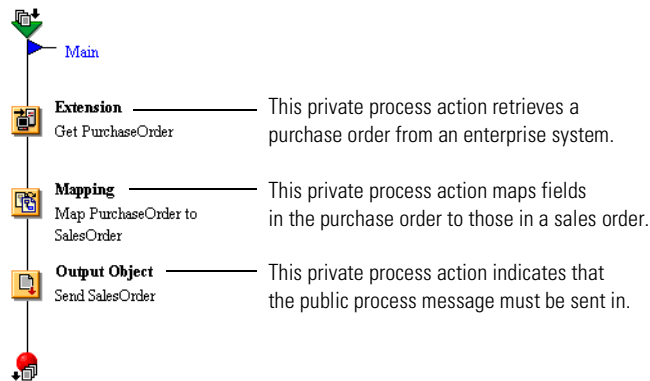
This chapter includes these sections:

# About mapping business objects

As you develop private processes, there might be occasions when you want to populate fields in a business object based on information from another source. That other source might be a value that you enter, or data contained in another business object type. You might, for example, want to set a field in a business object type to always have the same value.
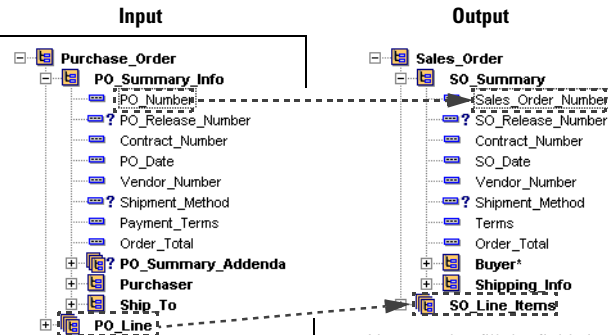
More often, you might want to move information from one type of business object to another business object type. For example, your inventory system generates a purchase order, but your partner needs to receive the information in the form of a sales order. Or you want to populate fields in an invoice based on information from a purchase order.

Within a private process you can transfer data into a business object variable by means of a *mapping action*. The mapping action calls a predefined *map* that establishes the source of information for fields in the output business object. This private process, for example, retrieves data from an enterprise system in the form of a purchase order and outputs it in the form of a sales order.



| | |
|---|---|
| **Extension**<br>Get PurchaseOrder | This private process action retrieves a purchase order from an enterprise system. |
| **Mapping**<br>Map PurchaseOrder to SalesOrder | This private process action maps fields in the purchase order to those in a sales order. |
| **Output Object**<br>Send SalesOrder | This private process action indicates that the public process message must be sent in. |

The map that runs during the mapping action determines which fields in the input business object type (in this case it's a purchase order) supply data to which output fields in the output business object type (a sales order). The two most common mapping actions map fields one-to-one (the data from one input field goes into a single output field), or they map the fields in a repeatable group to those in another repeatable group. A set of repeatable groups or fields is called a *sequence*.

The most common mapping fills a single field with the contents of another single field.



| Input | Output |
|---|---|
| Purchase_Order | Sales_Order |
| PO_Summary_Info | SO_Summary |
| PO_Number | Sales_Order_Number |
| PO_Release_Number | SO_Release_Number |
| Contract_Number | Contract_Number |
| PO_Date | SO_Date |
| Vendor_Number | Vendor_Number |
| Shipment_Method | Shipment_Method |
| Payment_Terms | Terms |
| Order_Total | Order_Total |
| PO_Summary_Addenda | Buyer* |
| Purchaser | Shipping_Info |
| Ship_To | SO_Line_Items |
| PO_Line | |

You can also fill the fields in a sequence with the contents of fields in another sequence.

**NOTE:** You can create a map between any combination of fields and groups. This chapter focuses on the two most common mapping operations.

## ABOUT MAPS

A map is a JavaScript or VBScript that populates fields in the output business object type. It determines which fields in the output business object type are mapped, and identifies the source of information. Partner Agreement Manager provides a Map Editor that automates some or all of the scripting.

**IMPORTANT:** A map might overwrite existing data in an output business object instance. At run time, if an instance of the output business object type already exists, Partner Agreement Manager adds the data that results from the map to that instance. If the instance does not exist, the map calls the createBO function to create it. See the *Partner Agreement Manager Script Developer's Guide* for more information about creating business object instances.

To map externally created DTDs (ones that were imported using the Element Definition Set Editor), you need to write a script.

In the simplest case, the source can be a JavaScript or VBScript expression or a literal that you enter. The expression you enter can include a function call, but it must return a string. If you enter a literal, it must be enclosed in quotes.

---

**IMPORTANT:** JavaScript is supported on both Windows NT and UNIX. VBScript is supported only on Windows NT.

---

**Output**

In this example, the highlighted field is set to the literal string that you type. Notice that the string is enclosed in quotes.



This is the expression that forms the data...

...for this field.

The asterisk (*) indicates that the field is mapped.

If you want to populate an output field with data from a field in an input business object, you can designate that business object type as an input to the map and create an *expression* that maps an input field to a field in the output business object type. You can type the expression, or drag and drop the input field on the output field to have Partner Agreement Manager create the expression for you.

**Input**  **Output**

In this example, the highlighted field is dragged to an output field to create a map expression.



Partner Agreement Manager creates this expression...

...when you drag an input field onto an output field.

The asterisk (*) indicates that the field is mapped.

If you want to populate the fields in a repeatable element group (a sequence) with data from a sequence in an input business object type, you can drag the input sequence to the output sequence. After you create a submap and add the appropriate code, an instance in the output sequence will be created for each corresponding instance in the input sequence. For example, if the input sequence has a sequence of four line items, the output business object type will also have a sequence of four line items.

Dragging an input sequence onto an output sequence creates a *map method* that will be used to fill the sequence. A map method is a reusable logical block of code that populates a particular type of element or element sequence.

In this example, the highlighted sequence element is dragged to an output sequence to create a map method.



Partner Agreement Manager creates this map method...

...when you drag an input sequence onto an output sequence.

The Map Script Editor displays the method Partner Agreement Manager creates.

Within a map method, you can write the expressions that map individual input and output fields in the sequence. Or you can create a submap and drag input fields to output fields and have Partner Agreement Manager create the appropriate mapping expressions. A submap is a secondary level map that is called by a map method to populate an output element other than the top-level element.

In this example, the highlighted field is dragged to an output field to create a submap expression.



Partner Agreement Manager creates this submap expression...

...when you drag an input field onto an output field.

**TIP:** When you drag a single input group onto a single output group, Partner Agreement Manager creates an empty map method and you must manually write the code that fills the output group. It's easier to drag and drop each of the fields in a single group (even if there are a lot of them) and let Partner Agreement Manager generate the necessary expressions.
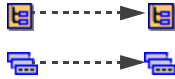


Other elements you can map include non-repeatable group elements and repeatable field elements (*field sequences*). In both of these cases, you must use a map method, rather than an expression, to fill the output element. Partner Agreement Manager creates the map method when you drag the input element to the output element, but you must supply the code—using the Partner Agreement Manager business object API—for populating fields in a non-repeating group or a field sequence.



You can also map more than one input field to an output field. You might, for example, want to populate a purchase order number field with a concatenation of the values in the date and vendor ID fields. In this case, you would drag the two input fields to the output field and insert an ampersand (&) operator into the mapping expression that results.



The ampersand operator (&) concatenates the two field references.

### Code generated by the Map Editor

The Map Editor generates four main types of entries in a business object:

- Non-repeatable field—a single data element.
- Repeatable field (also called a field sequence)—zero or more data elements.
- Non-repeatable group—a set of data elements (fields or groups).
- Repeatable group—zero or more groups.

This matrix shows the code that the Map Editor generates when a data element in an input business object (called inbo) is dragged to the output business object. The left column of the matrix is the input business object and the top row is the output business object:

| | Non-repeatable field (ONRF) | Repeatable field (ORF) | Non-repeatable group (ONRG) | Repeatable group (ORG) |
|---|---|---|---|---|
| Non-repeatable field (INRF) | inbo.getElement ("INRF") | Empty map method fill_seq_ORF with INRF as input | Empty map method fill_ONRG with INRF as input | Empty map method fill_seq_ORG with INRF as input |
| Repeatable field (IRF) | inbo.getElement Sequence("IRF") | Empty map method fill_seq_ORF with IRF as input | Empty map method fill_ONRG with IRF as input | Empty map method, fill_seq_ORG with IRF as input |
| Non-repeatable group (INRG) | inbo.getElement ("INRG") | Empty map method fill_seq_ORF with INRG as input | Empty map method fill_ONRG with INRG as input | Empty map method fill_seq_ORG with INRG as input |
| Repeatable group (IRG) | inbo.getElement Sequence("IRG") | Empty map method fill_seq_ORF with IRG as input | Empty map method fill_ONRG with IRG as input | Empty map method fill_seq_ORG with IRG as input. Option to have a submap (to map from one element of IRG to an element of ORG) auto-generated. |

## About the Map Script Editor

You can use the Map Script Editor to create variables and functions that can be accessed from anywhere in a map, and to add code that is executed at the very beginning and end of each mapping action. You can also use the Map Script Editor to write the code in map methods.

When you first create a map, the Map Script Editor contains four pages: one each for Global Variables, the Library, the **pre** method, and the **post** method. As you add map methods, additional pages appear in the script editor for each map method you add.



Choose a page in this list to view its contents.
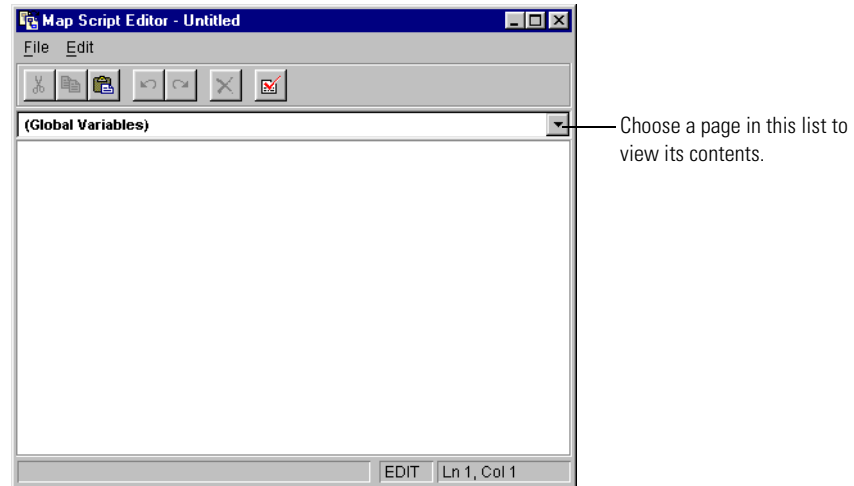
The Command toolbar at the top of the Map Script Editor gives you access to common Edit menu commands, lets you undo or redo a sequence of actions, deletes a method you might have added, and verifies JavaScript or VBScript syntax.



Undo Last Action — Redo Last Undone Action

Verify Script

Cut, Copy, Paste — Delete Method
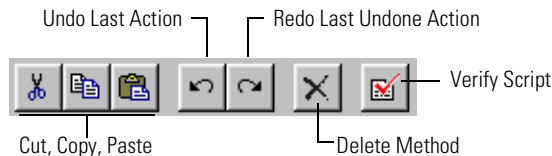
With the commands in the Map Script Editor's Edit menu, you can undo a sequence of steps and redo them, cut, copy, or paste text, delete map methods other than the two preset methods (**pre** and **post**) verify JavaScript or VBScript syntax, choose all text in a page, or go to a specific line number.

## About the Global Variables page

Global Variables are variables that you can access from anywhere in the map. You might, for example, use a global variable to set the value of a field or as the basis for conditional processing. Examples of global variables might be the current date or reorder points for different part numbers.

To define a global variable, click in the Global Variables page and type the appropriate JavaScript or VBScript syntax. Add code to the **pre** method to set the value for the global variable.

**Note:** Global variables must only be set in code executed in the **pre** method.

### About the Library

The Library is where you define functions or subroutines that can be called from anywhere in the map. You might use a library function to format a string or to perform a common calculation.

To define a library function, click in the Library list and type the appropriate JavaScript or VBScript function.

### About the pre method

The **pre** method is the first code that is executed when a mapping action runs. The parameters to the **pre** method are the map inputs. In the **pre** method you can access the map's inputs and set global variables based on their content.

### About the post method

The **post** method is the last code that is executed when a mapping action runs. Its only parameter is the output business object. This method lets you perform any post-processing required on the business object, and is the only time when you can safely modify fields in the output BO based on other fields in the output BO.

**Important:** Partner Agreement Manager doesn't define the order in which mapping expressions and map methods other than **pre** and **post** are executed. All mapping expressions and map method code must be based only on input objects and global variable values.

You might use the **post** method, for example, to set the value of a summary field based on the number of line items in the output business object. Or you might use the **post** method to examine a range of dates in a repeated group and extract the most recent date to be posted in a header field.

Partner Agreement Manager creates the header and footer for both the **pre** method and the **post** method. You can use the Map Script Editor to add JavaScript or VBScript code to the body of the method depending on the tasks you need to accomplish.

**To edit the pre or post method:**

1 Open the Map Script Editor and choose the appropriate page in the list.

2 Click in the **pre** or **post** page between the header and footer, and add code as needed.
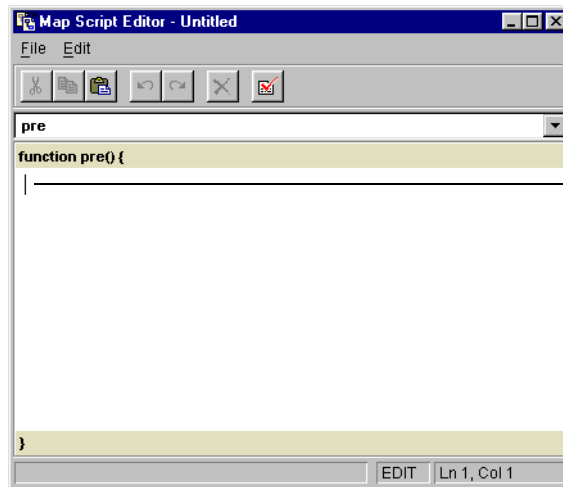


Click in the area between the header and the footer to add code.

3 Click the Verify button in the Command toolbar to check the JavaScript or VBScript syntax.

# ABOUT CREATING A MAP

When you create a map, you start by choosing the output business object type. Then, you add input business object types as needed, and create mapping expressions, map methods, and submaps as needed.

---

**IMPORTANT:** Partner Agreement Manager doesn't define the order in which mapping expressions and map methods other than **pre** and **post** are executed. All mapping expressions and map method code must be based only on input objects and global variable values.

---

The Map Editor is divided into two panels: an inputs panel and an outputs panel. The Inputs panel shows the business object types that have been chosen as inputs to the map.

The outputs panel shows the business object type that will be populated as a result of the map. The lower portions of each panel display the description of an element chosen in the corresponding upper portion.

You can resize the panels by dragging their borders.

Toolbar buttons give you quick access to common Map Editor commands.

This panel shows the map's inputs.

Drag a panel's border to resize the panel.

This panel shows the hierarchy of elements in the output business object type.

This area shows the attribute and value of the chosen element.

The status bar shows information about the map.



Group icon

Field icon

The Map Editor uses the same icons and graphical cues that appear in the Element Definition Set Editor.

These are the font cues in the input and output panels.



**Bold** text indicates a group.

*Italic* text indicates elements in a repeatable group.
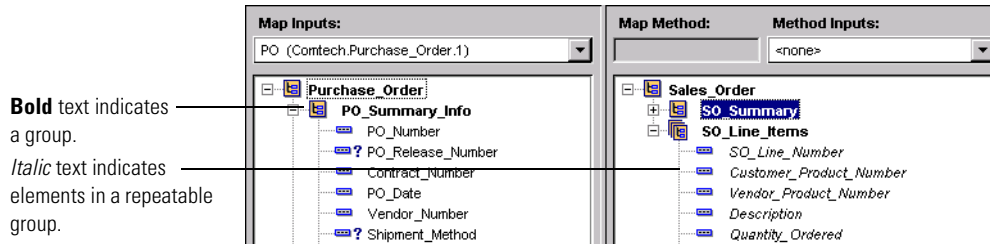
**NOTE:** Partner Agreement Manager doesn't support map expressions for fields in a repeatable group (a group sequence) at this map level. If you want to create expressions for these fields, you must first create a submap for the group. You can then create map expressions for these fields in the submap.
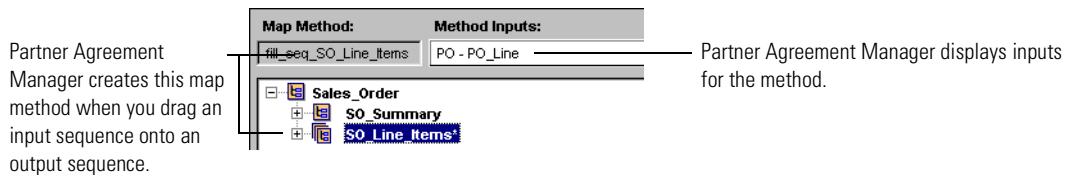
This table summarizes the icons that can appear in the Map Editor.

| This icon | Represents | This icon | Represents |
|---|---|---|---|
| | A required field | | A required group |
| ? | An optional field | ? | An optional group |
| | A required field sequence | | A required group sequence |
| ? | An optional field sequence | ? | An optional group sequence |

**NOTE:** Unless they are specifically designated as optional (a question mark appears to the right of the icon), all elements are required. This means that any element without a question mark is required in its parent group.

The content of the top of the output panel depends on the type of element that you choose in the Output panel. When you choose a single input field, Partner Agreement Manager displays a text field where you can view and edit map expressions.
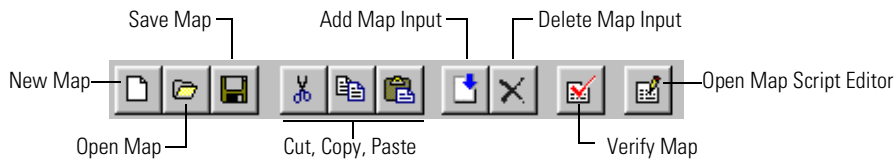


Partner Agreement Manager creates this expression...

...when you drag an input field onto an output field.
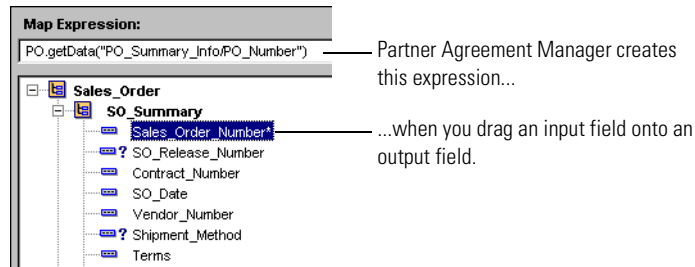
When you choose a group element or field sequence, Partner Agreement Manager displays the corresponding map method and its inputs.

Partner Agreement Manager creates this map method when you drag an input sequence onto an output sequence.



Partner Agreement Manager displays inputs for the method.

### About the Command toolbar

The Command toolbar at the top of the Map Editor gives you access to common File and Edit menu commands, lets you add or remove inputs in a map, verifies JavaScript or VBScript syntax, and opens the Map Script Editor.



Save Map

Add Map Input

Delete Map Input

New Map

Open Map Script Editor

Open Map

Cut, Copy, Paste

Verify Map

### About the status bar

The status bar at the bottom of the Map Editor shows information about the current user and the state of the map.



Map owner's name

User name

Profile name

Map version number

Current state of the map
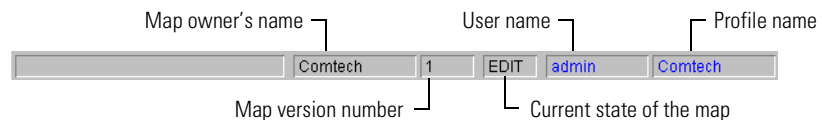
### About the menu bar

The Map Editor menu bar appears at the top of the window.

- The File menu contains commands for creating, opening, saving, closing, verifying, freezing, and testing maps.
- The Edit menu contains commands for cutting, copying, and pasting, for adding and deleting map inputs, for creating a map method call, and for generating the code for a submap call.
- The View menu contains commands for displaying the Script Editor, for viewing the top-level map, and for displaying a submap.

# CREATING MAPS

The first step in creating a map is to choose the output business object type. You can then add input business object types as needed and construct mapping expressions, map methods, and submaps as needed.

---

**IMPORTANT:** Partner Agreement Manager doesn't define the order in which mapping expressions and map methods other than **pre** and **post** are executed. All mapping expressions and map method code must be based only on input objects and global variable values.

---

If you want to use constant values for fields, you can choose the fields and type string values. If you plan to populate the output business object type with data from another business object type, you must add one or more input business object types. You can then drag and drop fields from the input business object types onto the fields in the output business object type.
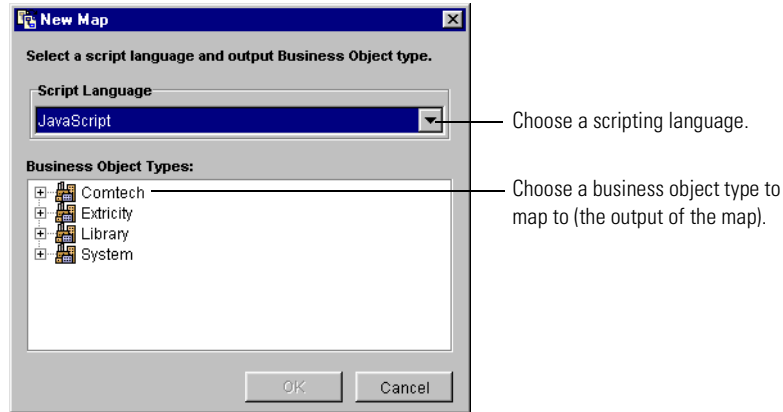
**To create a map:**

1  Click the New Map button in the Process Manager window's Command toolbar.

Or you can choose New from the Actions menu and then choose Map from the menu that appears.
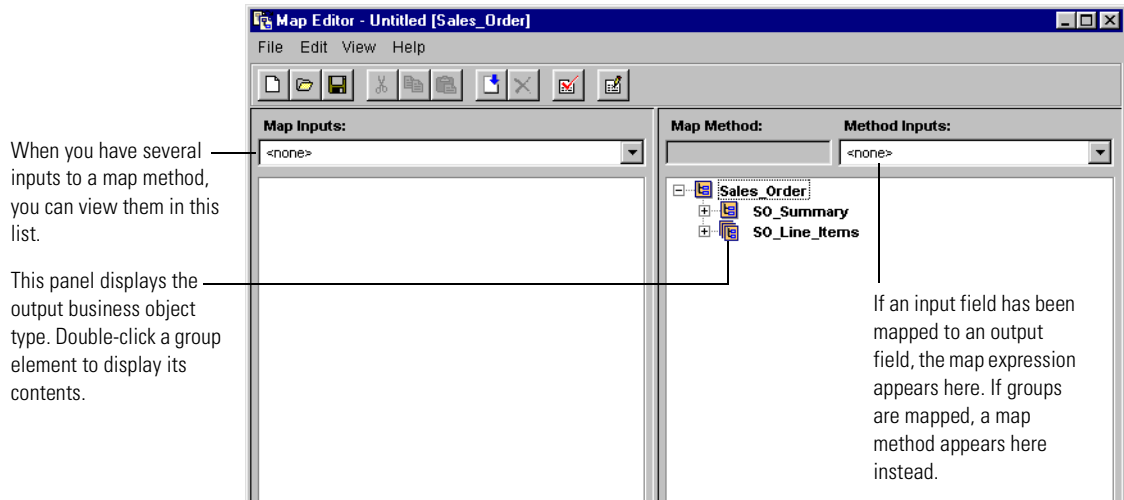
The New Map dialog box appears.

Choose a scripting language.

Choose a business object type to map to (the output of the map).

**2** Choose the scripting language you intend to use.

Note that JavaScript is supported on both UNIX and Windows NT. VBScript is supported only on Windows NT.
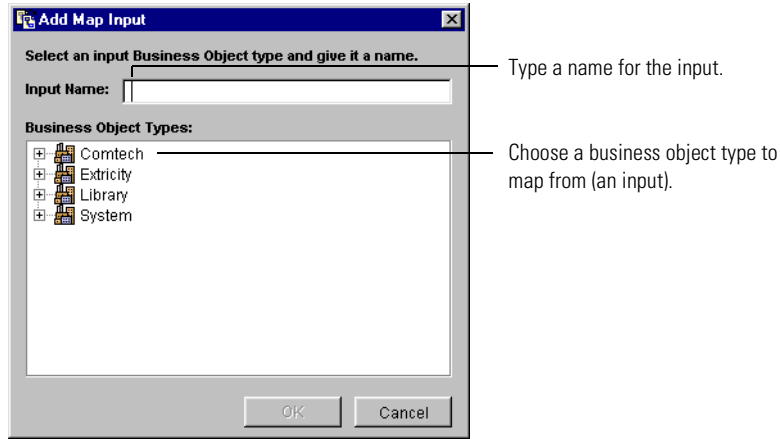
**3** Choose the business object type that you want to map to and click OK.

The business object type you choose will be the output for the map—the one that receives data as a result of the mapping action. The Map Editor appears. The business object type you chose appears on the right side of the Map Editor, in the output area.

When you have several inputs to a map method, you can view them in this list.

This panel displays the output business object type. Double-click a group element to display its contents.

If an input field has been mapped to an output field, the map expression appears here. If groups are mapped, a map method appears here instead.

**4** To add an input, click the Add Map Input button in the Map Editor's Command toolbar.

Or you can choose Add Map Input from the Edit menu. The Add Map Input dialog box appears.



Type a name for the input.

Choose a business object type to map from (an input).

5  Type a name for the input and choose the business object type that you want to map from. Click OK.

The business object type you choose appears in the input panel. All references to elements or sequences in the input business object type are identified with the input name you enter.



This panel displays the current input business object type. Double-click a group element to display its contents.
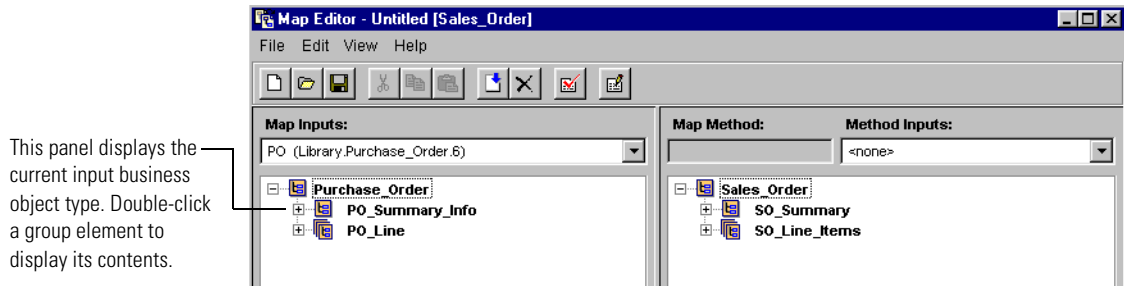
6  Expand the groups in the input and output business object types so that you can see their contents.

7  Add mapping expressions and map methods as needed, and then save the map.

Partner Agreement Manager supports the mapping of data from any position in the input business object type.

These are the most common mapping operations:

■ You can map a single input field to a single output field. See *Mapping fields to fields* on page 76.

■ You can map a repeating group (a sequence) to another group sequence. See *Mapping group sequences* on page 77.

■ You can map a repeating field (a field sequence) to another repeating field, or a non-repeating group to a like group. See *Other mapping combinations* on page 81.
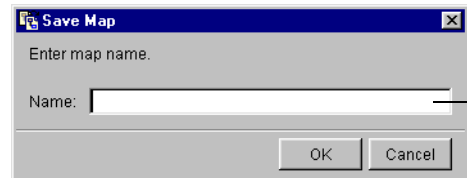
## SAVING A MAP

As you develop a map, it's a good idea to save your work periodically. The first time you save a map, you give it a name, and Partner Agreement Manager lists it in the Business Objects Maps folder. Thereafter, you can use the Save As command to save the map using a new name.

**TIP:** You can use the Save As command to create an editable version of a frozen map.

**To save a map:**

1 Choose Save from the File menu.

The Save Map dialog box appears.

```
┌────────────────────────────────────────────┐
│ ▣ Save Map                              ✕  │
│  Enter map name.                            │
│                                             │
│  Name: ┌──────────────────────────┐         │────── Type a name and click OK.
│        └──────────────────────────┘         │
│                                             │
│                    ┌─────┐  ┌────────┐      │
│                    │ OK  │  │ Cancel │      │
│                    └─────┘  └────────┘      │
└────────────────────────────────────────────┘
```

2 Type a name and click OK.

The map appears in the Maps folder in the Process Manager window's Business Objects folder.

# Mapping fields to fields

In most cases, field-to-field mapping will take care of the bulk of your mapping needs. You can type an expression or have Partner Agreement Manager create one for you. When you drag a single input field to a single output field, Partner Agreement Manager creates the required mapping expression. You can edit the expression if necessary. For example, you might want to add or remove leading zeros from the input data.
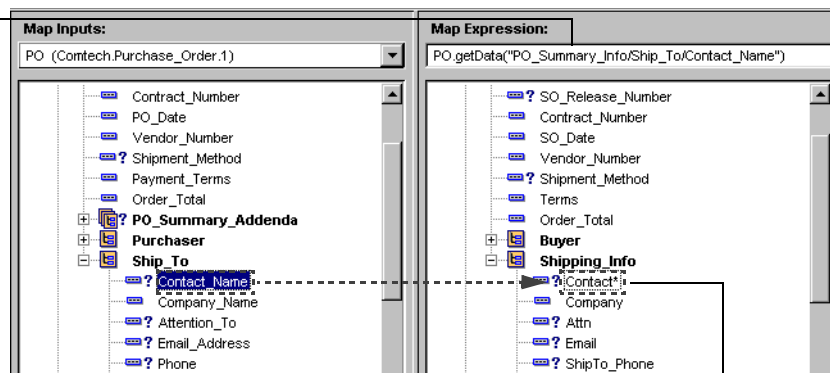
If you type an expression, it must evaluate to a string when the mapping action is run. The expression can include a call to a function defined in the Library that returns a string.

**To map fields to fields:**

▶ Choose the field and type an expression, or drag an input field to the output field you want to map to.

Partner Agreement Manager places an asterisk (*) beside the mapped output field and displays the map expression.

This expression evaluates to a string when the map is run.

| Map Inputs: | Map Expression: |
|---|---|
| PO (Comtech.Purchase_Order.1) | PO.getData("PO_Summary_Info/Ship_To/Contact_Name") |

Map Inputs:
- Contract_Number
- PO_Date
- Vendor_Number
- ? Shipment_Method
- Payment_Terms
- Order_Total
- ? PO_Summary_Addenda
- Purchaser
- Ship_To
    - ? Contact_Name
    - Company_Name
    - ? Attention_To
    - ? Email_Address
    - ? Phone

Map Expression:
- ? SO_Release_Number
- Contract_Number
- SO_Date
- Vendor_Number
- ? Shipment_Method
- Terms
- Order_Total
- Buyer
- Shipping_Info
    - ? Contact*
    - Company
    - ? Attn
    - ? Email
    - ? ShipTo_Phone

The asterisk indicates that the field is mapped.

**Note:** Although JavaScript or VBScript can convert many data constants to strings, it's good practice to enclose all literal strings in quotes. For example, you could type 123 as the map expression for Vendor_Number, but "123" would be a better choice.

# Mapping group sequences

A repeatable group is an element that can appear more than once in its parent group. For example, a line item group in a purchase order is likely to be repeated because companies typically use a single purchase order to manage several items from the same vendor. In contrast, the header fields that uniquely identify a purchase order are not likely to be repeated.

If you want the number of elements in an output sequence to match the number in an input sequence, the mapping code must determine how many times a group occurs in the sequence and then repeatedly create and fill corresponding output elements.

You can manually write all of the code required in a map method to populate a group sequence. You might find it easier to create a *submap* that can be called repeatedly to populate the members of a sequence.

With a submap you can map the fields of a group in the sequence in the same way as you do in the top-level map. In a submap, the inputs come from the parameters to a map method, and the output is a group that will be added to the map method's output sequence.

In the top-level map, the input is PO.

| Map Inputs: | Map Method: | Method Inputs: |
|---|---|---|
| PO (Comtech.Purchase_Order.1) | fill_seq_SO_Line_Items | PO - PO_Line |

- Purchase_Order
  - PO_Summary_Info
  - **PO_Line**

- Sales_Order
  - SO_Summary
  - SO_Line_Items*

The input for this submap is in_PO_Line.

| Map Inputs: | Map Method: | Method Inputs: |
|---|---|---|
| in_PO_Line (Library.Purchase_Order.6) | | <none> |

- PO_Line
  - PO_Line_Number
  - Customer_Product_Number
  - Vendor_Product_Number
  - Description
  - Quantity_Ordered
  - Unit_of_Measure
  - Unit_Price
  - Line_Total
  - Taxable_Flag
  - Requested_Ship_Date

- SO_Line_Items
  - SO_Line_Number
  - Customer_Product_Number
  - Vendor_Product_Number
  - Description
  - Quantity_Ordered
  - Unit_of_Measure
  - Unit_Price
  - Line_Total
  - Taxable_Flag
  - Requested_Ship_Date

Mapping sequences is a three-step process:

- First, you drag the input sequence to the output sequence so that Partner Agreement Manager can create a map method.

- Next, you create a submap and have Partner Agreement Manager generate the code that calls the submap from the map method it just created. You can edit it if you want. You might, for example, want to add conditional logic based on the value of a global variable, or another non-sequence parameter to the method.

- Finally, you map the fields from the input elements to those in the output element.

  In a submap, the easiest way to map the fields is to display the submap and use it to drag input fields to output fields. Partner Agreement Manager creates the mapping expressions.

**NOTE:** If you don't want to use a submap, you can use the Map Script Editor to write the individual mapping expressions for each field. See the *Partner Agreement Manager Script Developer's Guide* for more information about the Business Object API.

**To map group sequences:**

**1** Drag the input sequence to the output sequence.

Partner Agreement Manager displays a map method and a list of all inputs for the method.

For this map method.... ——
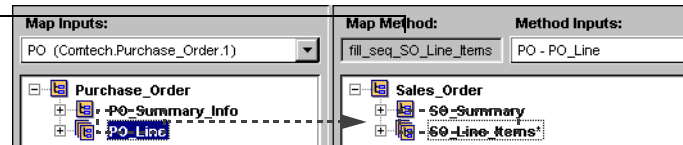
...this is the empty map method as it appears in the Map Script Editor. ——
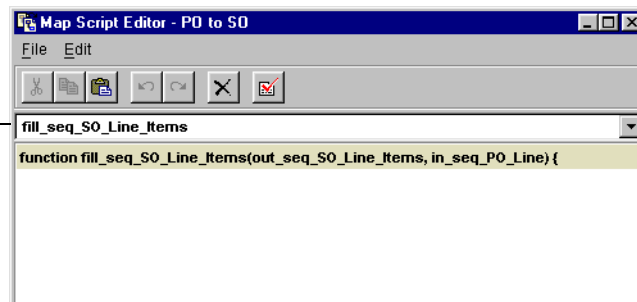


**2** Right-click the output sequence and choose Map Method Properties from the menu that appears.

You can also choose Map Method Properties from the Edit menu. The Map Method Properties dialog box appears.

This is the name of the element to which this method is attached.

The method name always begins with *fill_seq* to indicate that it will fill a sequence.

Click to create a submap.
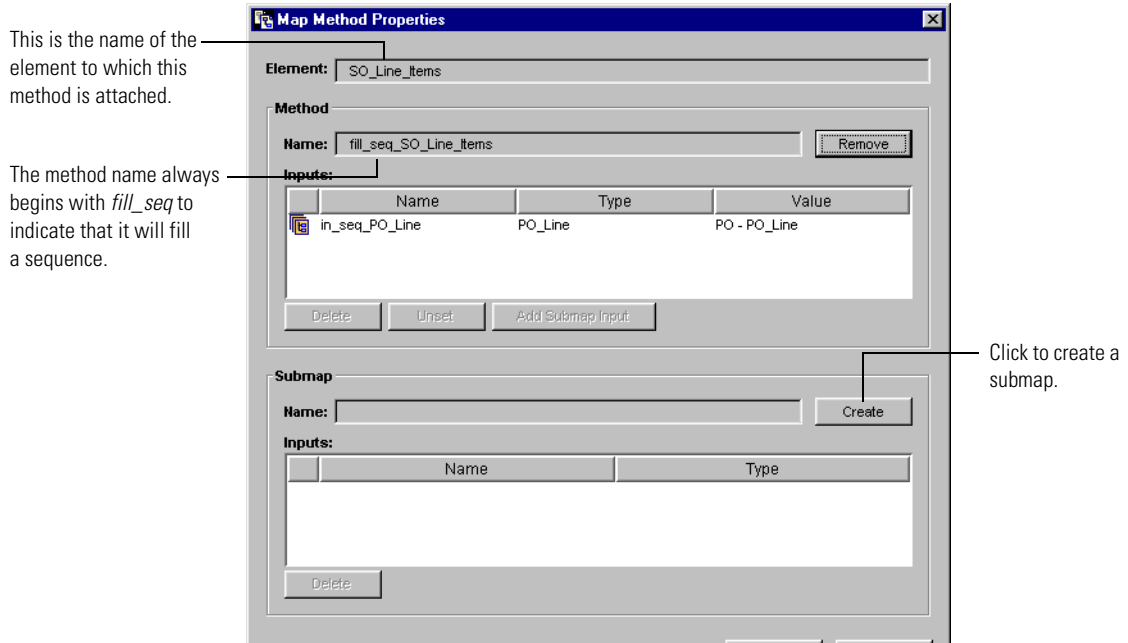


**3** Click Create to create the submap.

Partner Agreement Manager adds a submap (its name is map_*output element name*) with a single input. The input for the submap is in_*input element name*.

This submap...

...fills this output element...

...with data from this input element.
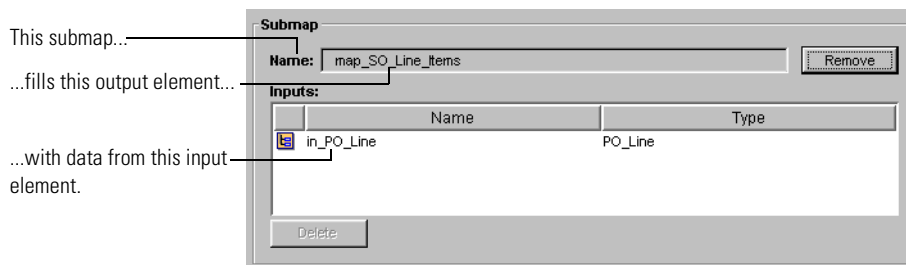


**4** Click OK.

Partner Agreement Manager asks if you want to generate code in the map method that calls the submap.

**5** Click OK.

Partner Agreement Manager adds code to the map method. If you want, you can click Cancel and generate the code later. To generate the code later, right-click the output element and choose Generate Submap Call Code from the menu that appears. Or select the output element and choose Generate Submap Call Code from the Edit menu.

This is the default generated code.

**fill_seq_SO_Line_Items**

**Sub fill_seq_SO_Line_Items(out_seq_SO_Line_Items, in_seq_PO_Line)**

```
Dim new_SO_Line_Items
Dim i, seq_length

seq_length = in_seq_PO_Line.length()
For i = 0 to seq_length - 1
  Set new_SO_Line_Items = out_seq_SO_Line_Items.newElementAt(i)
  map_SO_Line_Items new_SO_Line_Items, in_seq_PO_Line.getElementAt(i)     This is the call to the submap.
Next
```

6  To view the submap and map its output element, right-click the output element and choose View Submap from the menu that appears.

You can also choose Submap from the View menu and choose the submap from the menu that appears.

In the top-level map, the input is PO. At this level, the input is a PO_Line element named in_PO_Line.

| Map Inputs: | Map Method: | Method Inputs: |
|---|---|---|
| in_PO_Line (Library.Purchase_Order.6) | | \<none\> |

```
☐ PO_Line
    PO_Line_Number
    Customer_Product_Number
    Vendor_Product_Number
    Description
    Quantity_Ordered
    Unit_of_Measure
    Unit_Price
    Line_Total
    Taxable_Flag
    Requested_Ship_Date
⊞? PO_Line_Addenda
```

```
☐ SO_Line_Items
    SO_Line_Number
    Customer_Product_Number
    Vendor_Product_Number
    Description
    Quantity_Ordered
    Unit_of_Measure
    Unit_Price
    Line_Total
    Taxable_Flag
    Requested_Ship_Date
⊞? SO_Line_Addenda
```

7  Map the elements in the submap.

You can choose an output element and type a mapping expression, or you can drag an element from the input panel onto a field in the output panel. When you drag an element, Partner Agreement Manager creates the mapping expression.

When you drag an input field to an output field...



Partner Agreement Manager creates the map expression.

**8** To view the top-level map again, choose Top-Level Map from the View menu.

**9** Save the map before continuing.

## OTHER MAPPING COMBINATIONS

You can also map a field sequence to a group or another field sequence, a group to a field, field sequence, or a group, or any other possible combination of the above elements. Because these combinations are rare, you must write code to populate the output elements.
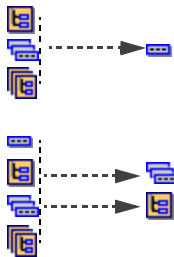
Dragging a group or repeatable field to a field generates code in the field's map expression that gets the Element or ElementSequence from the input object. Because map expressions must evaluate to a string, such an Element or ElementSequence must be passed to a Library function you have written that returns a string. See the *Partner Agreement Manager Script Developer's Guide* for more information.

If you map anything to a field sequence or a group, you must use a map method, rather than a map expression, to populate the output element or element sequence. Partner Agreement Manager creates the map method for you, but you must write the code that populates the output object fields. See the *Partner Agreement Manager Script Developer's Guide* for more information.

fill_Buyer

Sub fill_Buyer(out_Buyer, in_Purchaser) —————— Partner Agreement Manager creates
|                                                  an empty map method for non-
                                                   repeating groups or field sequences.

## SETTING MAP METHOD PROPERTIES

A map method is a logical, reusable block of code that fills a field sequence, a group, or a group sequence. Methods have parameters (inputs) that can take on different values depending on where you call the method from.
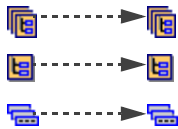
Partner Agreement Manager creates a method for you when you drag an input onto a field sequence, a group, or a group sequence in the output panel. The name of the method is fill_*groupname*, fill_seq_*fieldname*, or fill_seq_*groupname*, depending on the type of input element. The Map Editor determines the name and order of a map method's parameters.

The first parameter is always a reference to the group element or sequence in the output business object. The other parameters are references to elements or element sequences in the input business objects. The output parameter has an out_ or out_seq_ prefix; any input parameters have an in_ or in_seq_ prefix.

**Map Inputs:**

PO  (Library.Purchase_Order.6)

- Purchase_Order
  - PO_Summary_Info
    - PO_Number
    - PO_Release_Number
    - Contract_Number
    - PO_Date
    - Vendor_Number
    - Shipment_Method
    - Payment_Terms
    - Order_Total
    - PO_Summary_Addenda ——————— If you drag this group to the output panel...

...this is the resulting input that
Partner Agreement Manager creates.

**Inputs:**

| | Name | Type | Value |
|---|---|---|---|
| | in_seq_PO_Summary_Addenda | PO_Summary_Addenda | PO - PO_Summary_Info/PO_Summary_Addenda |

The path to an input element is made up of the map input's name, the hierarchy of nested groups (if any), and the name of the input field or group.

**PO - PO_Summary_Info/PO_Summary_Addenda**

Map Input name     Path from top-level element to input element
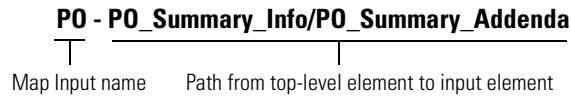
As you work with maps, you can manage the input parameters of a map method. You can also manage the values those inputs are set to when they are called to map data to a particular part of the output business object.

**To set map method properties:**

1 Right-click the output group or field sequence whose method you want to alter.

2 Choose Map Method Properties.

The Map Method Properties dialog box appears. If the element already has a method associated with it, the method's inputs appear. If there is no method associated with the element, the dialog box shows only the element name.



Choose an input to delete it or unset its value.

Click to delete the map method. If there's no method associated with the element, this button is Create.
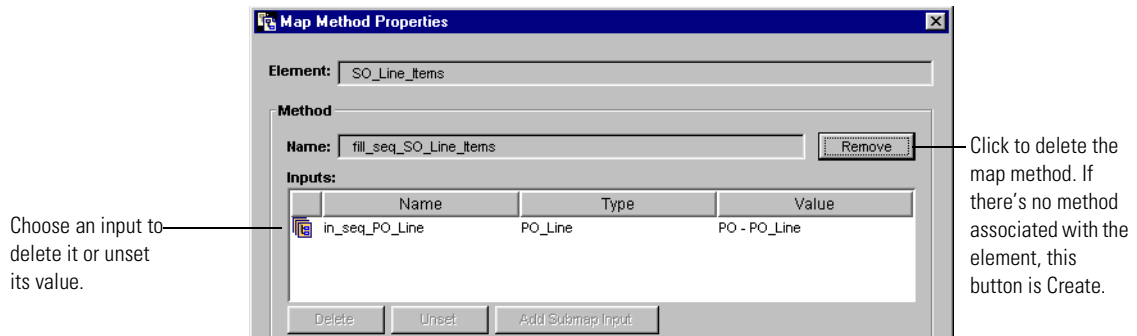
3 Make the appropriate changes.

■ To add a map method, click Create.



Click to create a map method.
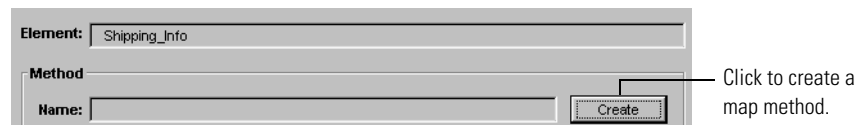
Partner Agreement Manager adds a method named fill_*elementname* or fill_seq_*elementname*.

■ To unset the value for an input, choose the input and click Unset.

- To delete a map method, click Remove.

- To delete an input, choose the input and click Delete.

**NOTE:** You might use the Map Method properties to unset a map method input value when the same map method is called from more than one location in the output business object, and there is no appropriate value for one of the parameters for one of those locations. In this case, the code in the map method must be able to accept null input values.

**4** Click OK.

## VERIFYING A MAP

At any point as you're developing a map, you can check its JavaScript or VBScript syntax for errors. If you run the Syntax Checker from the Map Editor window, it checks the syntax of all mapping expressions, all map methods, and all submap expressions. It also checks the syntax of a global variables or library items you might have added using the Map Script Editor.

If you run the Syntax Checker from the Map Script Editor, it checks the syntax for the current page.

**To verify a map:**

▶ From the Map Editor or the Map Script Editor, click the Verify Map Syntax button in the Command toolbar.

The Syntax Checker runs and reports back with the location by line number of the first error it encounters.

| Syntax Checker |
|---|
| ❌ Syntax Error Expected identifier in line no 2 |
| OK |
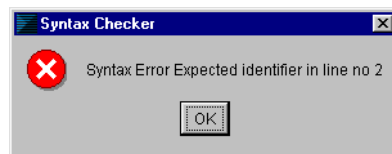
This message shows the line number for the first error it encounters.

| Syntax Checker |
|---|
| ⓘ No syntax errors found! |
| OK |
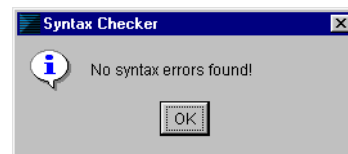
This message appears when there are no syntax errors.

# Testing a map

With Partner Agreement Manager, you can test a map to make sure that it produces the desired results. Using the Map Execution window, you can create an instance of an input business object type or set context variables so that you can run a map. After you have created a business object instance you can export it for use in later testing sessions. You can also edit a map and test it without having to save your changes. This lets you determine if the changes produce the desired result before you alter the map.

You use the Map Execution window to enter test data and test a map. You must add enough test data to mimic the behavior of a process instance. The Map Execution window is divided into three panels:

- The Script panel displays the code that has been generated by the Map Editor.

- The Catalog panel displays the Business Object Catalog so that you can see the elements in any given element definition set or business object type.

- The inputs panel displays the inputs into the map. You can choose an input and set values for a test instance of a business object type.

You can resize the panels by dragging their borders.

Toolbar buttons give you quick access to common Map Execution commands.

This panel shows the map code.

Drag a panel's border to resize the panel.

This panel shows the Business Object Catalog.

This panel shows the map's inputs.

```
// ---- map script entry point ----
function main(out_Sales_Order, PO) {
pre(PO);
map_Sales_Order(out_Sales_Order, PO);
post(out_Sales_Order);
}

// ---- element type map: SO Line Items ----
function map_SO_Line_Items(out_SO_Line_Items, in_PO_Line) {
//        map expressions
out_SO_Line_Items.setData("SO_Line_Number", in_PO_Line.getData("
//        map method calls
}

// ---- element type map: Sales_Order ----
function map_Sales_Order(out_Sales_Order, PO) {
//        map expressions
out_Sales_Order.setData("SO_Summary/Shipping_Info/Contact_Name",
```

Catalog
- Comtech
- Extricity
- Library
- System

| Name | Type | Value |
|------|------|-------|
| PO | Library.Purchase_Order.6 | <null> |

READ  Ln 31, Col 49

The Command toolbar at the top of the Map Execution window lets you verify or execute a script, create an empty business object instance, edit or clear the value of a variable, or import or export a business object instance.

Execute Map
Create Empty Business Object Instance
Verify Script
Export Business Object Instance
Edit or Clear Value of Variable
Import Business Object Instance
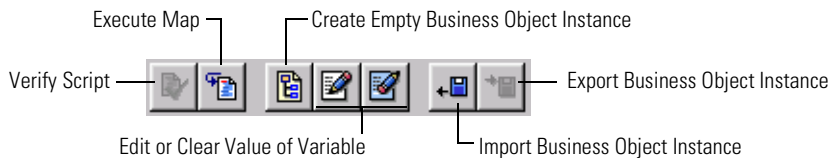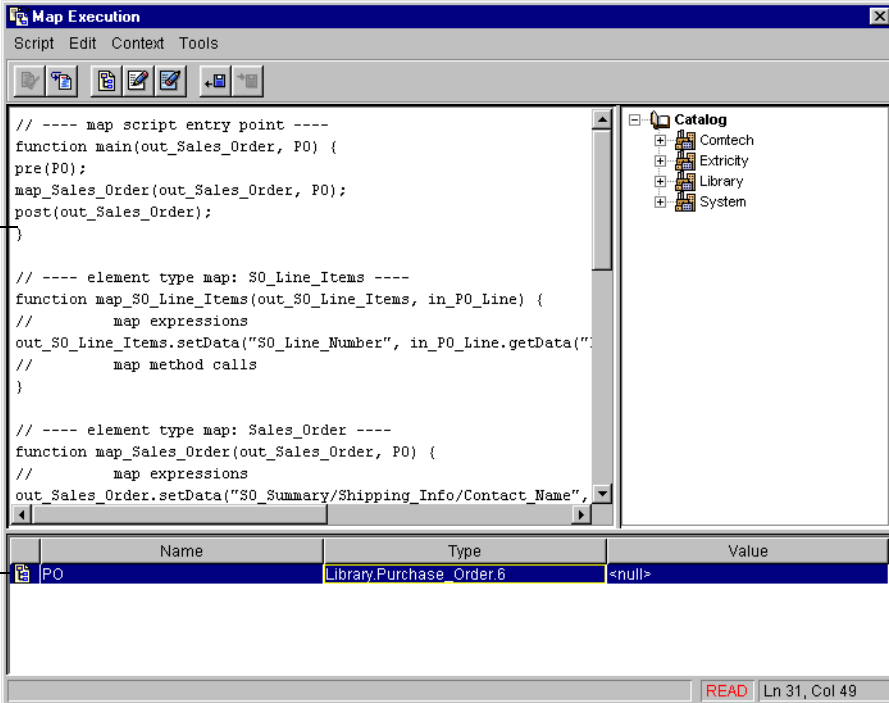
A command in the Map Execution Edit menu lets you to go to a specific line number. With commands in the Context menu, you can create an empty business object instance, edit the value of a variable, clear the value of one or all variables, and import or export business object instances.

**NOTE:** You must freeze the input and output business object types used in the map before testing the map.

**To test a map:**

**1** Choose Test from the Map Editor File menu.

The Map Execution window appears.

This panel shows the map code.

This panel shows the map's inputs.



**2** To enter test data, double-click one of the inputs for the map.

You can also choose the input and click the Edit Variable button in the Command toolbar, or select the input and choose Edit from the Context menu.

If you have not already started entering data for an input, you can choose the input and click the Create Business Object Instance button in the Command toolbar or choose Create from the Context menu.

A window for entering test data appears. You use this window to enter values for all of the required fields in a map input. If the map includes a group or field sequence, you can add several repeatable elements to the test data.
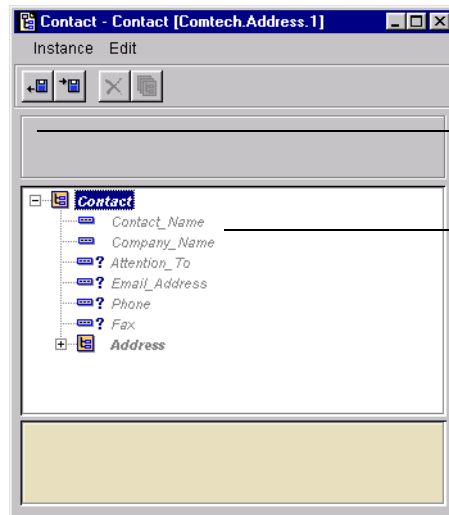


Choose an element in the list below and type a value here.

These are the elements in the input business object type.

**3** To enter data for an element, choose the element in the list and type data in the entry area below the Command toolbar.

A group element name appears in red if it contains required elements that still need test data. After you enter data, you can export it to an XML file so that you can use the same data in another testing session.
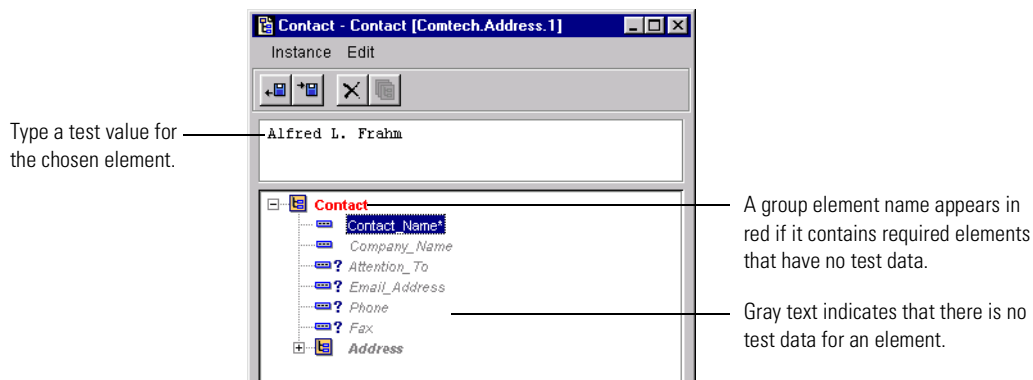


Type a test value for the chosen element.

A group element name appears in red if it contains required elements that have no test data.

Gray text indicates that there is no test data for an element.

**4** Add all required test data and close the window you used to enter test data.

You can export the data you've entered and import previously exported data. If the map input contains a group or field sequence, you can add several repeatable elements.

■ To export test data, click the Export icon, type a name for the business object instance, and click Save.

■ To import previously exported test data, click the Import icon, choose the file that contains the business object instance, and click Open.

■ To add a repeatable element, choose a group sequence element and click the Add Repeatable Element button in the Command toolbar.
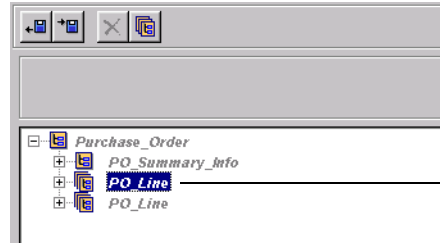


You can add repeatable elements to test the populating of group sequences.

5 To test the map, click the Execute button in the Command toolbar, or choose Execute from the Tools menu.

The map runs, using the test data you entered or imported. After the map finishes executing, a results window appears. This window displays the elements in the output business object. A red group element name indicates test data as a result of the map.
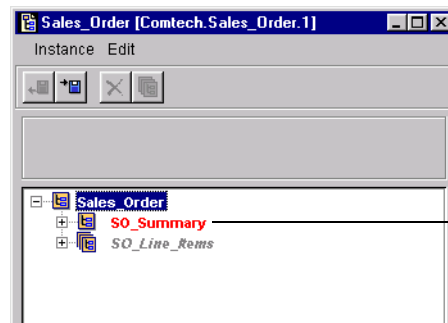


A group element name appears in red if it is missing required data.

A gray group element name indicates no test data. You can choose a field to see the data set by the map.



This is the resulting test data for the chosen element.

Gray text indicates that there is no test data for an element.
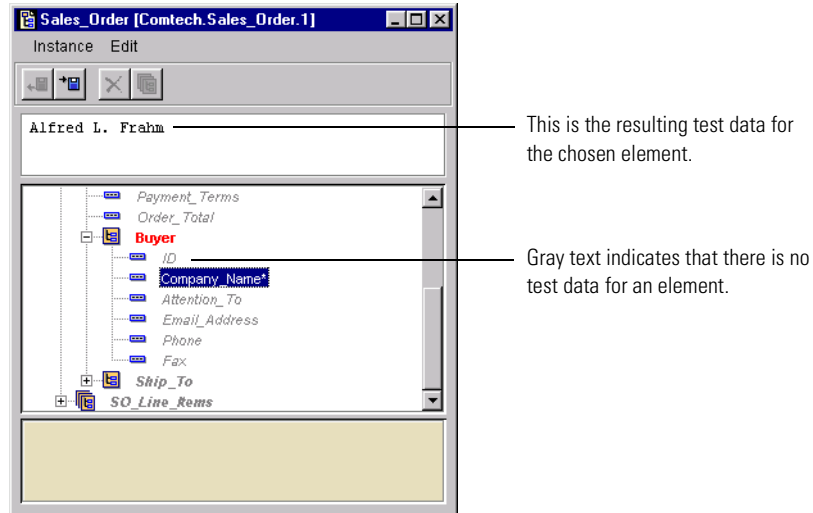
> **6** To export the results of a map for use in other testing sessions, click the Export icon, type a name for the file, and click Save.

> **7** Close the Map Execution window when you finish testing the map.

## Freezing a map

As you define maps, you might find that you want to make numerous changes. With Partner Agreement Manager, you can revise maps freely before committing to version control.

When you reach a point where you're ready to activate a private process that includes a mapping action for an installed public process, you freeze the map to institute version control. Its state changes from Edit to Read, which means that no further changes can be made to the current version. Although you can refer to a map in a process before it is frozen, the private process can't be active in an installed public process until you freeze the map.

From that point onward, you must save a map as a new version when you want to make changes. If you must make changes to a frozen map, use the Save As command to create a new version of the map. You can then revise the map and replace the prior version with the new one in any process that refers to it.

### To freeze a map:

**1** Open the map in the Map Editor.

**2** Choose Freeze from the File menu.

The state in the Process Manager window changes from Edit to Read.

| Name | Output Type | State |
|------|-------------|-------|
| SalesOrd to PurchOrd | Purchase_Order (Comtech.Purchase... | Read |
| PO to SO | Sales_Order (Comtech.Sales_Order.1) | Edit |

Read state means that the map is frozen.

# DEFINING PUBLIC PROCESSES

Read this chapter for information about how to define a public process. For an introduction to processes, see *About processes* on page 16. This chapter includes these sections:

# About defining public processes

A process operates on two levels: the public process determines the flow of messages and information between partners, and the private process defines the internal private actions that each partner performs at its steps in the public process.

## About the Process window

You use the Process window to model the flow of steps and messages in the process.
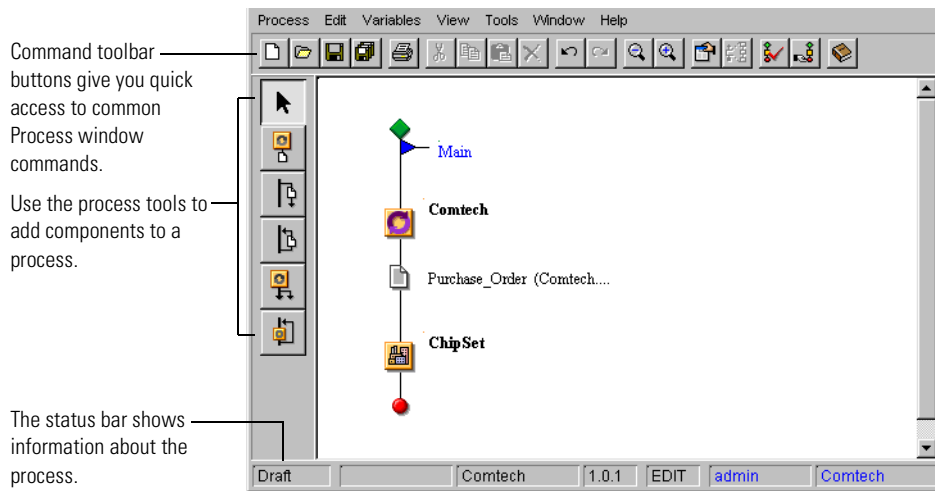
Command toolbar buttons give you quick access to common Process window commands.

Use the process tools to add components to a process.

The status bar shows information about the process.

The Process window includes two toolbars that give you access to the most frequently used commands and process components.

## About the Command toolbar

The Command toolbar at the top of the Process window gives you access to common File and Edit menu commands, and lets you zoom, set process properties, add or show private processes, verify the public process, view or change the status of the process, and view the Business Object catalog.
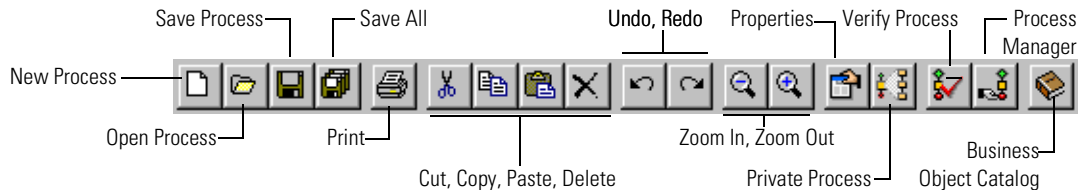


Partner Agreement Manager is preset to display the Command toolbar whenever you open a Process window. You can hide the toolbar if you don't need it.

### To hide the Command toolbar:

▶ Choose Toolbar from the View menu to hide it.

▶ Choose Toolbar from the View menu a second time to display it.

## About the Public Process toolbar

Using the toolbar on the left side of the Process window, you can add components to the process. Partner Agreement Manager displays the toolbar when you work with a new or draft process.
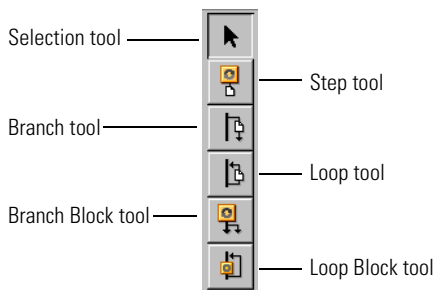


For a complete description of each component that appears in the toolbar, see *About public process components* on page 98.

**NOTE:** The Process toolbar doesn't appear if a process has already been distributed. This is because Partner Agreement Manager doesn't permit you to edit a public process after it's been distributed. If you want to revise a distributed process, open the process and choose Save As or Save As New Version from the Process menu. You can then edit the new version.

### ABOUT THE STATUS BAR

The status bar at the bottom of the Process window shows information about the current user and the state of the process.
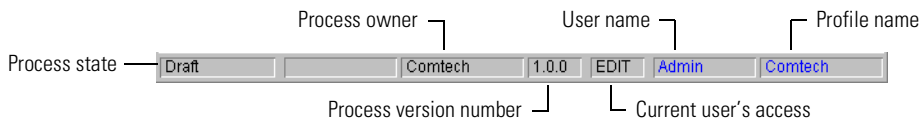


- The process state shows whether the process is a draft, is being implemented, or is activated. See *About managing processes* on page 204 for a more complete description of the possible states.
- The process owner is the profile name for the user who created the process.
- The process version is a three-digit number. Each time you save a draft version of a process, the last digit increments. After you distribute a process and save new versions of it, the second digit increments. The first digit changes when you save a new version of an activated process.

The first digit changes when you save a new version of an activated process. —— **2. 3. 5** —— The last digit changes when you save a draft version.

The second digit changes when you save a new version of a distributed process.

- Access is the current user's access to the process. With Read access, the user can view a process but not edit it. Users with Read access can use the Save As command to save a new, editable version of a process. With Edit access, the user can edit and distribute a process.
- The user name is the login name for the current user.
- The profile name is usually the company name.

Partner Agreement Manager is preset to display the status bar whenever you open a Process window. You can hide the status bar if you don't need it.

### To hide the status bar:

▶ Choose Status Bar from the View menu to hide it.

▶ Choose Status Bar from the View menu a second time to display it.

## About the menu bar

The Process menu bar appears at the top of the window.

- The Process menu contains commands for creating, opening, saving, closing, and printing processes, and changing process properties.

- The Edit menu contains commands for undoing or redoing an action, cutting, copying, pasting, and deleting steps, clearing all steps, replacing partner or business object names, displaying or activating private processes, working with partner groups, and viewing item properties.

- The Variables menu contains commands for creating, editing, or deleting a variable, or clearing all variables.

- The View menu contains commands for changing your view of the Process window's contents, and for hiding or showing the toolbar, status bar, Business Object catalog, or variables.

- The Tools menu contains commands for verifying a process or displaying the Process Distribution Manager.

## Zooming in the Process window

Partner Agreement Manager provides several zoom controls that let you resize the contents of the Process window as you work. You might want to zoom out to see an entire process, or zoom in to see the details of a step or block. Zoom settings are 25%, 50%, 100%, 200%, and 400%.
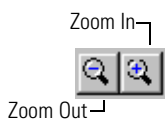
Zoom In⌐

Zoom Out⌐

### To zoom the contents of the Process window:

▶ Click the Zoom Out or Zoom In buttons in the Command toolbar to zoom from one setting to the next.

▶ Right-click anywhere in the Process window and choose a zoom setting from the View menu that appears.

▶ Choose a zoom setting from the View menu (in the main menu bar).

**NOTE:** To display a small version of the entire process in a separate window, choose Thumbnail from the View menu.

## Printing process information

To help you get an overview of your process, Partner Agreement Manager lets you print the contents of the Process window as you work.

**To print process information:**

▶ Choose Print from the Process menu, or click the Print button in the Command toolbar.

## About public process components

When you build a public process, you work with five types of components: steps, branches, branch blocks, loops, and loop blocks. A step or block consists of an action taken by a partner, a path, and at least one message that transfers control of the process from one partner to another.

Partner Action

Path

Business object

Step                    Branch

Loop
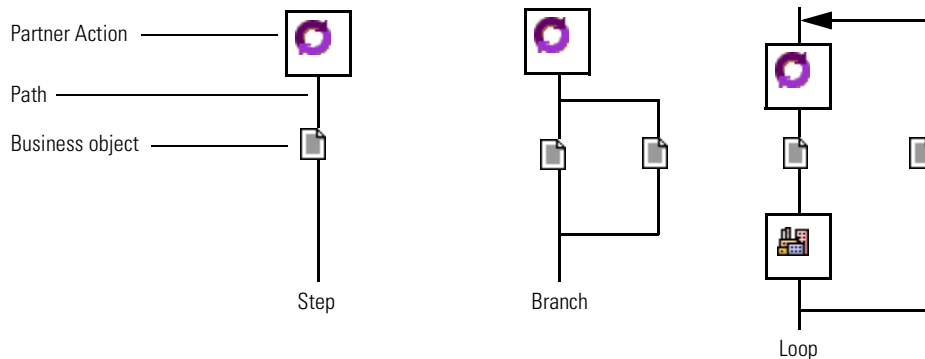
A *step* is the simplest component. It consists of a partner action, a message, and a direct path to the next action in the process. You can use a step for a simple task, such as sending a message to a partner, or you can include a step as part of a larger, more complex task. Here's an example of a step.

Partner2 receives payment from Partner1 and...

...sends an acknowledgment of receipt.
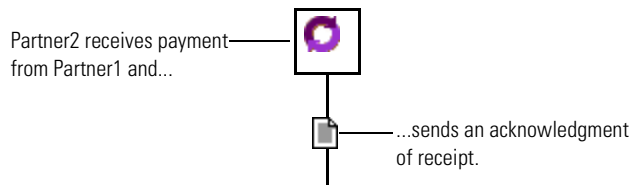
Partner Agreement Manager uses three different icons to represent partner actions in a process.

| This icon | Represents |
|---|---|
|  | The current partner. |
|  | Any other partner in the process. |
|  | A partner group (as opposed to a single partner). See *Setting up a partner group* on page 106 for more information. |

 A *branch* consists of an alternate or additional path and a message. Depending on how you set the branch's logic, you can use a branch to indicate an alternate course of action (the logic is XOR) or two actions that need to occur at the same time (the logic is AND).

For example, you might use a branch for an inventory check: one branch represents the steps to take when the part is in stock and the second branch shows the steps to take when it's not.

Partner2 receives a purchase order and checks inventory. The part is either in stock or backordered.

The logic is set to XOR.

Path 1: If the part is in stock, Partner2 sends Partner1 a purchase order confirmation.

This branch shows two alternate courses of action.

Path 2: If the part is backordered, Partner2 sends Partner1 a backorder notice.
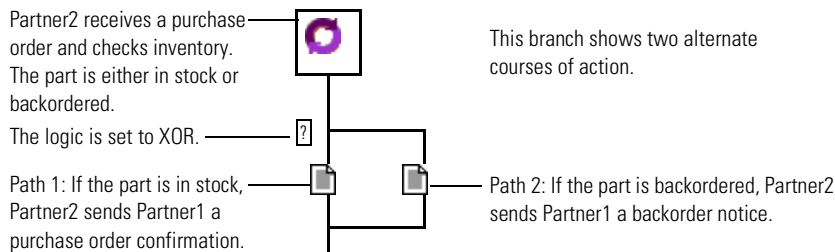
Or you can use a branch when you want to perform tasks simultaneously. For example, you might use a branch in response to an action by another partner, where each path represents a message you send when you receive a purchase order.

Partner2 receives a purchase order from Partner1.

The logic is set to AND.

Path 1: Partner2 sends an acknowledgment to Partner1.

This branch shows actions that occur simultaneously.

Path 2: Partner2 sends an inventory status notice to Partner1.
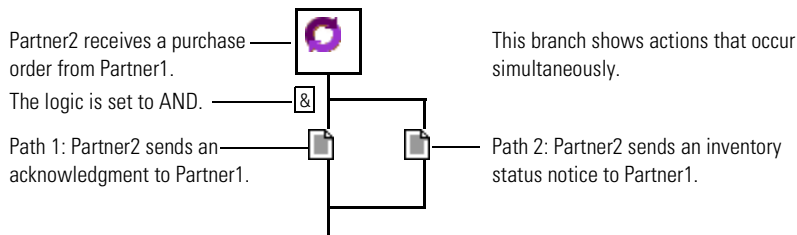
You can add as many branches as you need. For example, you can use branches to create three alternate courses of action. Similarly, you can add branches if you want three actions to take place at the same time.

This branch block shows three alternate courses of action.

You can add as many branches as your process requires.
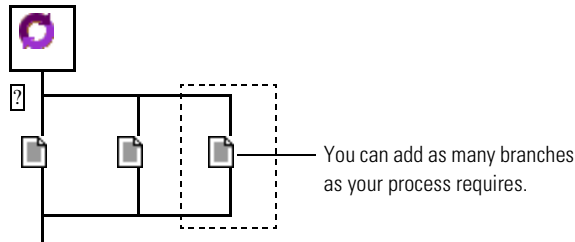
You can also add a branch elsewhere in a process: within another branch, within a loop, or around either.

This illustration shows two nested branches.

This branch represents one possible course of action.

This branch represents a second possible course of action, which is in turn divided into two additional branches.

This is a second branch added using the branch tool.
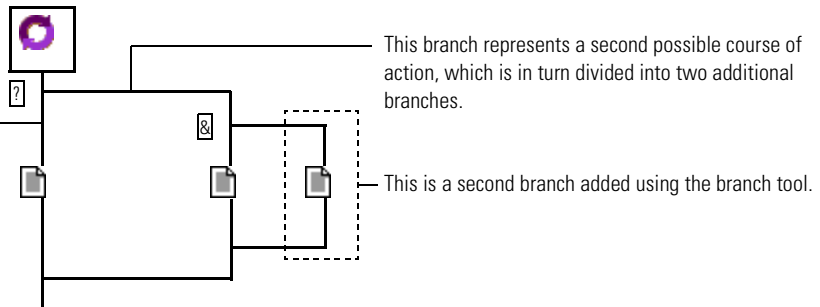
Like a branch, a *loop* adds a second path and a message to a process. With a loop, one path goes to the next step in the process, while the other loops back to repeat one or more earlier steps. You use a loop when you want to repeat one or more steps in the process.

For example, you can use a loop to keep resubmitting a request for clarification until a product specification is complete. One path leads to the next step in the process. The other path loops back and keeps repeating the sequence of steps until the conditions you specify have been met.

Partner1 receives a specification clarification notice from Partner2 and processes it.

Partner1 sends a revised specification to Partner2.

Partner2 evaluates the revised specification to determine if it's complete.

If the revised specification is complete, Partner2 sends an acknowledgment to Partner1.

If the revised specification isn't complete, Partner2 sends Partner1 a new specification clarification notice. Partner1 receives the notice, sends another revised specification, and the process keeps looping until Partner2 determines that the specification is complete.

You'd use a branch tool or loop tool when you just want to add a branch or loop around or within an existing structure.

A *branch block* consists of an initial action and at least two different paths. You can think of a branch block as a branch and two partner actions. The branched paths then merge and the process continues. Depending on how you set the block's logic, you can use a branch block to indicate two alternate courses of action (the logic is XOR) or two actions that need to occur at the same time (the logic is AND). Partner Agreement Manager is preset to create a branch block with two branches, but you can add additional branches as needed.

A branch block includes an initial action, branching paths, and a concluding action.
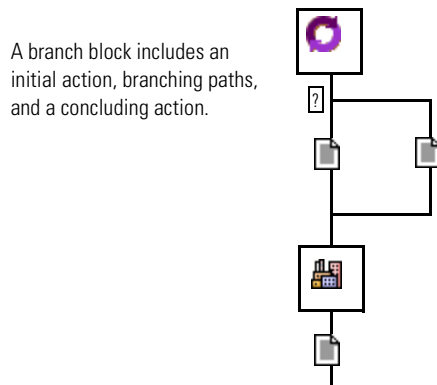
Like a branch block, a *loop block* also consists of an initial action and two alternate paths. In the loop block, one path goes to the next step in the process, while the other loops back to repeat earlier steps. You can think of a look block as a shortcut for a loop and two partner actions.

A loop block includes an initial action, a looping path, and a concluding action.
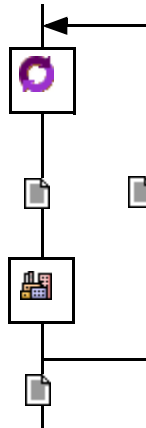
You'd use the branch block tool or the loop block tool when you know that you want to insert the whole structure and then fill it in.

## Combining components

With Partner Agreement Manager, you can combine branches and loops to model even the most sophisticated processes. For example, you can combine the earlier loop example with a branch to handle the case when a complete specification is submitted the first time.

Partner2 receives a specification from Partner1 and evaluates it to determine if it's complete.

Path 1: If the specification is complete, Partner2 sends an acknowledgment to Partner1.

Path 2: If the specification isn't complete, Partner2 sends Partner1 a specification clarification notice.

Partner1 processes the clarification notice from Partner2.

Partner1 sends a revised specification to Partner2.

If the revised specification isn't complete, Partner2 sends Partner1 a new specification clarification notice, and the process keeps looping until Partner2 determines that the specification is complete.

Partner2 evaluates the revised specification to determine if it's complete.

If the revised specification is complete, Partner2 sends an acknowledgment to Partner1.

Partner1 receives and processes the acknowledgment.

## Creating a new process

Each new process you create begins with a starting point (a green diamond), and a default *Main* path. You can rename the path or add a description that other partners can see when they view the properties for the path.

**NOTE:** Typically, the owner of the process (the partner who creates it) is the *hub* partner, as well as the partner who is most likely to run a similar type of process with other partners.

**To create a new process:**

1 Click the New Public Process button in the Command toolbar, or you can choose New Process from the Process menu.

The Public Process window appears.
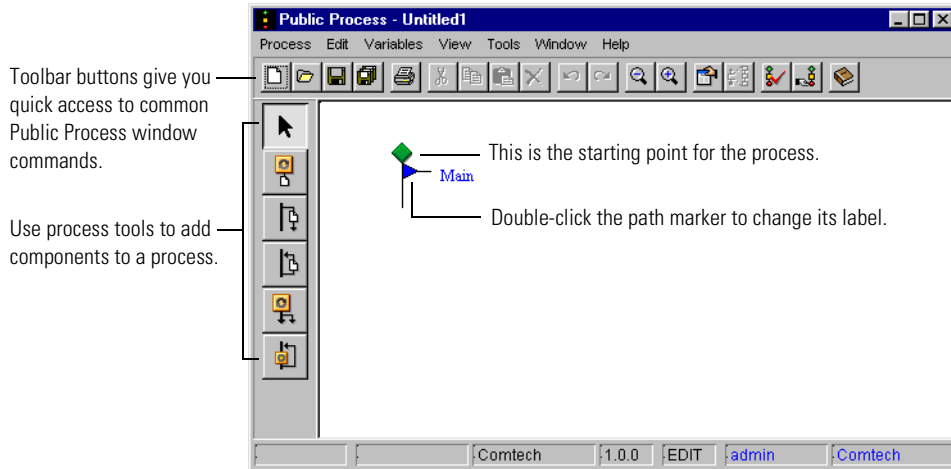
Toolbar buttons give you quick access to common Public Process window commands.

Use process tools to add components to a process.

This is the starting point for the process.

Double-click the path marker to change its label.

2 To relabel the main path, double-click the path marker.

You can also right-click the path marker and choose Properties. The Label Properties dialog box appears.

Type a new label.
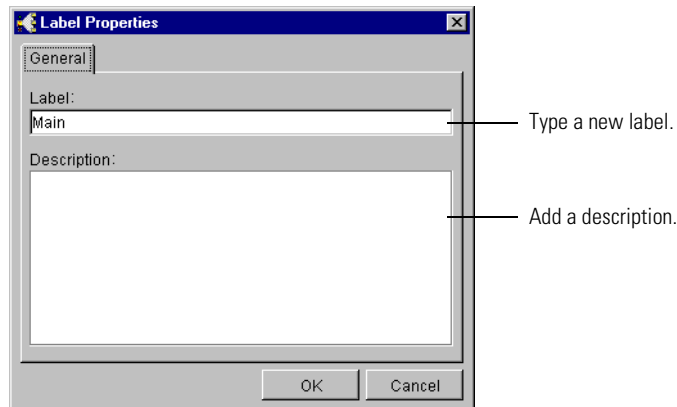
Add a description.

3 Type a new label and an optional description, and click OK.

Partner Agreement Manager displays as many characters of the label as will fit on one line in the Process window. You can leave the label as is, and change it later if necessary. You can use the description to annotate the process.

# Setting process properties

Each process you create has a name and an optional description. The process name appears in the Process window's title bar and in the Process Manager window. Anyone viewing the process can display its description. You might use the description to outline the purpose of the process.

In addition to the name and description, you can set security properties for a process. Partner Agreement Manager is preset to use the security options you set in your Partner Agreement Manager profile. You can override those settings for each process.

**To set process properties:**

1   With nothing selected, click the Properties button in the Command toolbar.

    You can also right-click anywhere in the Process window and choose Properties.

    The Process Properties dialog box appears.
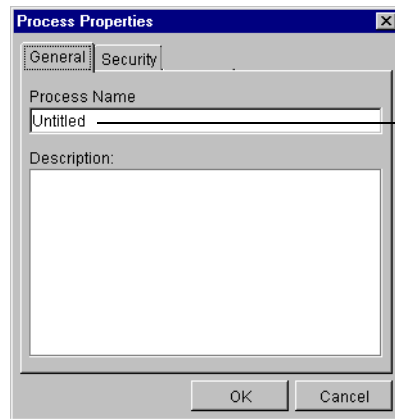


You can name the process and add a description.

2   Type a name for the process and an optional description.

3   Click the Security tab to set security options for the process.

The Security tab appears.



Select the security options you want.

The algorithms you can choose depend on the security options you select.
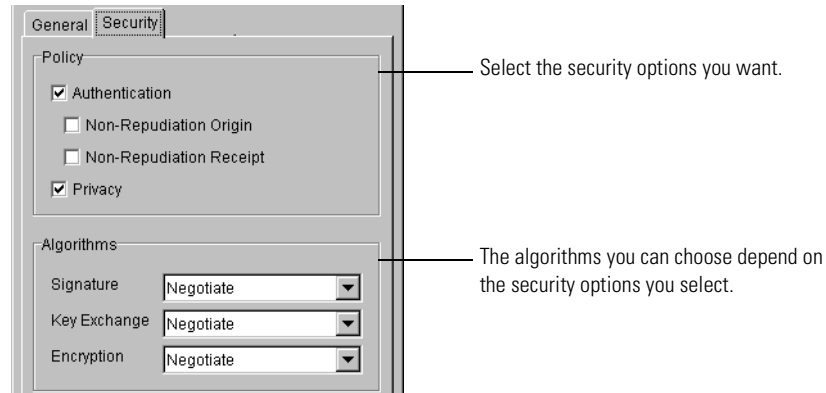
4 Select the security options you want to apply and choose a set of algorithms.

If you turn on Authentication, you can also choose non-repudiation settings. Each policy level has a preset group of algorithms associated with it. You can choose different algorithms, or set Partner Agreement Manager to negotiate based on a common agreement between your profile and your partner's profile. If you turn on Authentication, you can choose Signature algorithms. If you turn on Privacy, you can select Key Exchange and Encryption algorithms.

**NOTE:** The policy defaults to the policy defined in your Partner Agreement Manager profile, but you can override your profile policy for individual processes. For more information about security profiles, see the *Partner Agreement Manager Administrator's Guide*.

5 Choose the algorithms you want to use.

6 Click OK.

## SETTING UP A PARTNER GROUP

In the interests of efficiency, you might want to design a public process in which different partners fill the same role at different times. For example, you might design a generic purchasing process that can work equally well with any of your suppliers—not just one specific supplier. Instead of duplicating a public process and substituting a different partner's name, Partner Agreement Manager lets you create a partner group for design purposes and designate which partner is the participant when you run each instance of the process.

**Important:** You can define only one partner group per process, and the group can only be used in that process. If a partner is a member of a group, you can't assign actions to that partner outside of the group.

When you set up a partner group, you assign it a name. You can then add partners, or return and add them later. A partner must be set up as a Channel partner—with an exchange of profiles—before you can add that partner to a group. See the *Partner Agreement Manager Administrator's Guide* for more information.

Later, when you run instances of this process, you specify which partner in the group is participating on an instance-by-instance basis.

**To set up a partner group:**

1 In the Process window, choose Partner Groups from the Edit menu.

You can also right-click anywhere in the Process window and choose Partner Groups from the menu that appears.

The Partner Groups dialog box appears.



Click to add a partner to a partner group.

Click to delete a partner from a group.

Click to create a new partner group.
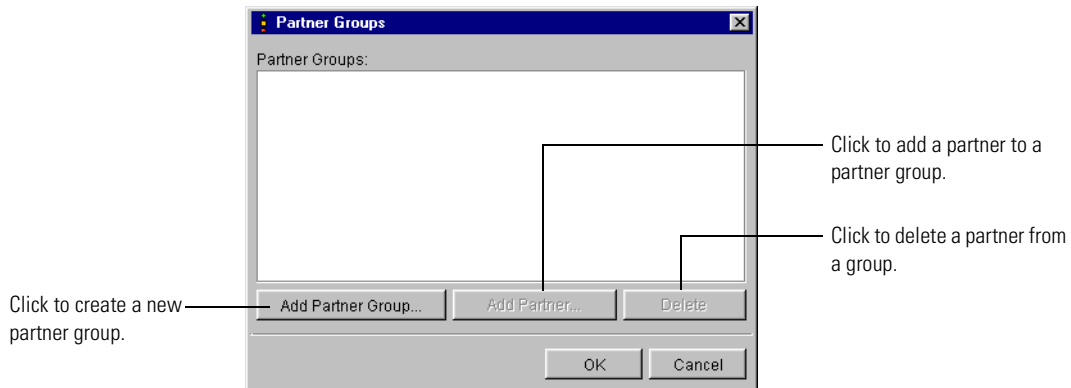
2 Click Add Partner Group to create a group.

The Add Partner Group dialog box appears.



Type the group name.
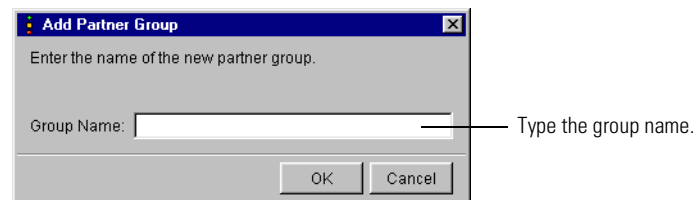
**3** Type a name for the group and click OK.

**4** If you want to add partners to the group, click Add Partner.

The Add Partner dialog box appears. You can add as many partners as you want. For example, you can create a group with no partner members, and then return to add partners after you add them as Channel partners.



Choose the partners to add.
Control-click to choose more
than one partner.
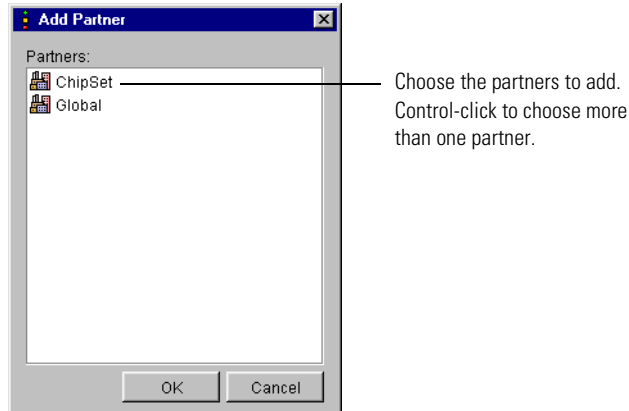
**5** Choose the partners to add and click OK.

The partners are listed in the Partner Groups dialog box.



Open the group to see the names of the
member partners.
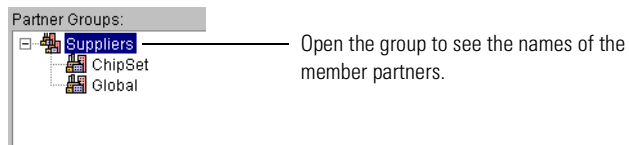
**6** Click OK in the Partner Groups dialog box.

## ADDING PARTNERS TO A GROUP

When you first create a group, you can add any partners who have been set up through Partner Agreement Manager profile exchange. If you add Channel partners at a later date, you can return to a group and add the new partners.

**To add more partners to a group:**

**1** Open the Partner Groups dialog box.

**2** Choose the group and click Add Partner.

**3** Choose the partners and click OK.

**4** Click OK in the Partner Groups dialog box.

### Deleting partners

You can remove partners from a group without affecting the flow of a process. Deleting partners from the group does not remove them from your Partners folder; it removes them from the group.

**To delete partners:**

**1** Open the Partner Groups dialog box.

**2** Choose the partner and click Delete.

**3** Click OK.

## Adding a step

To help design processes that have logical integrity, use Partner Agreement Manager to insert steps and other components only where they might logically fit. For example, you can add a loop within an existing loop block, but you can't add a loop that intersects another loop.

Each new step consists of an unnamed partner action and an unidentified message. You can leave the action and message as they are and continue to build the process flow, or you can assign properties to them as you go.

**To add a step:**

**1** Click the Step tool in the Process toolbar.

Partner Agreement Manager displays insertion points (small turquoise circles) at all locations where you can add a step.



Insertion points (turquoise circles) appear at all locations where you can add a step.
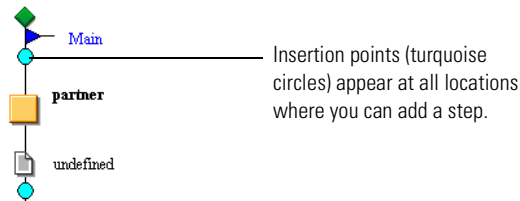
**2** Click an insertion point.

Partner Agreement Manager inserts the step at the location you clicked.

The step appears at the location you clicked.

The shaded area shows what elements are included in the step.
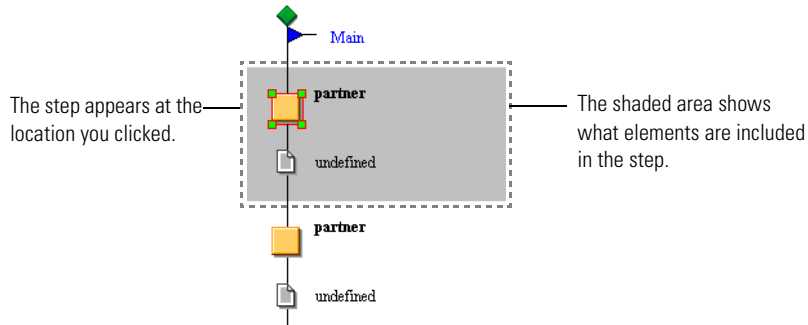
## Assigning action properties

As you build a public process, you assign each action to one of the participating partners or to a partner group. If you set up your partners before building a process, you can assign actions to them as you build the process. Otherwise, you can assign actions to placeholder partner names and replace the placeholder assignments with the real ones later. Likewise, you must set up a partner group before you can assign an action to it.

In addition, you can label the action (the label appears as a separate line under the partner or group name) and add a description that partners can see when they view the action's properties. Partner Agreement Manager uses the label to identify the steps in a process for auditing purposes.

**To assign action properties:**

Action icon

**1** Right-click the action icon and choose Properties.

You can also double-click the action icon or choose it and click the Properties button in the Command toolbar.

The Action Properties dialog box appears. The Partner tab appears on top.



Click to assign a partner who has already been set up.

Click to assign a partner group.

Click to define and assign a placeholder for the partner name.
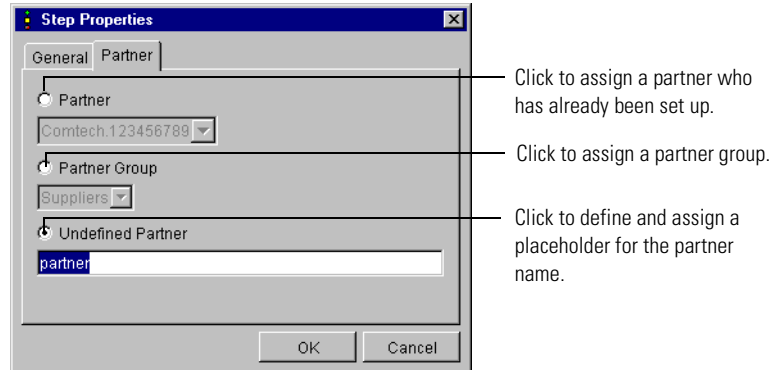
**2** Assign the action to a partner, partner group, or placeholder.

■ To assign the action to an existing partner, click Partner and choose a partner in the list.

■ To assign the action to a partner group, click Partner Group.

■ To assign the action to a placeholder partner name, click Undefined Partner and type a name.

The action icon changes to reflect the type of partner you choose.

| This icon | Represents |
|---|---|
| | The current partner. |
| | Any other partner in the process. |
| | A partner group (as opposed to a single partner). See *Setting up a partner group* on page 106 for more information. |

**3** Click the General tab to add a label and description for the action.

The General properties tab appears.



Type a label for the action.

Add a description.

**4** Type a label and an optional description, and then click OK.



The action shows the partner's profile name.

The label appears under the partner name.
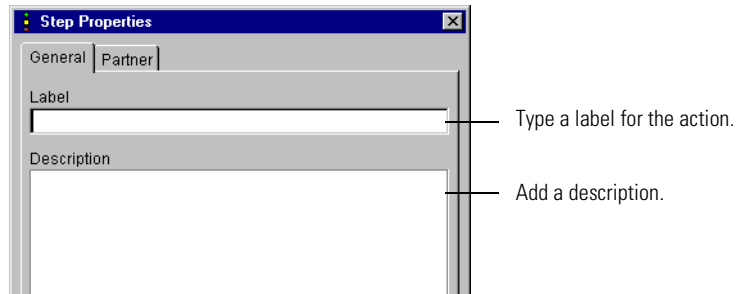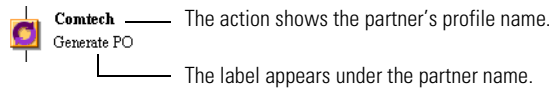
## ASSIGNING MESSAGE PROPERTIES

With the exception of the last (terminal) step in a public process path, each step in a public process transmits one or more messages. The role of a message is to convey business object information and pass control of the process to the next partner in the flow.

For each message in a public process, you can choose the business object type that's included and the way in which it's transmitted. The business object type can be one that has already been defined, or it can be a placeholder that you replace later with the actual business object type.

Transmission choices are synchronous or asynchronous. Partner Agreement Manager supports dialup and Internet for synchronous communications, as well as MQSeries for asynchronous communications.

Whether you choose synchronous or asynchronous services depends on several factors.

■ What's installed? All partners must have IBM MQSeries installed in order to use the Partner Agreement Manager MQSeries communication asynchronous services. If you choose a service type for which no services are installed, the process will fail.

- Are all partners up and running? If you use synchronous services (Internet or dialup), your partners must be up and running when you send a message in order to receive it. Because of the nature of asynchronous service, you can still send a message successfully even if a partner is not up and running when you send it.

- How quickly do you need feedback? Synchronous services provide almost instant feedback because they time out within a very short period of time. On the other hand, asynchronous services do not provide instant feedback because of the relatively longer time-out setting.

**NOTE:** You can also use a message to terminate a process or a path in a process (the setting is None).

If you have already defined your business object types, you can assign one to the process step. Otherwise, you can assign a placeholder business object type name and replace it later when your business object types have been created.

**To assign message properties:**

Message icon

**1** Right-click the Message icon in the Process window and choose Properties.

You can also double-click the Message icon or choose it and click the Properties button in the Command toolbar.

The Message Properties dialog box appears.
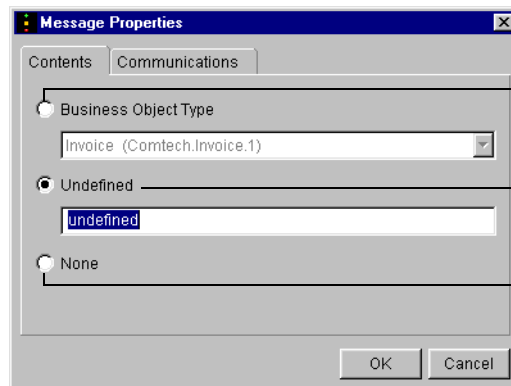
Click to assign an existing business object type.

Click to assign a placeholder business object type name.

Click to terminate the process.

**2** Select the contents for the message.

| To | Do this |
|---|---|
| Assign an existing business object type | Click Business Object Type and choose one in the list. |
| Assign a placeholder business object name | Click Undefined and type a placeholder business object name. |
| Terminate a process (or a path) | Click None. |

**3** Click the Communications tab.

The Communications tab appears.



Select a service type.
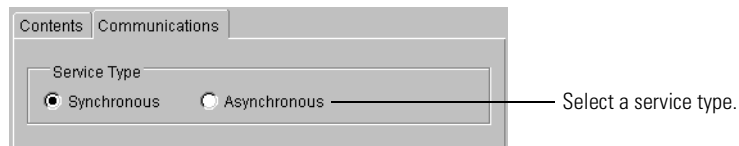
**4** Select Synchronous (for dialup or Internet) or Asynchronous (for MQSeries) service type.

Partner Agreement Manager rolls over between the different installed services of a single type if necessary. Partner Agreement Manager does not roll over between synchronous and asynchronous service types.

**5** Click OK.

## ADDING A BRANCH BLOCK

With Partner Agreement Manager, you can insert branch blocks at the beginning of a process, after a simple step, or after a message in a branch or a loop body path. You can't add a branch block to the looping path of a loop block.

Each new branch block consists of an unnamed partner action and two alternate paths—each with its own message. The branch paths merge, and the branch block concludes with a final partner and message. You define the branching conditions in a script action at the start of the branch block.

The branch block starts with a partner action.
Conditional branch logic can be AND or XOR.
Path 1 label

Path 1 business object

Join logic shows when the paths merge.

Based on the private process associated with this action, the path splits into two branches.
Path 2 label

Path 2 business object

After merging, the path leads to the next partner action in the process.
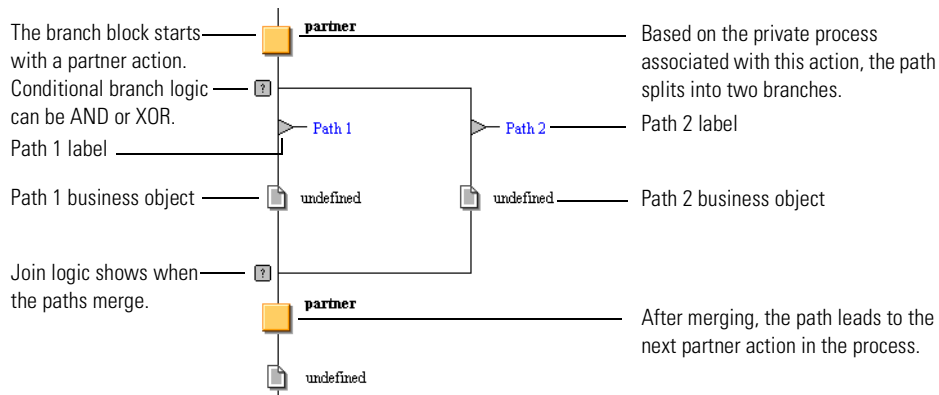
After you insert a branch block, you can assign partners to the actions and contents to the messages just as you do for a step. You can also label the paths and set the branch logic.

**TIP:** Use the branch tool to add additional branch paths. See *Adding a branch* on page 120.

### To add a branch block:

**1** Click the Branch Block tool in the Process toolbar.

Partner Agreement Manager displays insertion points (small turquoise circles) at all locations where you can add a branch block.

Insertion points (turquoise circles) appear at all locations where you can add a branch block.
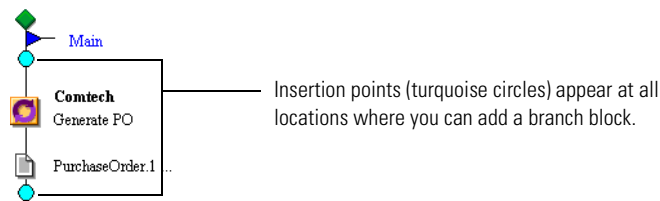
**2** Click an insertion point.

Partner Agreement Manager inserts the branch at the location you clicked.

The branch block
appears at the location
you clicked.

Double-click the path marker
to change its label. Path names must be
unique within a public process.
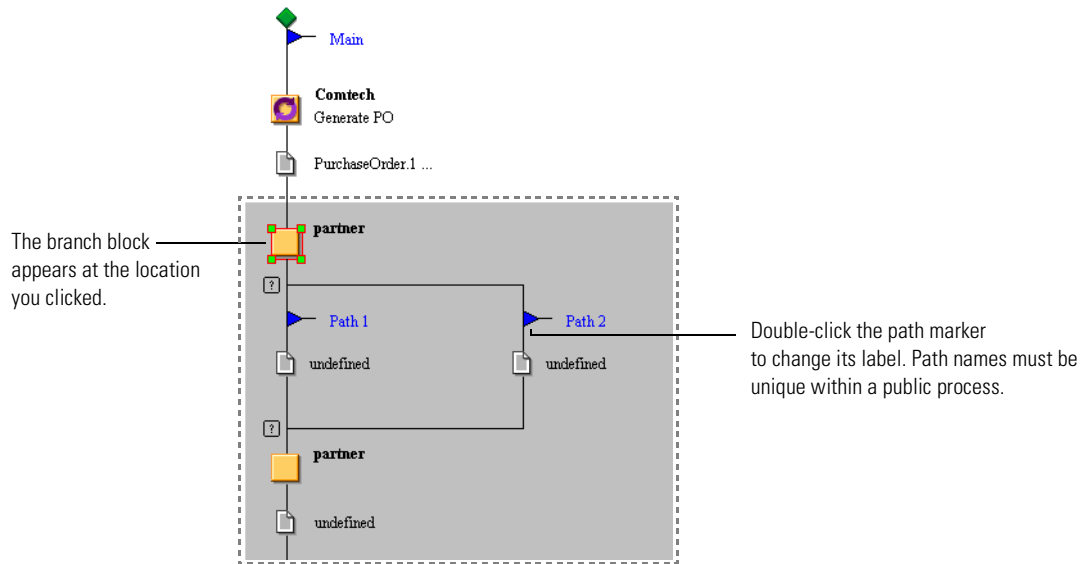
## Labeling a path

When you add a branch to a process, its paths are labeled Path 1 and Path 2.
You can change the path labels to better reflect the choices in the branch. For
example, if the branch deals with whether specifications are clarified, you
might label one path Clarified, and the other path Not Clarified.

**NOTE:** Path names must be unique within a public process.

**To label a path:**
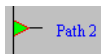
**1** Double-click a path marker.

You can also right-click the path marker and choose Properties or select the
path marker and click the Properties button in the Command toolbar.

The Label Properties dialog box appears.



Type a label for the path.
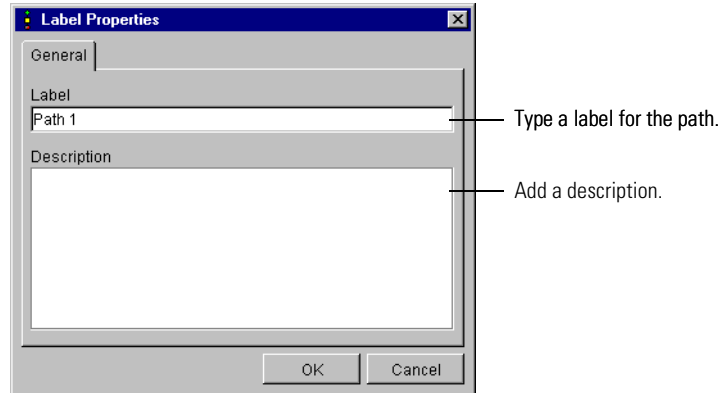
Add a description.

**2** Type a label and an optional description, and then click OK.

## Setting branch logic

The logic for a branch block determines if its paths represent alternate courses of action (the logic is XOR) or actions that need to occur at the same time (the logic is AND).

If the branch logic is set to AND, all branches must be completed before continuing to the next step. If the branch logic is XOR, only one of the paths must be completed before the process can continue.

Small logic markers at the beginning and end of a branch block show the conditional logic setting.



In the conditional logic marker, **&** means AND and **?** means XOR.

In the join logic marker **&** means AND and **?** means XOR.
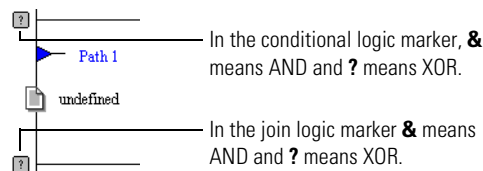
**To set branch logic:**

**1** Right-click a logic marker and choose Properties.

You can also double-click a logic marker or select a logic marker and click the Properties button in the Command toolbar.

The Branch Properties dialog box appears.



Select a branch logic.
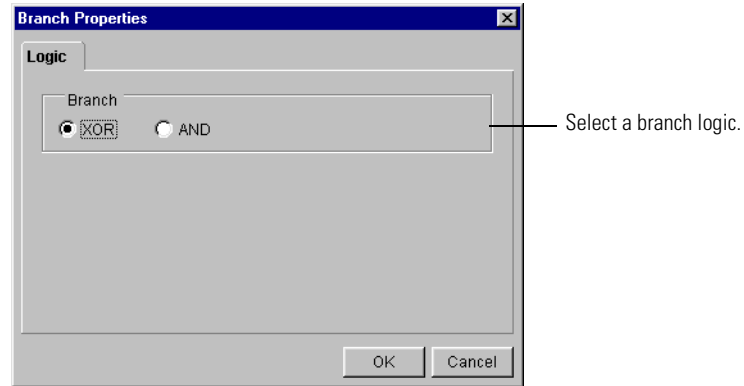
**2** Set the Branch logic to AND or XOR.

If the logic is AND, all paths must be completed. If the logic is XOR, only one path will be completed.

## ADDING A LOOP BLOCK

With Partner Agreement Manager, you can insert loop blocks at the beginning of a process, after a simple step, or after a message in a branch or a loop body path. You can't add a loop block to the loop path of another loop block.

Each new loop block consists of an unnamed partner action and two alternate paths—each with its own message. You define the looping conditions in a script action at the start of the loop block.



The loop block starts with a partner action.

The Loop Body path is the main path through the loop.

The Loop path is used if the loop condition is met.

This business object is generated if the loop condition is met.

The underlying private process establishes a condition that is either met or not met.

This business object is generated if the loop condition is not met.
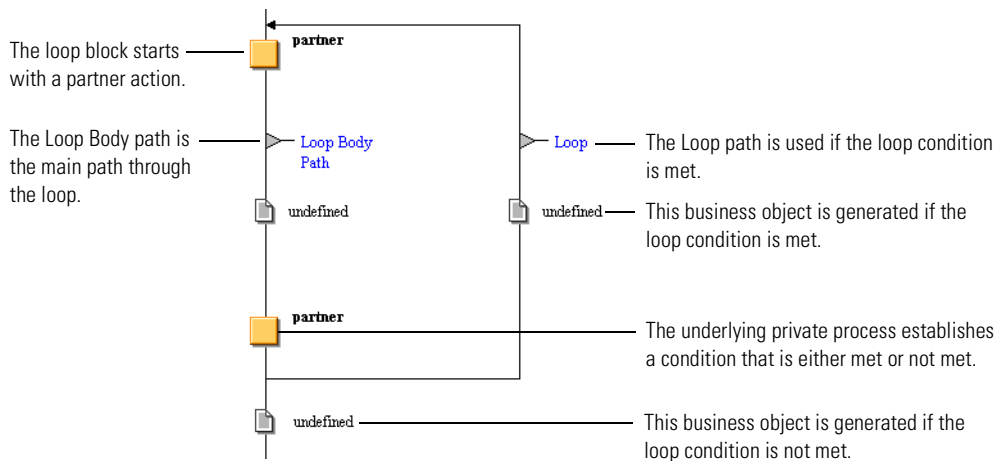
After you insert a loop block, you can assign partners to the actions and contents to the messages just as you do for a step. You can also label the loops.

**To add a loop block:**

**1** Click the Loop Block tool in the Process toolbar.

Partner Agreement Manager displays insertion points (small turquoise circles) at all locations where you can add a loop block.



Insertion points (turquoise circles) appear at all locations where you can add a loop block.
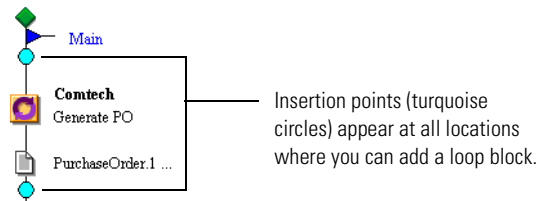
**2** Click an insertion point.

Partner Agreement Manager inserts a loop block at the location you clicked.



The loop block appears at the location you clicked.

You can assign partners to the actions and contents to the business objects. You can also change the loop labels.
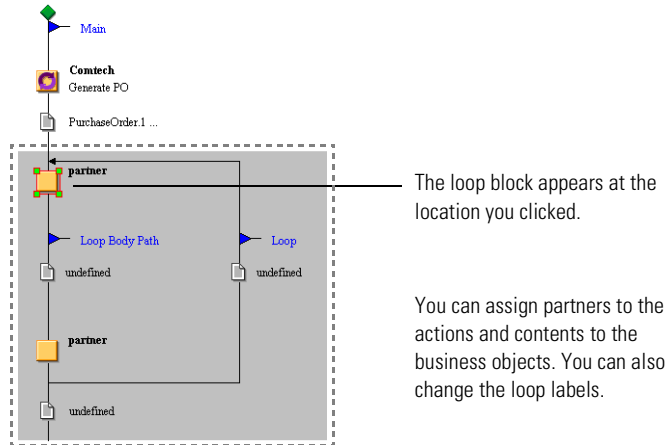
# ADDING A BRANCH

If your process includes an action that can have more than two alternate or concurrent courses of action, you can add extra branches to an existing branch block. You can also add branches around existing steps or sequences of steps, or around or within a loop block.

No matter where you place it, a branch always consists of an undefined message and a path. Partner Agreement Manager labels the path with the next available number (Path 3 or Path 4, for example). If you add a branch to an existing branch block, Partner Agreement Manager places the branch at the location you choose.



When you add an extra branch to an existing branch block, it appears wherever you place it.
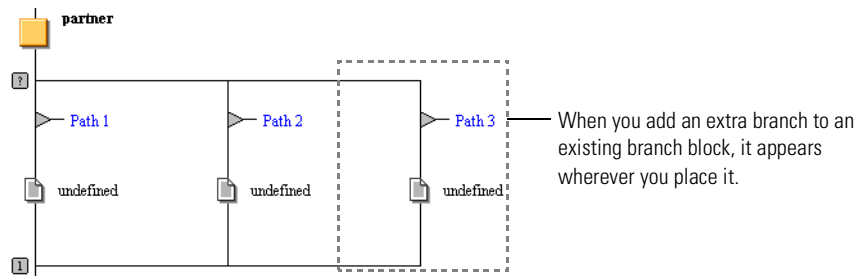
If you add a branch elsewhere in a process, you choose both the starting and ending points for the branch. Partner Agreement Manager inserts a branch with a path label and message. The path that contains the new branch's start and end points also receives a label and logic markers that control the behavior of the new branch.



Adding a branch also adds logic markers and a path label on the existing paths.

When you add a branch elsewhere in a process, it runs from the starting point to the ending point you choose.
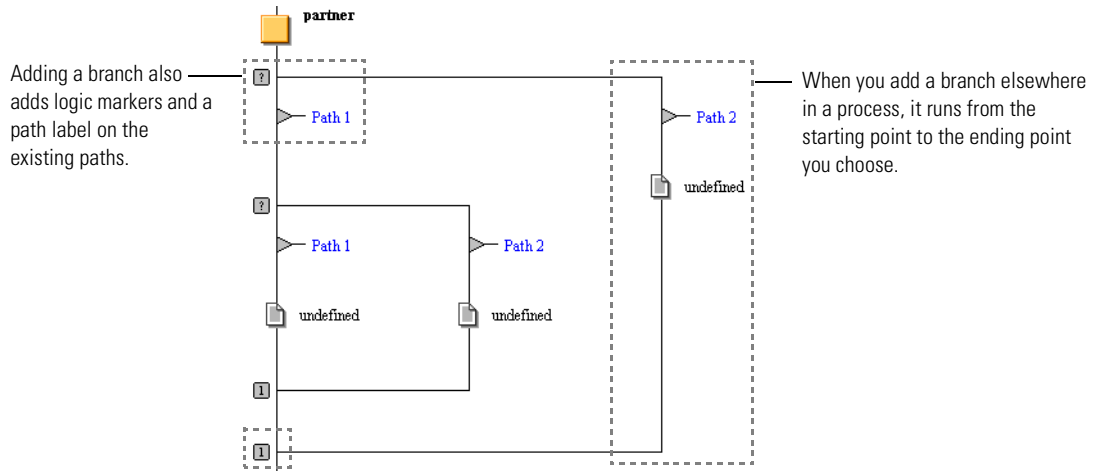
After you insert a branch, you can label the paths and set the branch logic just as you do for a branch block. See *Adding a branch block* on page 114.

**To add an extra branch:**

**1** Click the Branch tool in the Process toolbar.

Partner Agreement Manager displays insertion points (small green circles) at all locations where you can start a branch.

Click here to add a branch that starts here.

Click here to add a branch to this path.

Click any of these insertion points to add another path to the existing branch.

Click here to add a branch to this path.
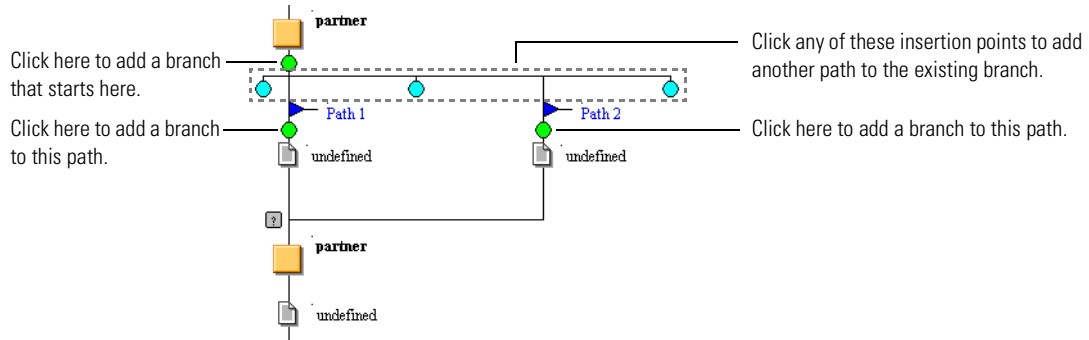
**2** Click an insertion point where you want the branch to start.

If you're adding a path to an existing branch block, Partner Agreement Manager inserts the extra path.

If you're adding a branch elsewhere in a process, Partner Agreement Manager displays end points (red circles) at all logical spots where the branch can end.

If you choose this as the start point for a branch...

...this is the logical end point.
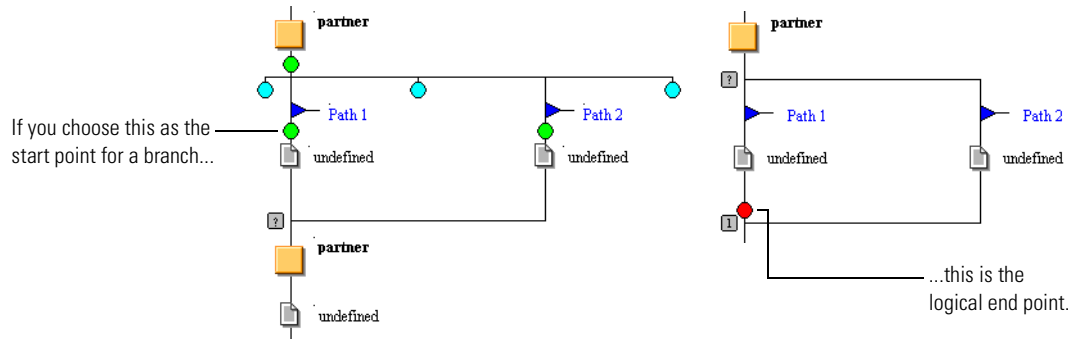
**3** Click an end point to terminate the branch.

Partner Agreement Manager inserts the extra branch.



In this example, both branches share a common partner action.

The branch appears at the location you select.
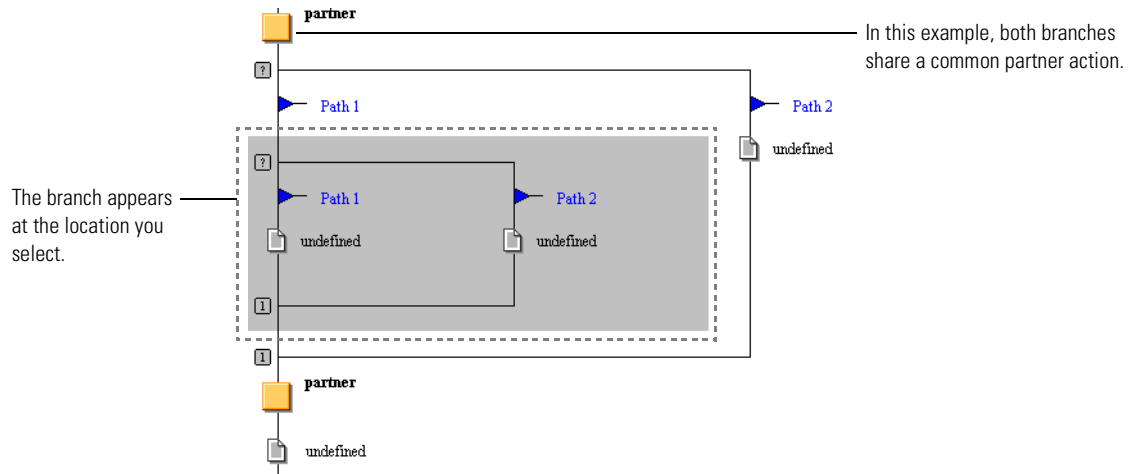
# Adding a loop

If your process includes one or more steps that might need to be repeated, you can add a loop around existing steps in the process. With Partner Agreement Manager, you can add a loop around existing steps, within an existing branch block, or within an existing loop block. A loop always begins after a partner action and loops up to the start of the same or a previous partner action.

When you add a loop, you choose both its starting and ending points.

The loop ends before an action.

Adding a loop also adds a path label on the existing path.

When you add a loop, it starts and ends at the locations you specify.

Loop Body Path

Loop

Path 1

Path 2

The loop starts after an action.
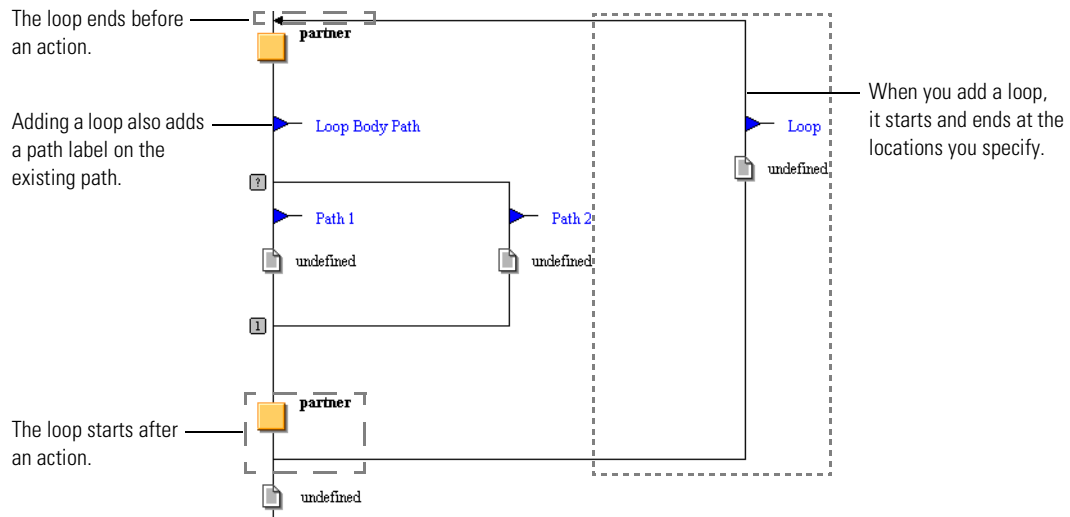
After you insert a loop, you can label the paths just as you do for a loop block. See *Adding a loop block* on page 118.

**To add a loop:**

**1** Click the Loop tool in the Process toolbar.

Partner Agreement Manager displays insertion points (small green circles) at all locations where you can add a loop.

partner

undefined

partner

Path 1

Path 2

undefined

undefined

partner

Click here to start a loop.

undefined
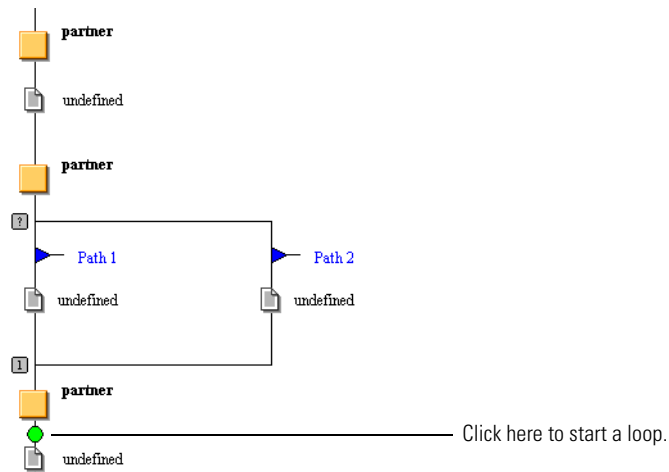
**2** Click an insertion point to choose a starting point for the loop.

Partner Agreement Manager displays end points (red circles) at all logical termination points for the loop. You can use any termination point (it doesn't have to be the one closest to the starting point).



Click here to end a loop.

Click here to end a loop.

This is the starting point for this example.
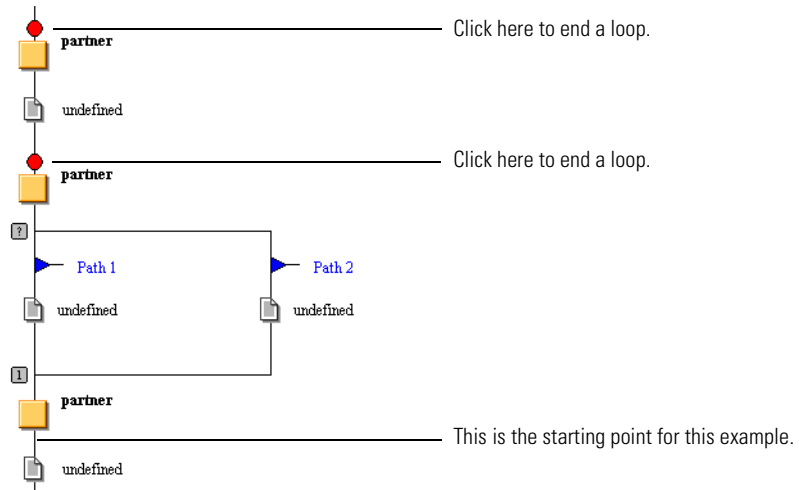
**3** Click an end point to choose a termination point for the loop.

Partner Agreement Manager inserts the loop at the locations you chose.



Adding a loop also adds a path label on the existing path.

Partner Agreement Manager inserts the loop starting and ending at the locations you specify.
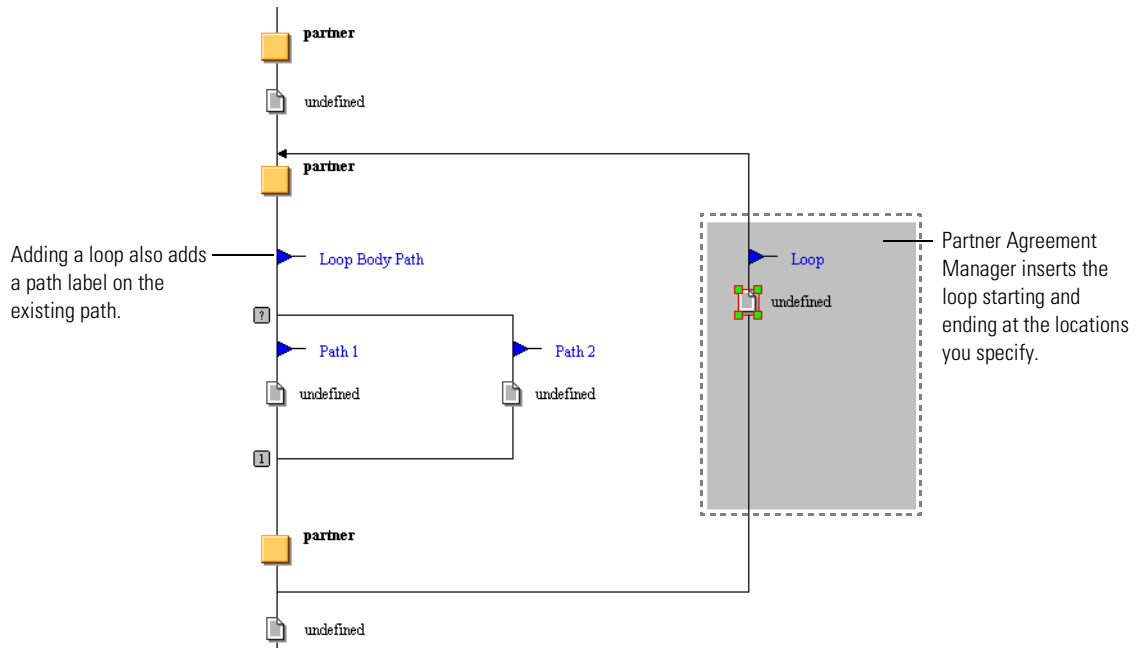
# Deleting a component

You can select and remove components from a process. You can delete entire blocks (steps, branch blocks, or loop blocks) as well as individual branch paths or loop paths. You can't delete a message from a branch or loop without removing its path segment as well.

When you delete part of a block (just one branch, for example), Partner Agreement Manager leaves the remaining elements as they were.

**To delete a component:**

1 Choose the element you want to delete.

| To delete | Choose |
| --- | --- |
| A step | The step's partner action or message. |
| A branch block | A partner action in the block or its final message. |
| A loop block | A partner action in the block or its final message. |
| A branch | The path marker.  |
| A loop | The loop marker.  |

When you choose an element, Partner Agreement Manager shades the area that will be deleted.



The shaded area shows the section that will be deleted.
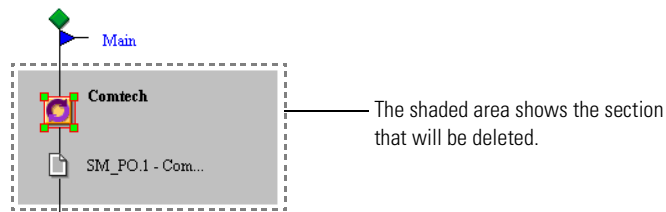
2 Click the Delete button in the Command toolbar or choose Delete from the Edit menu.

The Delete button is active only when you choose an element that can be deleted.

# Saving a public process

As you develop a public process, it's a good idea to save your work periodically. The first time you save a new process, you give it a name. Partner Agreement Manager creates a folder with the same name in your Process folder in the Process Manager window. Each subsequent time you save a draft process, Partner Agreement Manager increments the last digit of the version number. You can also save a process using a different name, or save all open private process windows associated with a public process.

If a process has already been distributed, you can save changes to the process by creating a new version (the second digit of the version number changes). Partner Agreement Manager creates a corresponding folder with the new version number in the Process Manager window.

**To save a public process:**

1 Choose Save from the Process menu or click the Save Process button in the Command toolbar.

The Save Process As dialog box appears.
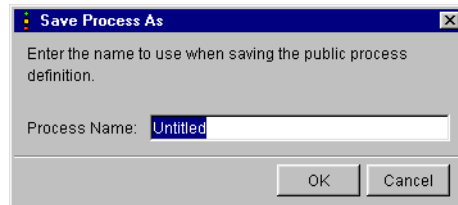


2 Type a name for the process and click OK.

Partner Agreement Manager creates a folder with the same name in your Processes folder in the Process Manager window.



The new process folder has the same name as the process. It appears in the Processes folder in the Process Manager window.

The first time you save a process, its version number is 1.0.
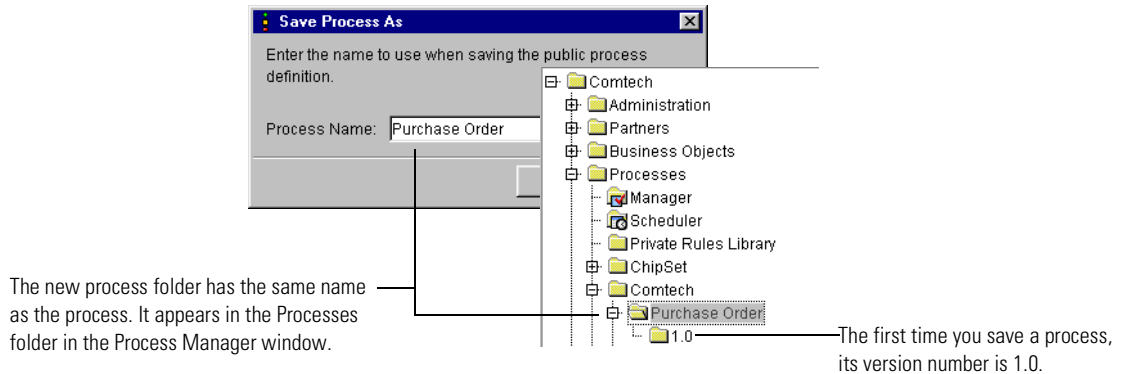
**To save a process using a different name:**

**1** Choose Save As from the Process menu.

The Save Process As dialog box appears.
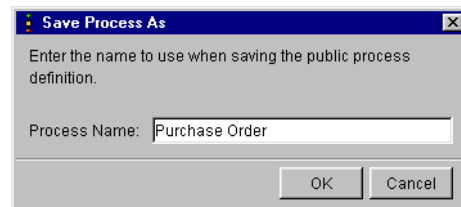


**2** Type a new name for the process and click OK.

**To save a new version of a process:**

▶ Choose Save As New Version from the Process menu.

If the process has not yet been installed, Partner Agreement Manager increments the second digit of the version number and creates a corresponding folder in the Process Manager window. If the process has already been installed, Partner Agreement Manager increments the first digit in the version number (for example, from 1.2.3 to 2.0.1)

**To save the public process and all open private processes associated with this public process:**

▶ Choose Save All from the Process menu or click the Save All button in the Command toolbar.

# Verifying a process

With Partner Agreement Manager, you can check a process to make sure that there are no loose ends or errors in logic that prevent the process from running. Typical errors include undefined messages, no termination for a process, or undefined partners.

After you verify the process, you can keep the Verify Public Process window open as you continue to work on the process.

**To verify a process:**

1  Click the Verify Process button in the Command toolbar.

    You can also choose Verify from the Tools menu.

    The Verify Public Process window appears.

The total number of errors appears here.

Icons show the type of element where the error occurs.

Individual errors are listed in the order in which they occur.

Click an entry to highlight its action or message.

Click to recheck the process.
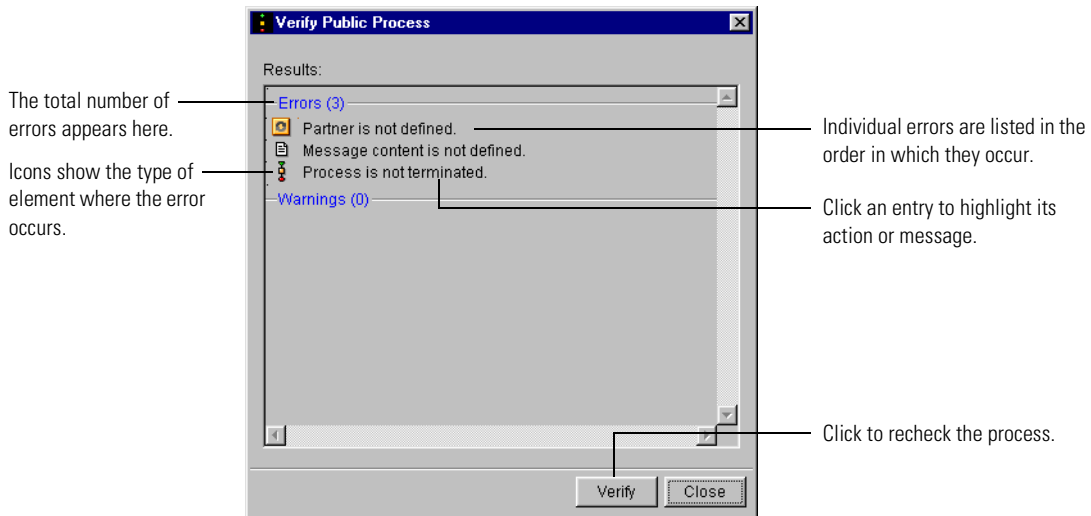
2  To see which action or message is responsible for a specific error, click the error in the Verify Public Process window.

    Partner Agreement Manager highlights the offending action or message in the Process window.

3  To recheck the process after making changes, click Verify.

    As Partner Agreement Manager checks the process again, it removes any errors that have been resolved and adds any new ones that have occurred.

4  Click Close to close the Verify Public Process window.

# REPLACING PLACEHOLDERS IN A PROCESS

As you design a process, you might find it easier to use placeholders for the names of partners or messages. For example, you might want to start designing processes while you wait for partners to install and configure Partner Agreement Manager. Or you might want to create business object types after you do a first pass on the public process.

With the Partner Agreement Manager replacement feature, you can replace partner and business object type placeholders with the correct information. You can update one partner or message at a time, or you can update them all at once.

**To replace placeholders in a process:**

**1** Open the process to update.

- To update the entire process, choose Replace from the Edit menu.
- To update a single partner or message (or the entire process), right-click the item and choose Replace from the menu that appears.

The Replace dialog box appears.



Select the item to be replaced.

Select the replacement item. The available choices depend on the item you select above for replacement.

Select the replacement scope.
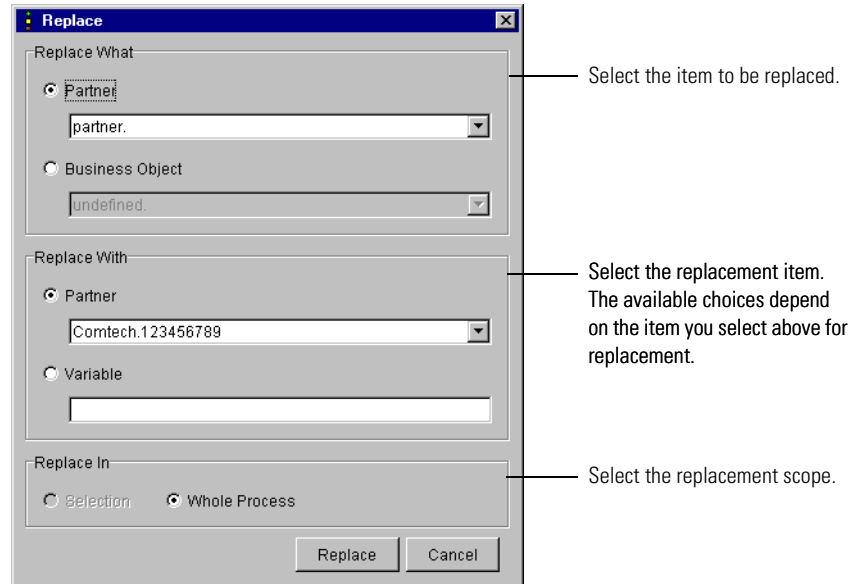
2  Select the item to replace.

- To update a partner, click Partner in the Replace What area and choose the partner.
- To update a business object type, click Business Object in the Replace What area and choose the business object type.

3  Select the replacement item in the Replace With area.

You can replace a business object type with either another business object type or a variable. See *Using variables in a public process* on page 134.

4  Select whether to replace a single item (Selection) or all items in the process with the same name (Entire Process).

5  Click Replace.

6  Click Cancel to close the Replace dialog box.

## Importing and exporting processes

With Partner Agreement Manager's process export and import feature, you can transfer process definitions between Partner Agreement Manager installations. This feature is most useful when you're building a process at one location and intend to run it from another. For example, you might design a public process during a Partner Agreement Manager training session and want to move it to your own Partner Agreement Manager installation. You can also use import and export to move a process back and forth between installations of Partner Agreement Manager running on a desktop computer and laptop so that you can continue working during business trips.

---

**Important:**  Importing and exporting is a method for transferring processes from one computer to another, not from one partner to another. It is not a replacement for Partner Agreement Manager's standard distribution feature that sends processes to partners for implementation.

---

### Exporting processes

Exporting a process creates an XML file that contains any combination of information from the process: the public process itself, its private processes, its business object types, and its maps.

**To export processes:**

1 Click the Import/Export Manager button in the Command toolbar in the Process Manager window.

The Import/Export Manager appears.

Click to choose a process to be exported.

This area shows a list of the items in a pending import or export action.
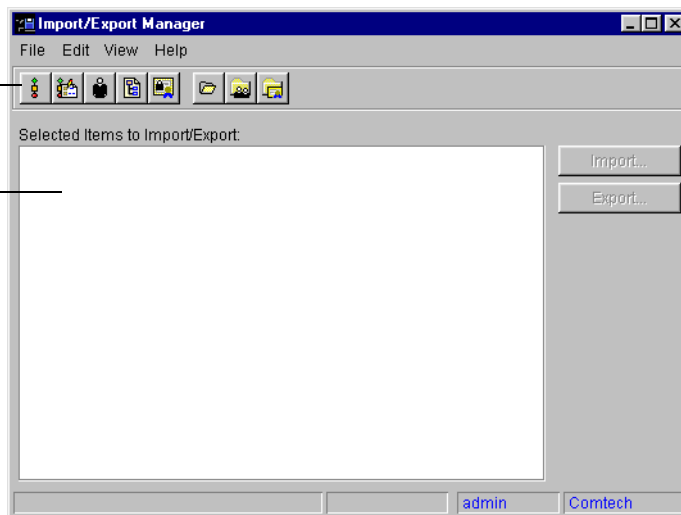
2 Click the Select Process to Export button in the Command toolbar, or choose Select for Export from the File menu, and then choose Process from the menu that appears.

The Public Processes dialog box appears. It lists all available public processes.
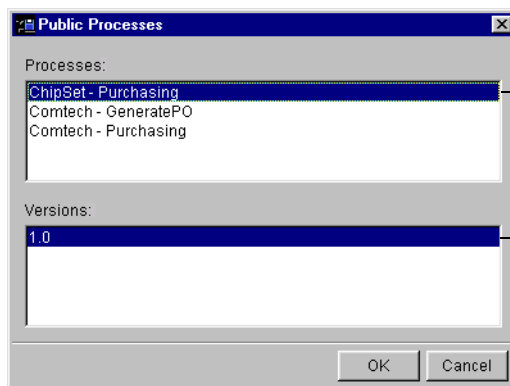
When you choose a process in this list...

...this list shows the available versions.

3 Choose a process and version, and then click OK.

The Import/Export Manager lists the public process, along with its business object types, maps, and private processes.



This list shows all available components for the process you choose.
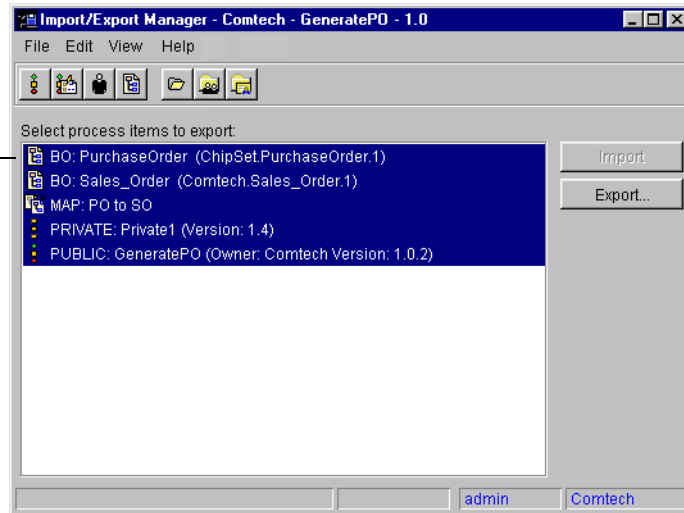
**4** Choose the items that you want to export and click Export.

You can also choose Export from the File menu. The Export to File dialog box appears.

**5** Type a name and choose the location where you want to save the export file.

Partner Agreement Manager exports the items you chose and alerts you when the export is complete.

**6** Close the Import/Export Manager.

## IMPORTING PROCESSES

When you import a process, you can choose any exported XML file, regardless of who in your organization created it. You can also choose which components you import. For example, the import file might contain all business objects and private processes for a public process, but you can choose to import only one of the private processes.

**To import processes:**

**1** Click the Import/Export Manager button in the Command toolbar in the Process Manager window

The Import/Export Manager appears.

**2** Choose Open for Import from the File menu, and then choose XML File.

The Open Exported File dialog box appears. You can choose any export (XML) file.

**3** Choose the file you want to import and click Open.

The Import/Export Manager shows all available components in the file.

This list shows all available components you can import.
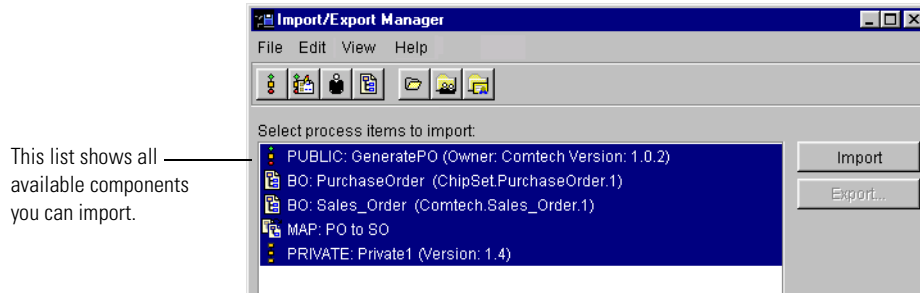


**4** Select the components to import, and choose Import from the File menu or click Import.

Partner Agreement Manager imports the items you selected and alerts you when the import is complete.

**5** Close the Import/Export Manager.

## Using variables in a public process

Partner Agreement Manager uses variables to give you maximum flexibility in designing public and private processes. You can think of variables as temporary storage for some of the information that moves within a process. Partner Agreement Manager uses two types of variables: variants and business object variables.

Variants are the equivalent to single field variables: they store text strings—the type of information that might be contained in a single field element. You can use variant variables to store the input for actions, to set flags (such as the time-out flag for an approval action), to move information within scripts, or to store the results of an approval action.

Business object variables create an instance of the business object type within the process so that you can use it to move information. You use business object variables to store the output from extension application actions, for the inputs for map actions, or for the inputs and outputs for subprocesses.

You can define variables from either the Process window or the Private Process window. If you define a variable in the Process window, it's available to all of your private processes associated with the public process (but not to your partners' private processes). If you define a variable from a Private Process window, it's available for that private process only.

Each time you run a process, Partner Agreement Manager inserts any current values into the variables you've defined. It then uses those values to continue with a private process and to pass its results back to the public process.

## Displaying the variables panel

Partner Agreement Manager can display variables in a special panel at the bottom of the Process and Private Process windows. You can display the variables as you work with them, and then hide them when you no longer need them in view.

**To display the variables panel:**

▶ Choose Variables from the View menu.

The variables panel appears.

Variables panel in the
Public Process window.

| ▼ Name | Type | Usage | Description |
| --- | --- | --- | --- |
| PONumber | Variant | Process | |
| PO | Purchase_Order (Comt... | Process | |

Variables panel in the
Private Process window.

| Name | Type | Usage | Description |
| --- | --- | --- | --- |
| ActiveInputSet | Variant | | Indicates which input set is active. |
| SalesOrder | Sales_Order (Comtech.... | Process | |
| PO | Purchase_Order (Comt... | Process | |

Variant variable —

Business object —
variable

The columns in the variables panel show the variable's name and type, the business object it's based on, how it can be used, and a description.

## CREATING A VARIABLE

You can create variables in both the Process window and the Private Process window. If you define variables in the Process window, you can also specify how to use them: for input, output, or general processing.

Partner Agreement Manager expects input variables to be provided to the process when it starts running. A process can be started by the scheduler, by an enterprise system event, or by a subprocess call from another process.

Partner Agreement Manager uses output variables to store the outcome of a subprocess and return it to the process that called it.

**NOTE:** Variables designated as input or output can also be used for general processing, but variables designated as general can't be used for input or output.

After you create a variable you can name it, select its type, choose how it is to be used, and type a description.

**To create a variable:**

**1** Choose New Variant or New Business Object from the Variables menu.

You can also right-click anywhere in the variables panel and choose New Variant or New Business Object. Partner Agreement Manager adds the new variable to the bottom of the variables panel.

The New Variant Variable or New Business Object Variable dialog box appears.

Type a name. ────────

For business object ──────── variables, choose a business object to base the variable on.

Type a description. ────────
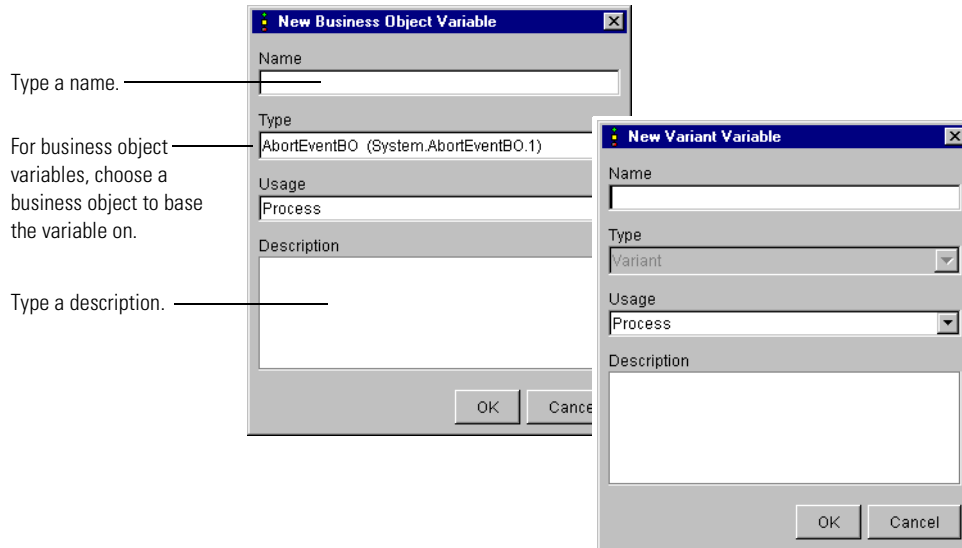
**2** Type a name for the variable.

**3** If it's a business object variable, choose a business object type.

The type you choose determines the business object definition to use.

**4** If you created the variable in the Process window, select how the variable is to be used.

**5** Type an optional description.

**6** Click OK.

## DELETING A VARIABLE

If you no longer need a variable, you can delete it.

**To delete a variable:**

▶ Choose the variable in the variables panel, right-click, and choose Delete.

You can also select the variable and choose Delete from the Variables menu.

**To delete all variables:**

▶ Right-click in the variables panel, and choose Clear All.

You can also choose Clear All from the Variables menu.

# 7

# DEFINING PRIVATE PROCESSES

Read this chapter for information about how to define the underlying private processes that implement the steps in a public process.

This chapter includes these sections:

# About private processes

Each step in a public process has an associated private process that determines the actions that each partner takes. Private processes provide you with tremendous flexibility to perform the actions needed to integrate with your internal systems and procedures.

A private process begins with input from the public process and ends with output that returns to the public process. The input can be the receipt of a business object from a partner, or it can be a triggering event from an internal system, such as inventory control. The output is the business object that transfers control of the process to the next step.



The input from the public process can be a business object or an event.

Each type of action has a different set of properties and is represented by its own icon.

The output from the private process goes back to the public process.
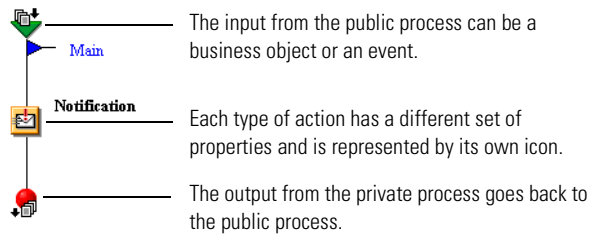
Like the public process itself, the private process that you associate with each public process step can vary in complexity according to your business needs. Private process actions can be as simple as generating an e-mail notice.

Or they can be more complex, including any combination of scripts written in VBScript or JavaScript, interactions with your other systems or application programs, other Partner Agreement Manager processes, or requests for approval. A private process can also include a timing factor that determines how long to wait before progressing to the next action.

**NOTE:** VBScript is supported only on Windows NT. JavaScript is supported on both Windows NT and UNIX.

You might think of a private process as the intelligence behind the public process. For example, the private process determines what happens when a public process branches or loops.

An action that branches in the public process...

...uses a script in the private process...



```
Script
sub main()
    Dim isValid

    isValid = ConfigResultBO.getData("valid_flag")

    if isValid = "true" then
        setPath "Yes"
    else
        setPath "No"
    end if
end sub
```

...to define the two branches.
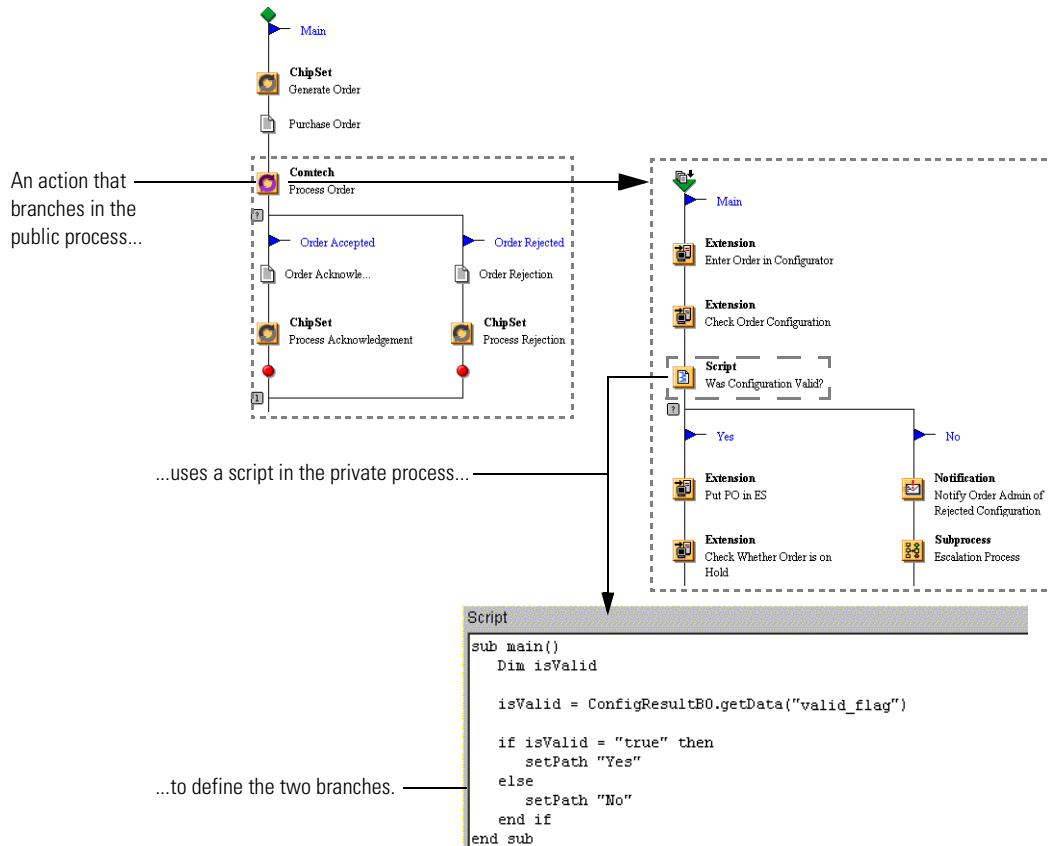
Because a private process manages the interaction between Partner Agreement Manager and any external applications, it might receive an error notification if a Partner Agreement Manager adapter fails. It is good process design to include actions that handle errors without sending them to the public process, if possible.

For example, if you have an extension action that calls an adapter and the adapter fails, you can use a notification action to send an e-mail message to the appropriate person in your company, giving as much information about the failure as possible. The private process can wait for an approval action identifying the problem as resolved, and then loop the process back to retry the extension action. If the approval action fails or times out, you can escalate the problem to another person for help resolving it.

**NOTE:** The best error-handling practice for dealing with external applications is to resolve adapter errors in the adapter itself. See the *Partner Agreement Manager Adapter Developer's Guide* for more information about error handling in adapters.

# ABOUT DEFINING A PRIVATE PROCESS

There are two steps involved in defining a private process. Like a public process, a private process is concerned with the flow of actions, so the first step is to diagram which actions happen and in what order. Next, you define the exact nature of each action.

For example, if you include a notification, you need to specify who is notified and what the message is. If you include an approval action, you need to choose the recipient, enter instructions for the recipient, enter the questions the recipient answers, and set a time-out that puts a deadline on the approval.

In addition to its specific settings, each type of action (with the exception of output object actions and termination actions) lets you include a script written in VBScript or JavaScript.

**TIP:** If you want to reuse an existing private process, you can insert it into the Private Process window. See *Inserting a private process* on page 197.

# About the Private Process window

You use the Private Process window to diagram the actions associated with a single step in a public process. The Private Process window includes two toolbars that give you access to the most frequently used commands and private process actions.

Toolbar buttons give you quick access to common Private Process window commands.

Action tools help you add actions to a process.

Builder tools help you diagram the flow of actions.

The status bar shows information about the private process.
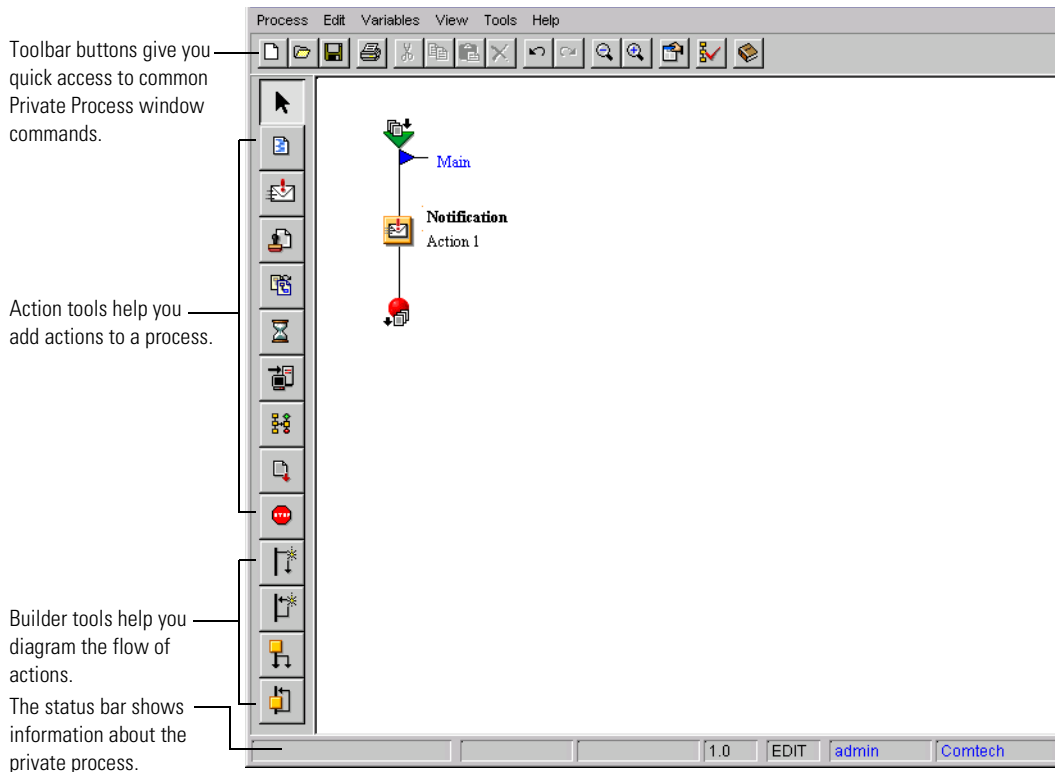


## About the Command toolbar

The Command toolbar at the top of the Private Process window gives you access to common File and Edit menu commands, lets you zoom, set properties, verify the process, and display the Business Object Catalog.

Save Process

Print

Zoom In, Zoom Out

Verify Private Process

New Process



Business Object Catalog

Open Process

Cut, Copy, Paste, Delete

Undo, Redo

Properties

Partner Agreement Manager shows the Command toolbar whenever you open a Private Process window. You can hide the toolbar if don't need it.

### To hide the Command toolbar:

▶ Choose Toolbar from the View menu to hide it.

▶ Choose Toolbar from the View menu a second time to display it.

### ABOUT THE PRIVATE PROCESS TOOLBAR

With the toolbar on the left side of the Private Process window, you can add actions and building blocks to the private process. Partner Agreement Manager normally displays the Private Process toolbar whenever you work with a private process.



Selection tool

Script action

Notification action

Approval action

Mapping action

Timer action

Extension action

Subprocess action

Output object action

Termination action

With action tools, you can add actions to a private process.

Branch tool

Loop tool

Branch Block tool

Loop Block tool

Use builder tools to add branches and loops to the flow of actions.
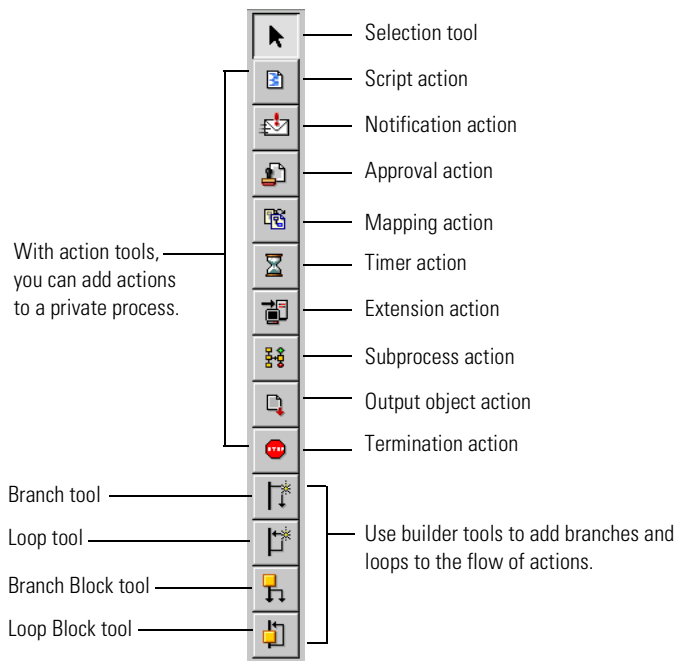
For a complete description of each tool, see *About private process actions* on page 147. For information about the building blocks, see *About private process components* on page 148.

**TIP:** The Private Process toolbar doesn't appear if a private process is activated in an installed public process. This is because Partner Agreement Manager doesn't permit you to edit an activated private process. If you want to revise the private process, you can either activate another private process in its place or open the private process and choose Save As New Version from the Process menu. You can then edit the new version.

## ABOUT THE STATUS BAR

The status bar at the bottom of the Private Process window shows information about the state of the process and the current user.



Private Process state — Active — 1.1 — READ — Admin — Comtech — Profile name
User name
Version number — Current user's access
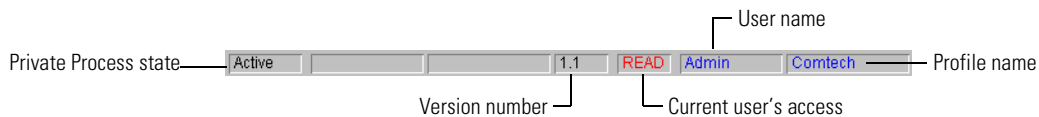
- The private process state shows whether this version of the private process is active or inactive.

- The private process version is a two-digit number. Each time you save a private process, the last digit increments. The first digit changes when you save a new version of the private process.

The first digit changes when you save a new version of the private process. — **2. 3** — The second digit changes each time you save the private process.

- Access is the current user's access to the private process. All users who have Edit access to the public process can edit any of their private processes associated with it—until the public process is installed.

- The user name is the login name for the current user.

- The profile name is usually also the company name.

Partner Agreement Manager is preset to display the status bar whenever you open a Private Process window. You can hide the status bar if you don't need it.

**To hide the status bar:**

▶ Choose Status Bar from the View menu to hide it.

▶ Choose Status Bar from the View menu a second time to display it.

**NOTE:** The Private Process window title bar shows the name of the public process that this private process is associated with.

### About the menu bar

The Private Process menu bar appears at the top of the window.

- The Process menu contains commands for opening, saving, setting properties for, or printing a private process.

- The Edit menu contains commands for undoing and redoing actions, cutting, copying, pasting, and deleting actions and blocks.

- The Variables menu contains commands for creating a new variable, or editing, deleting, or clearing all variables.

- The View menu contains commands for changing your view of the elements in the Private Process window, and for hiding or showing the toolbar, status bar, variables list, and Business Object Catalog.

- The Tools menu contains commands for verifying a private process and testing a script.

### Zooming in the Private Process window

Partner Agreement Manager provides several zoom controls that let you resize the contents of the Private Process window as you work. You might want to zoom out to see all actions, or zoom in to see the details of an action. Zoom settings are 25%, 50%, 100%, 200%, and 400%.
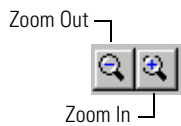
Zoom Out ⌐

Zoom In ⌐

**To zoom the contents of the Private Process window:**

▶ Click the Zoom Out or Zoom In buttons in the Command toolbar to zoom from one setting to the next.

▶ Right-click anywhere in the Private Process window and choose a zoom setting from the View menu that appears.

▶ Choose a zoom setting from the View menu (in the main menu bar).

## Printing private process information

To help you get an overview of your private process, Partner Agreement Manager lets you print the contents of the Private Process window as you work.

**To print process information:**

▶ Choose Print from the Process menu.

## About private process actions

Partner Agreement Manager provides nine types of actions that you can include in a private process:

A *script action* is a script written in VBScript or JavaScript that manipulates information or sets up conditional actions based on input. You can use a script to establish decision-making criteria for branches or loops, to set variables, or to calculate values that are used elsewhere in the private process.

For example, if a private process contains a loop or branch, you'd use a script to specify the branch to take or whether to follow the loop. You might also use a script to analyze an incoming purchase order and divide it into different business objects for each product line.

**NOTE:** You can also attach a script to any other type of action.

A *notification action* sends an e-mail, fax, or pager message to the addressees that you specify. You use a notification to let someone inside or outside of the company know that an event has occurred. For example, you might use a notification to alert the order entry department when a purchase order arrives from a customer.

An *approval action* lets you ask for a response from a user before proceeding in a private process. You might use an approval action, for example, to ask for an OK when a purchase order exceeds a predetermined amount.

A *mapping action* lets you call a map that populates the fields in a business object type based on information from another source. You might use a mapping action when you want to transform data from one business object type to another. For example, you can use a mapping action to transform a purchase order generated by your inventory system into a sales order that your partner expects.

A *timer action* inserts a pause in the private process. You use a timer action whenever you want the specified period of time to elapse before continuing with the next action. For example, you can use a timer action to insert a delay in a private process loop that polls a business system for a specific event.

An *extension action* calls any external application that's registered with Partner Agreement Manager. You might use an extension action to launch a spreadsheet application, perform some calculations, and update the enterprise system. You might also use it to get information from an enterprise system or listen for an event in the enterprise system. For example, you can use an extension action to generate a purchase order when inventory drops below a certain point.

A *subprocess action* calls an existing Partner Agreement Manager public process. You can use any existing Partner Agreement Manager process for which your company has the first partner action. For example, you might use a subprocess to get a quote approved by a third-party supplier before responding to a customer.

An *output object action* binds a business object in your private process to the expected output object and path in the public process. You might use an output object action at any point in a private process where you are ready to assign a business object to the public process (typically at the end of a path in the private process).

A *termination action* stops a process at a predetermined point for a reason that you specify. You might use it to allow for errors in data that prevent the process from completing successfully. For example, you can use a termination action to stop a process when the enterprise system passes incomplete or corrupted information.

## About private process components

The Private Process window uses four components: branch, loop, branch block, and loop block. The names and toolbar buttons are the same for both the Public Process and Private Process tools, but their actions are slightly different (a private process, unlike a public process, doesn't contain business objects at every step).

*Branches* or *branch blocks* divide a private process path into two alternate paths. You can use the paths to set up alternate courses of action or two actions that need to happen simultaneously.
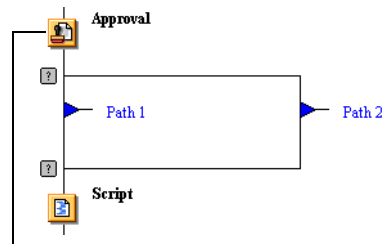


In a branch, a **script attached to the action** at the beginning of the branch establishes the branch conditions.
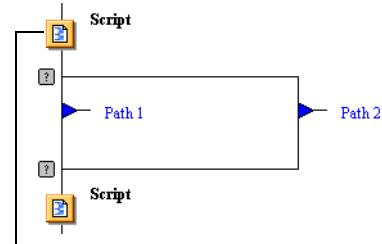
In a branch block, a **script action** at the beginning of the branch block establishes the branch conditions.

A *branch* adds a second path to the private process. You use the script tab of the action immediately before the branch to set up the branch conditions. You can add branches between any two actions in a private process. For example, you can use a branch for the different actions to take, based on the results of an approval action.

A *branch block* consists of a branch (two alternate paths) with scripts before and after it. You use the initial script to set up the branching conditions. You can add branch blocks anywhere within a private process.

*Loops* or *loop blocks* add a repeating path that loops back from the conclusion of one action to the start of the same or another action. You might use a loop to set up a path that repeats an action until a condition is met. For example, you can use a loop to repeat the approval process on a purchase order until it's complete, or to delete items from a queue until the queue is empty.
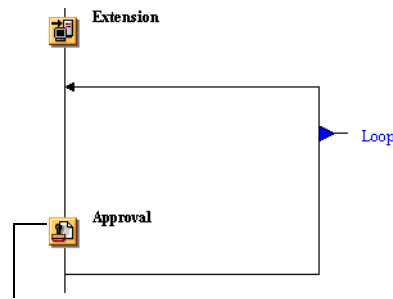


In a loop, a **script attached to the action** at the beginning of the loop establishes the looping conditions.
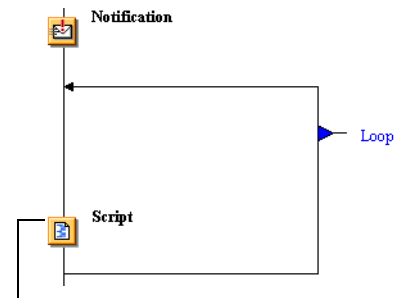
In a loop block, a **script action** immediately before the loop path establishes the looping conditions.

A *loop* adds a looping path to the private process. You use the script tab of the action immediately before the loop to set up the loop conditions.

A *loop block* consists of a loop and a script. The script action establishes the loop conditions. You might use a loop block when you want to add both a loop and a script at the same time.

# CREATING A NEW PRIVATE PROCESS

You can create a private process for any step in a public process to which you are assigned. You can also create an independent private process and store it in the Private Process Library.

**To create a new private process:**

1  Right-click a step in the public process and choose Private Process.

- To create a private process associated with a public process, you can also choose the public process step and click the Private Process button in the Command toolbar.

- To create a private process to be stored in the Private Process Library, click the New Private Process button in the Process Manager window.

The Private Process window appears.



With action tools, you can add actions to a process.

Builder tools help you diagram the flow of actions.

The input object is the starting point for the private process. Double-click to view the input set.

Double-click the path node to change its label.

The output object is the conclusion of the private process. Double-click to view the output set.
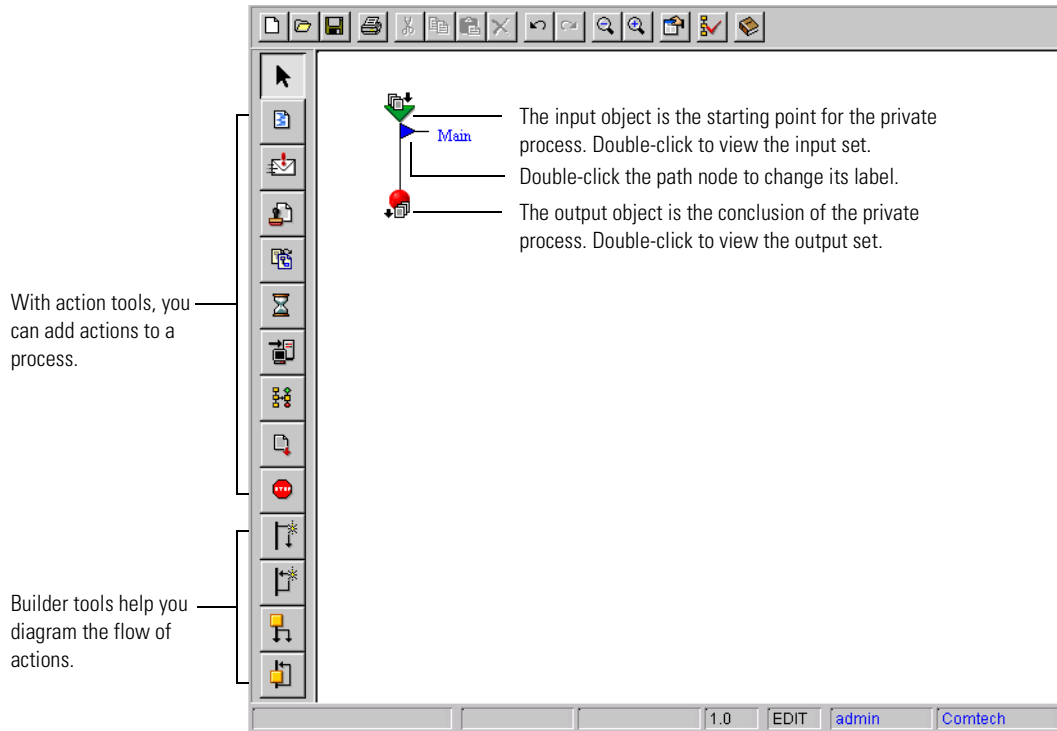
Main

**2** To relabel the main path, double-click the path node.

You can also right-click the path node and choose Properties. The Label Properties dialog box appears.



Type a new label.
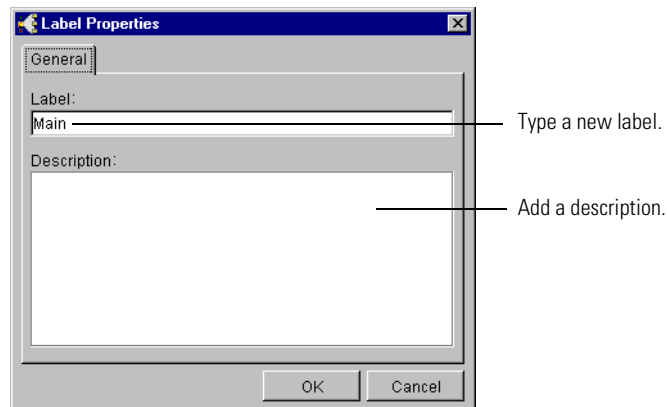
Add a description.

**3** Type a new label and an optional description, and then click OK.

# ADDING AN ACTION

You add an action to a private process the same way you add a step to a public process: by selecting a tool and then clicking a location. After you add an action, you can label it, enter a script, and set its properties.

**To add an action:**

1   Click the appropriate action tool in the Private Process toolbar.

Partner Agreement Manager displays insertion points (small turquoise circles) at all locations where you can add the action you selected.

Insertion points (turquoise circles) appear at any location where you can add the action.
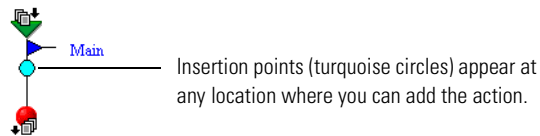
2   Click an insertion point.

Partner Agreement Manager inserts the action at the location you clicked.

The action appears at the location you clicked.

Each type of action has its own icon and label.
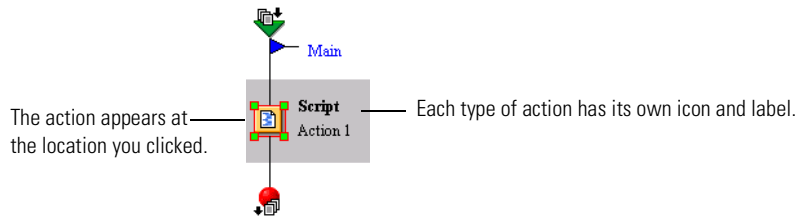
# CHANGING THE ACTION TYPE

As you design a private process, you might change your mind about the type of actions to use. If you haven't already set the action's general properties or created a script, it's easy to delete the action type and add a new one. If you have already set an action's properties, you might want to change the action to a different type. Depending on the location or the action type you start with, Partner Agreement Manager lets you switch freely between approval, extension, mapping, notification, output object, script, subprocess, termination, and timer action types—and maintain the general properties, the script properties, or both.

Output object and termination actions do not have script tabs in their properties. If you switch to either of these action types from one that has a script attached, the script will not be preserved. The only action you can change to a termination action is the last action on a path.

**To change the action type:**

1 In the Private Process window, right-click the action you want to switch and choose Change Action Type from the menu that appears.

The Select Action Type dialog box appears. The action types listed depend on the type of action and its location in the private process.

Contents of this list depend on the initial action type and location.

Turn this setting on to keep the action's general properties.

Turn this setting on to keep the action's script properties.
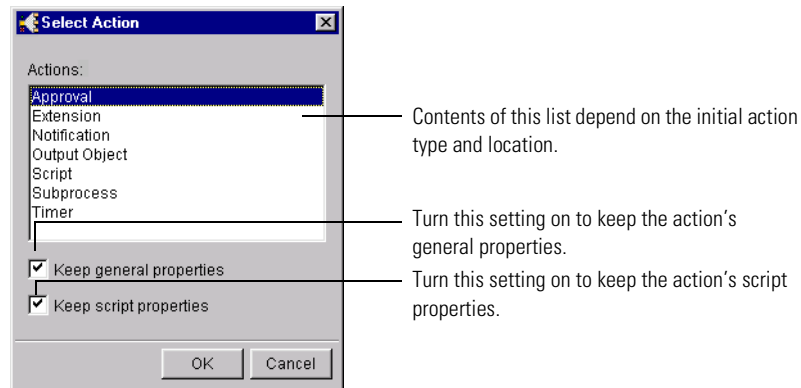
2 Choose the action type.

3 Change the settings to maintain or remove the action's current general and script settings.

4 Click OK.

## ADDING BRANCHES OR LOOPS

Adding branches or loops is as simple as selecting the appropriate tool and then clicking a location. When you add branches or loops, you specify both starting and ending points. For blocks, you choose an insertion point.

After you add a branch or loop block, you can label its paths. You can also change the logic for both branches and branch blocks.

**To add branches or loops:**

**1** Click the action tool in the Process toolbar.

If you're adding a branch or a loop, Partner Agreement Manager displays insertion points (small green circles) at all locations where you can start a branch or loop. If you're adding a branch to an existing branch, Partner Agreement Manager displays green insertion points at all locations where you can add another branch. If you're adding a branch block or a loop block, the insertion point circles are turquoise.



You can add a new branch to an existing branch block at any of these points.

You can add a new branch here.
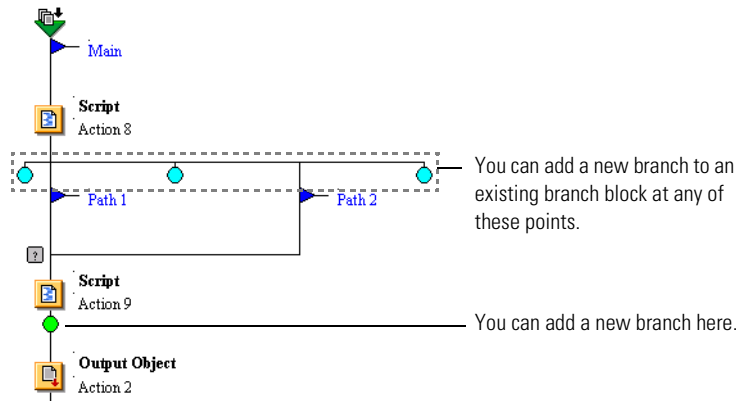
**2** Click an insertion point.

When you add a branch block or a loop block, Partner Agreement Manager inserts the block at the location you clicked.



In a branch block, double-click to change the logic.

Double-click a node to change the label.
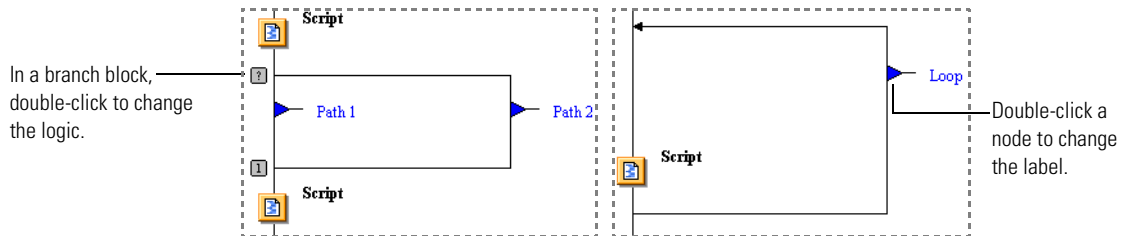
When you click a green circle to add a branch or a loop, Partner Agreement Manager displays end points (red circles) at all possible ending points.

**3** Click an end point (red circle) to specify where the branch or loop ends.

Partner Agreement Manager adds the branch or loop around existing actions.

In a branch, double-click to change the logic.



Double-click a node to change the label.
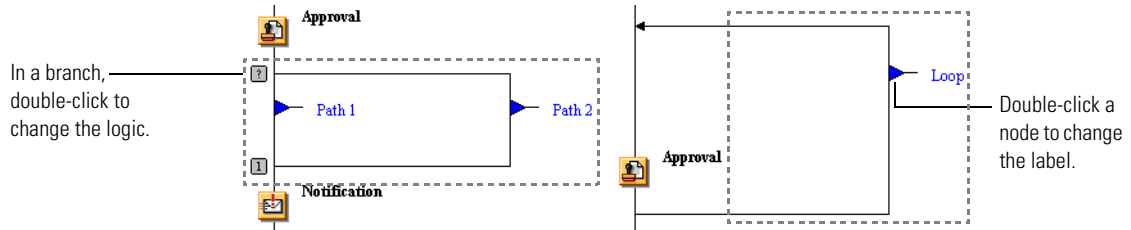
If you insert a branch or branch block, you can set its logic to AND or XOR. See *Setting branch logic* on page 117. Loops and loop blocks do not have logic settings.

## ADDING A SCRIPT

With Partner Agreement Manager, you can add a script action to a private process or attach a script to any other type of action. If a script is part of another action, it runs after the action completes.

You can use scripts for many different types of tasks. For example, you can use a script to define the alternate courses of action in a branch, define the conditions for a loop, move the value from a variable generated by an extension application to another part of the process, or map attributes from one business object to another.

You might also use a script in a notification action to set a variable, or in an approval action to reroute the approval after it times out.

Partner Agreement Manager supports scripting using Visual Basic or JavaScript, which means that you can use the full capabilities of the scripting language. In addition, you can use the extensions that Partner Agreement Manager has added and that are tailored to performing Partner Agreement Manager functions. For more information about the Partner Agreement Manager extensions to Visual Basic and JavaScript, see the *Partner Agreement Manager Script Developer's Guide*.

**IMPORTANT:** JavaScript is supported on both Windows NT and UNIX. VBScript is supported only on Windows NT.

Here's an example of a simple script that determines the path to take in a branch.

```
Script Type:
VBScript

Script:
sub main()
   Dim isValid

   isValid = ConfigResultBO.getData("valid_flag")

   if isValid = "true" then
       setPath "Yes"
   else
       setPath "No"
   end if
end sub
```

After you choose the script type, Partner Agreement Manager gives you several ways to enter a script. You can:

■ write the script directly in an action's Script tab.

■ open the Script Editor and use it to write and verify the script.

■ copy and paste all or part of another script.

■ insert an entire script that has been saved in the Script Manager.

In addition to creating and storing scripts, the Script Manager also lets you edit existing scripts or remove scripts from the list. See *Managing scripts* on page 159.

### To add a script:

**1** To add a script to an action, open the action's Properties dialog box and click the Script tab.

You can also add a script action and double-click it to open its Properties dialog box.

The Script tab appears.



Type a script...

Choose a script type.

Saves the current script in the script library.

Checks the current script.

... or click Insert Script to choose a script.

Opens the Script Manager.
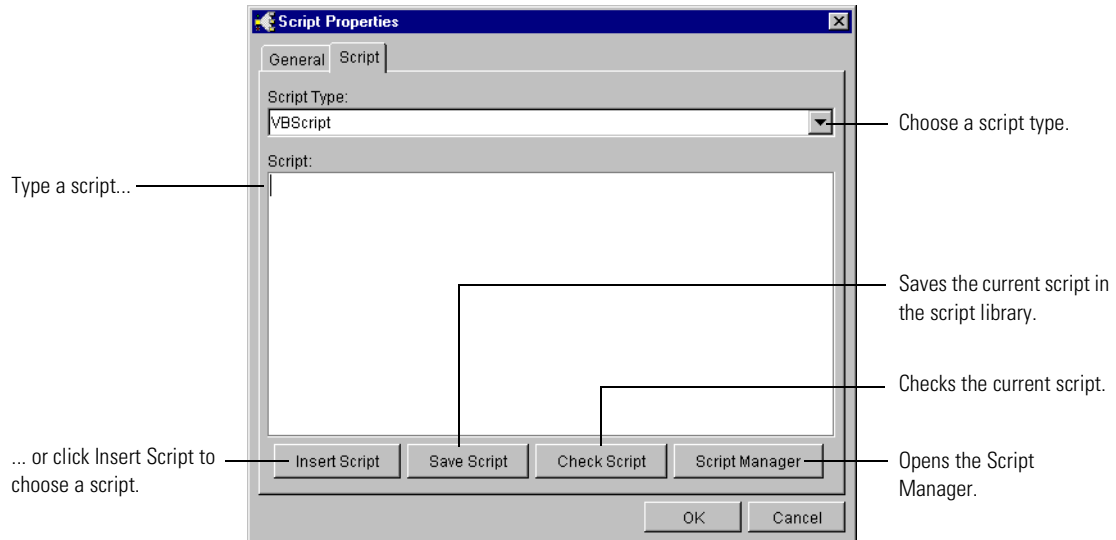
**2** Choose VBScript or JavaScript in the Script Type field.

**3** Type the script or click Insert Script to add an existing script.

You can also choose text in another script field, and copy and paste it. Or you can use the Script Editor (see *Using the Script Editor*, next).

If you click Insert Script, the Insert Script dialog box appears.



Choose a script to add it to the current Script field.
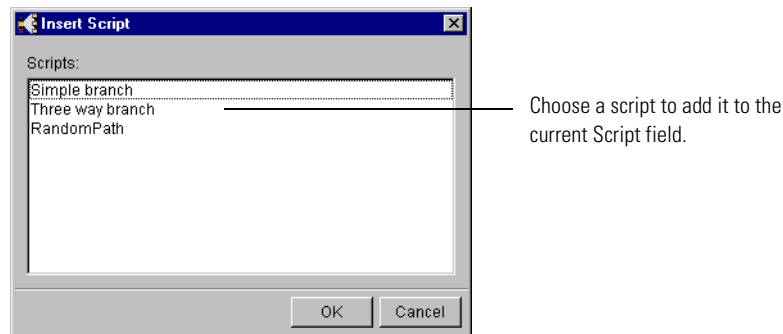
**4** Choose a script in the list and click OK.

Partner Agreement Manager adds the script you chose at the insertion point.

**5** When you've completed the script, click Check Script to have Partner Agreement Manager verify its syntax.

If there are any syntax errors in your script, Partner Agreement Manager displays Script Error messages.

# USING THE SCRIPT EDITOR

You can use the Partner Agreement Manager Script Editor to write and verify scripts in either VBScript or JavaScript. The Script Editor is divided into two panes. The left pane is the script area, where you write the script. The right pane shows the Business Object Catalog, which contains all of your element definition sets and business object types.



Choose the type of script you want to work with.

Use this pane to enter the script.

This pane displays the Business Object Catalog.
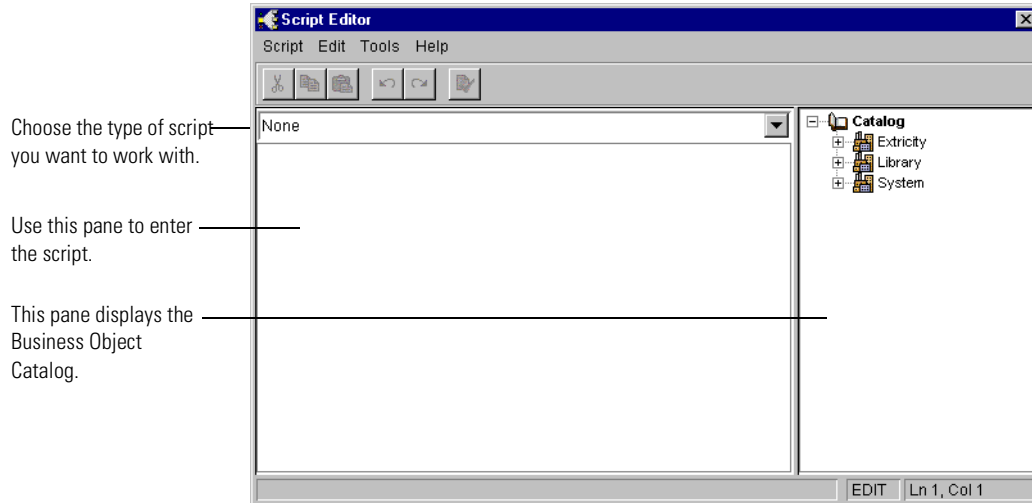
**To use the Script Editor:**

1 Choose the private process action for which you want to add a script.

   You can add a script to a script action or to any other type of action except termination or output object.

2 Choose Script Editor from the Edit menu.

   You can also right-click the action and choose Script Editor from the menu that appears.

3 Choose a script type and enter the script.

4 Choose Update Script from the Script menu to save the script.

5 To check the script's syntax, choose Verify from the Tools menu.

   For more information about using the Script Editor, see the *Partner Agreement Manager Script Developer's Guide*.

## Managing scripts

With the Script Manager, you can save scripts for reuse in other actions, create new scripts from scratch, edit existing scripts, or remove scripts you no longer need. Any scripts you create are available in all your private processes.

Choose the type of script you want to work with.

Choose a script to edit or remove.

Click to remove an existing script.

Click to create a new script.

Click to edit an existing script.
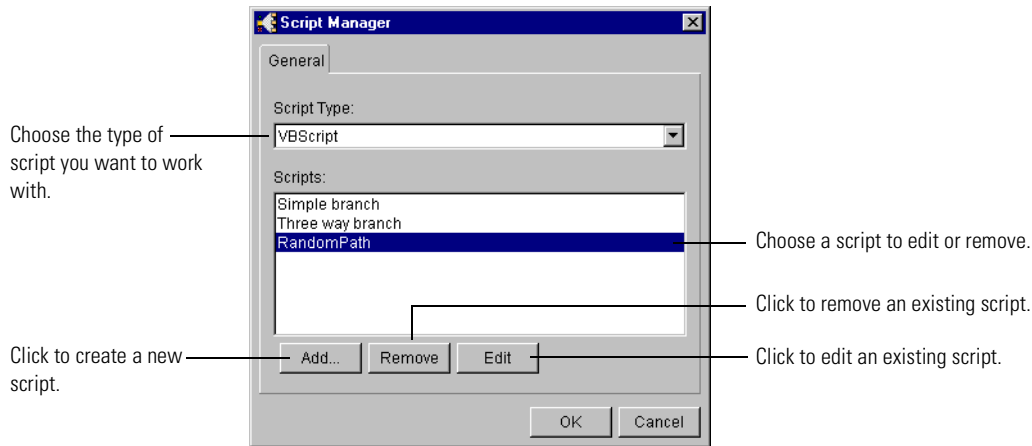
### To open the script manager:

▶ Click Script Manager in the Script tab in any action's properties dialog box.

| To | Do this |
|---|---|
| Save the current script for reuse | **1.** In the Script tab, click Save Script.<br>**2.** Type a name for the script and click OK. |
| Create a new script | **1.** In the Script tab, click Script Manager.<br>**2.** Click Add in the Script Manager dialog box.<br>**3.** Type a script name and choose the type of script you want to create.<br>**4.** Type the script (or copy and paste it from another application).<br>**5.** Click OK. |

| To | Do this |
|---|---|
| Edit an existing script | **1.** In the Script tab, click Script Manager. |
| | **2.** Choose a script type and a specific script, and click Edit in the Script Manager dialog box. |
| | **3.** Edit the script as necessary. |
| | **4.** Click OK. |
| Remove a script | **1.** In the Script tab, click Script Manager. |
| | **2.** Choose a script type and a specific script, and click Remove in the Script Manager dialog box. |
| | **3.** Click OK. |

# Setting action properties

All private process actions have general properties that consist of a label and an optional description. In addition, each type of action has its own set of properties that govern the tasks that each action performs. And to provide even more capabilities, each action also lets you attach a script written in VBScript or JavaScript that can be run with the action. Each action's scripting properties are the same as those for the script action. See *Adding a script* on page 155.

## Labeling an action

When you add an action to a private process, Partner Agreement Manager labels the action with the action type (for example, notification, script, and so forth). You can add a second label that appears in two lines under the main action label. Partner Agreement Manager uses the label you enter to list private process actions in the Audit window, so it's a good idea to make the label as descriptive as possible.

**Notification**
Was SO accepted or rejected?——— The label appears below the action name and in the Audit window.

**Tip:**  Because Partner Agreement Manager uses a variable-spaced screen font, the maximum display length of a label depends on its contents.

In addition to the label, you can also add a description for the action. Anyone who has access to a private process can display its description by opening the action's properties.

**To label an action:**

**1** Double-click the action.

You can also right-click the action and choose Properties, or choose the action and click the Properties button in the Command toolbar.

The action's Properties dialog box appears.

**2** Click the General tab.

The General tab appears.



Type a label for the action.

Add a description if you want.
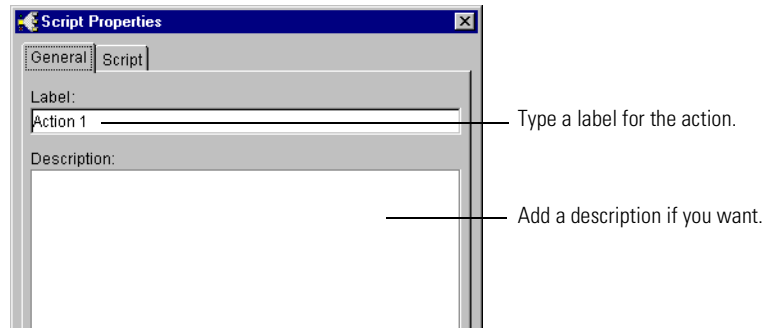
**3** Type a label and an optional description, and then click OK.

## Setting notification properties

A notification action sends e-mail, fax, or pager messages to the individual or group of people you specify. You can use a notification to send information you type into the message field or information that comes from another source (such as fields from a business object). If you plan to reuse the same message in other notification actions, you can save it as a template. You can also create a template instead of a message, or insert an existing template into the message. See *Managing notification templates* on page 165.

This notification conveys information that was typed directly into the Properties dialog box.
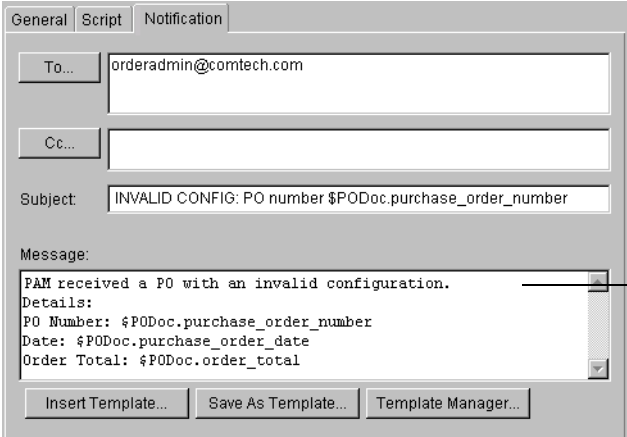
Everyone in these two lists receives the notification.

Each time this message is sent, it sends text that was typed directly into the Message field. You can also copy text from elsewhere and paste it into this field.

This notification uses information from fields that are part of a business object.

Each time this message is sent, it retrieves the current data from fields in the purchase order business object.

**To set notification properties:**

1 Double-click a notification action to open its Properties dialog box.

The Notification Properties dialog box appears.

Click to choose recipients.

Click to select copied recipients.

Type a subject.

Type a message.

Click to insert an existing message template.

Click to save the current message as a new template.

Click to create a new template or edit an existing one.
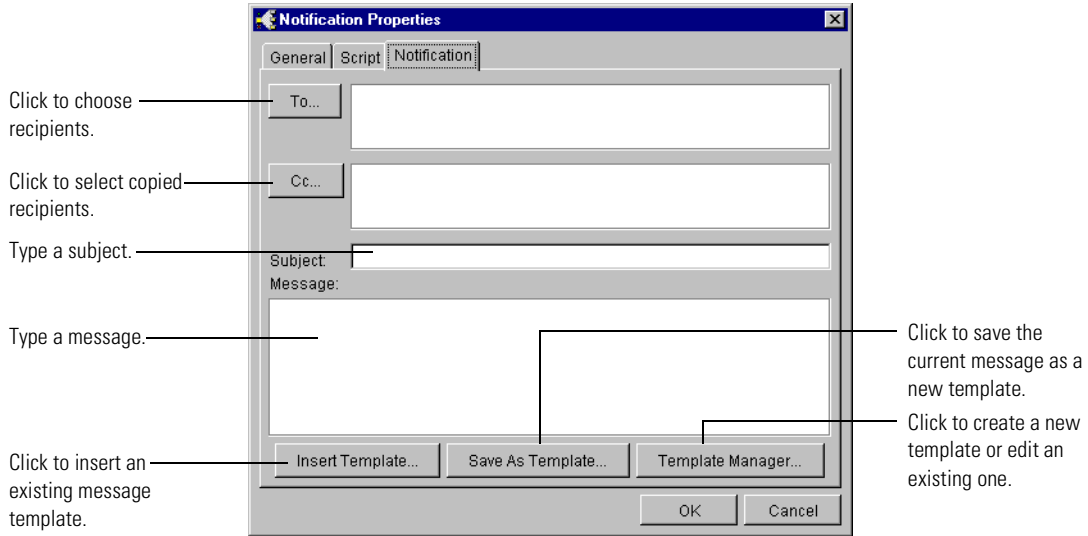
**2** Click the To button to choose recipients.

The Select Recipients dialog box appears.

Type a recipient's e-mail address and click Add.

Choose the variable that identifies the person to be notified as a result of the action, if any.

Choose a recipient and click Add.

The names of recipients you've added appear in this list.

These lists show the user names of all Partner Agreement Manager users at your company. When you choose a user from a list, Partner Agreement Manager supplies that user's e-mail address based on the user's contact information.

**3** To choose a recipient that is not a Partner Agreement Manager user at your company, type an e-mail address in the Type an Address field and click Add.

You can include as many recipients as you need.

**4**  Choose the variable that identifies the person to be notified as a result of the action, if any.

The variables list all variants in the private process, which might include the e-mail address of the person to notify. The value of the variable e-mail ID must be set in a script before notification.

**5**  To choose an Partner Agreement Manager user at your company, select an Email, Fax, or Pager address and click Add. You can also double-click the address.

You can include as many recipients as you need.

**6**  Click OK.

**7**  Click the CC button, add copied recipients to the list, and click OK.

**8**  Type a subject and add a message.

- To add a new message, type it in the Message field.

- To use a message that's been saved as a template, click Insert Template, choose one in the Notification Templates list, and click OK.

Partner Agreement Manager inserts the template you choose at the insertion point.

**NOTE:**  Partner Agreement Manager assumes that any text in the body of a message that follows a dollar sign ($) is the name of a variable. To include a price or other information that uses a $ in the message, you must precede the $ with an extra $—for example, "The price per widget is $$500."

**9**  Click OK in the Notification Properties dialog box.

## Managing notification templates

You can save the current message as a template for reuse in other notification actions. You can also create new templates from scratch, edit existing templates, or remove templates you no longer need. Any templates you create are available in all your private processes.

Choose a template to edit or remove. — 

Click to remove an existing template.

Click to create a new template.

Click to edit an existing template.
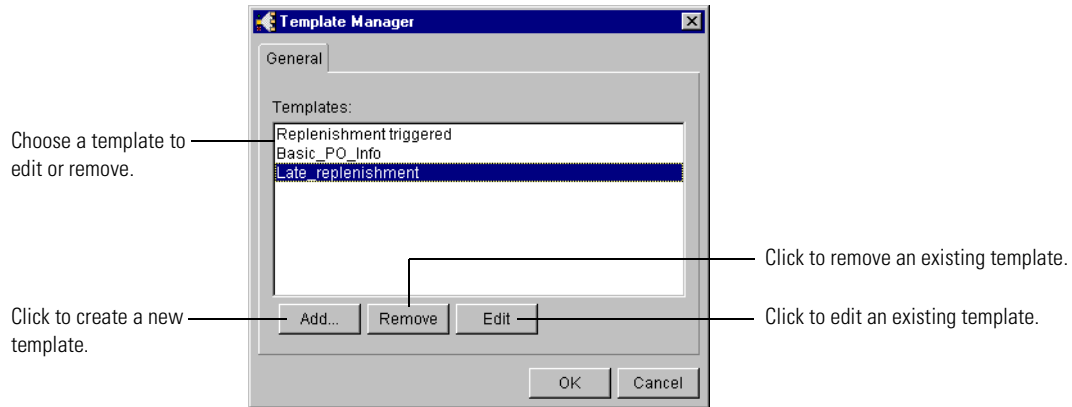
### To open the template manager:

▶ Click Template Manager in the Notification tab in the Notification Properties dialog box.

| To | Do this |
|---|---|
| Save the current message as a template | **1.** In the Notification Properties dialog box, click Save As Template.<br>**2.** Type a name for the template and click OK. |
| Create a new template | **1.** In the Notification Properties dialog box, click Template Manager.<br>**2.** Click Add in the Template Manager dialog box.<br>**3.** Type a template name and message.<br>**4.** Click OK. |

| To | Do this |
|---|---|
| Edit an existing template | **1.** In the Notification Properties dialog box, click Template Manager. |
| | **2.** Choose a template, and click Edit in the Template Manager dialog box. |
| | **3.** Edit the message as necessary. |
| | **4.** Click OK. |
| Remove a template | **1.** In the Notification Properties dialog box, click Template Manager. |
| | **2.** Choose a template, and click Remove in the Template Manager dialog box. |
| | **3.** Click OK. |

## Setting approval properties

There are two main parts to an approval action: an e-mail message to a Partner Agreement Manager user at your company, and an approval form that is located on your Partner Agreement Manager Web server. The e-mail message is almost identical to the one generated by a notification action.

The e-mail message also contains the URL of the approval form.

The notification portion of the approval generates an e-mail message that contains your instructions to the approver.

When you send an approval, Partner Agreement Manager includes the address of a dynamically generated Web page that contains an HTML form. The HTML form contains the questions you specify and space for the recipient to respond.

The approval portion creates the HTML form.

Instructions for Recipient:

Click YES to approve the PO for replenishment.
Click NO to send the PO to the purchasing manager for review.

Questions:

Do you approve this PO?

## Approval

User Name Charles Grey          Request ID 100015
Request Time 10/15/99 5:05:53 PM  Process Type

**Instructions**
Click YES to approve the PO for replenishment.
Click NO to send the PO to the purchasing manager for review.

**Questions**
Do you approve this PO?

You can specify how long to wait for the recipient to respond, and include a script that instructs Partner Agreement Manager to send the approval to another user after the time-out has expired. In this example, the approval action is addressed to a variable called **Approver**, and the time-out is set to four hours. The time-out status is stored in the **timeout_status** variable. Valid values for the time-out variable are **success** and **timeout**.

The recipient for this action is a variable: Approver.

The time-out for this action is 4 hours, and the time-out status is stored in a variable you create: timeout_status.

Recipient

○ User:  engr_mgr

● Variable:  Approver

Timeout Interval

Days:

0

Hours:

4

Timeout Status Variable:

timeout_status

In the following example, an initial script action sets the value of **Approver** to **PurchasingMgr.** The script attached to the approval action looks at the value of **timeout_status.** When the time-out has expired, the script changes the value of Approver to the user name of another Partner Agreement Manager user. A loop in the private process sends the approval action to the new recipient.

Script Type:

VBScript

Script:

```
SUB MAIN
        timeout_result = getVar("timeout_status")
        current_approver = getVar("Approver")

        IF timeout_result = "timeout" THEN
                IF current_approver = "PurchashingMgr" THEN
                        ' escalate now
                        setVar "Approver", "VPPurchasing"
                        setPath "Send approval to VP Purchasing"
                ELSE
                        ' already escalated before
                        setPath "Main"
                END IF
        ELSE
                ' was approved or rejected
                setPath "Main"
        END IF
END SUB
```

This script checks the value of timeout_status. When the time-out has expired, the script escalates the approval to the VP Purchasing.

**To specify approval settings:**

**1** Double-click an approval action to open its Properties dialog box.

The Approval Properties dialog box appears, with the Notification tab displayed. The Subject and Message areas are the same as those in the Notification Properties dialog box.



Select a recipient.

Type a subject.

Type a message or select a message template.
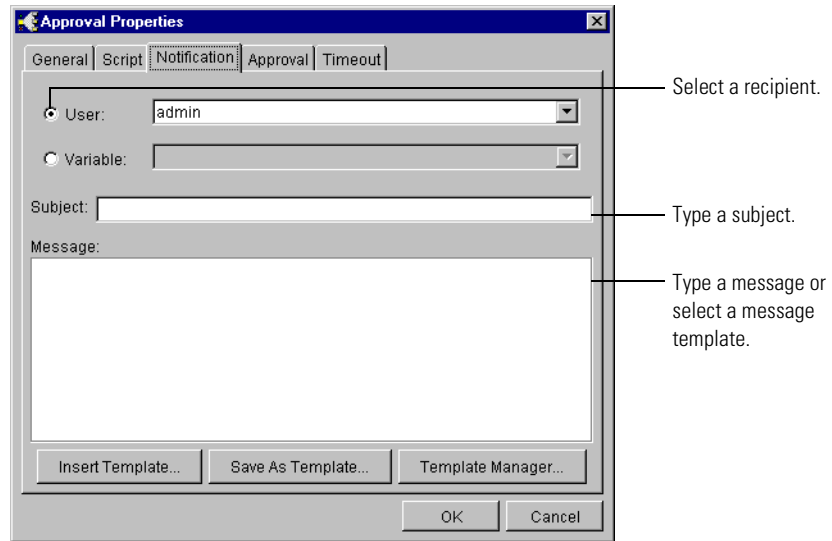
**2** Choose a recipient.

You can choose an existing Partner Agreement Manager user, or select a variable as a placeholder. You can use any variant variable set up in the current public process or private process. You use a variable, for example, if you plan to use a script to reroute the approval after it times out.

**3** Type a subject and add a message.

The message appears in an e-mail message that gives the URL for a page on the Partner Agreement Manager Web server. You can type a message that directs the recipient to the Web site. And a message has already been saved as a template, you can click Select Template to include the existing message template. For more information about message templates, see *Managing notification templates* on page 165.

**4** Click the Approval tab to enter instructions and questions.

The Approval tab appears. The instructions and questions you enter here appear on an HTML page on the Partner Agreement Manager Web server.

| | |
|---|---|
| General | Script | Notification | **Approval** | Timeout | | |
| Instructions for Recipient: | Type instructions for the recipient. |
| Questions: | Use these buttons to move questions up or down in the list. |
| Click to add questions. — Add... Remove Properties | |
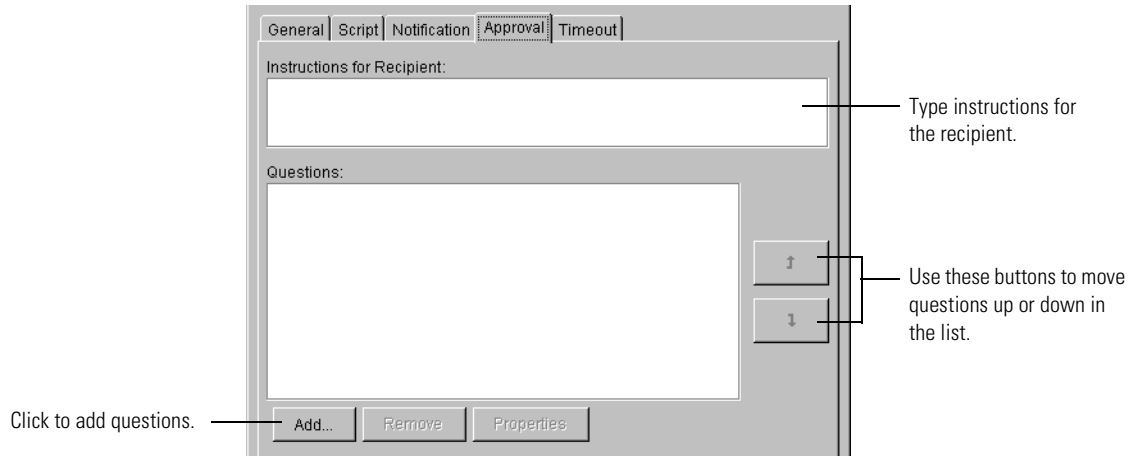
**5** Type instructions for the recipient, and click Add to add one or more questions.

The Approval Question dialog box appears.

Approval Question — Type a question.

Response — Select the type of response you want.
- Text
- Yes/No
- Multiple Choice

Enter each multiple choice option on a separate line.

Response Variable — Choose a variable to store the response.

OK    Cancel
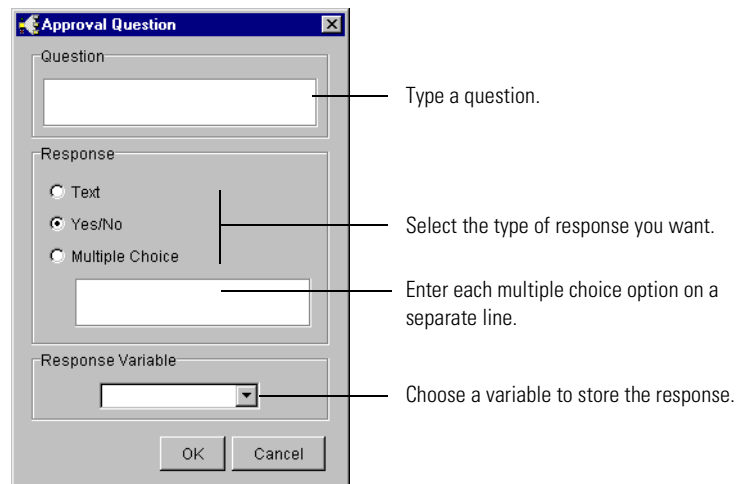
**6** Type a question and select a response type.

| If you select | Partner Agreement Manager creates | Sample |
|---|---|---|
| Text | A text field where the recipient can type an answer. | Ship-to # |
| Yes/No | Two option buttons labeled Yes and No. The result returned by the Yes/No question is 1 for Yes or 0 for No. | Do you approve?  ○ Yes    ○ No |
| Multiple choice | A drop-down list that includes the selections you specify. | What day is today?  Monday ▾ |

**7** Choose a variable to store the response to your question.

You can use any variant variable defined for this public or private process. In this example, the script uses the response variable to set the value of another variable used elsewhere in the private process.

This variable stores the response to the approval question.

Response Variable:

approval_ship_to_number

This script moves the value from the approval action's response variable to a field of a business object.

```
Script:
Sub main()

    ShipToNumber.number = getVar("approval_ship_to_number")

End Sub
```
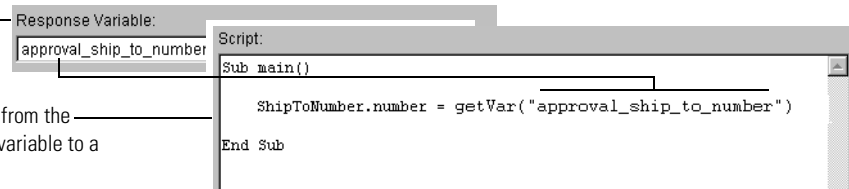
**IMPORTANT:** You must define the variable before you can use it in an approval action.

**8** Click OK.

The question appears in the Questions list. You can click Add again to add more questions and use the up-arrow or down-arrow buttons to move questions up or down in the list (and change the order in which they appear in the approval page).

**9** Click the Timeout tab to set the time-out interval.

The Timeout tab appears. The time-out interval determines how long to wait for the recipient to respond. If the recipient doesn't respond within the set interval, the request fails and the approval server returns a time-out indicator. You can store the time-out indicator in a variable and use it in a script to resend the approval to another user. Valid values for the time-out variable are success and timeout. See the example on page 168.

Set the number of days and hours until the approval times out.

Values of 0 and 0 set the approval action to never time out.

Use a variable to set a flag when the recipient fails to respond before the approval times out.
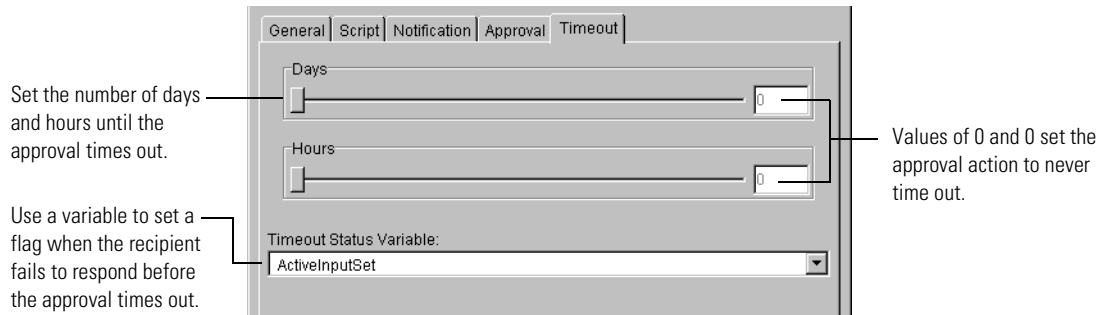
**10** Set the time-out interval, choose a time-out status variable, and click OK.

Here's a partial example of a script that uses the time-out status variable.

This variable stores the time-out indicator.

This script uses the time-out indicator to send the approval to another user.

```
Timeout Status Variable:
timeout_status
Script:
SUB MAIN
        timeout_result = getVar("timeout_status")
        current_approver = getVar("Approver")

        IF timeout_result = "timeout" THEN
                IF current_approver = "PurchashingMgr" THEN
                        ' escalate now
                        setVar "Approver", "VPPurchasing"
                        setPath "Send approval to VP Purchasing"
```

## SETTING MAPPING PROPERTIES

With a mapping action, you can call an existing map in a private process. The map determines how the fields in a business object type are filled from another source. The source can be another business object type, or a string specified in the map. You might use a mapping action, for example, to transform data from a business object type generated by an enterprise system into a business object type that a partner requires.

When you add a mapping action, you specify the output business object type, an output variable, and the name of the map. If the map has inputs, you must also set variables for the map inputs. The private process uses the output variable to store the output business object type created by the map. The input variables provide the input data for an instance of each input business object type used in the map.

**To set mapping properties:**

1 Double-click a mapping action to open its Properties dialog box.

The Mapping Action Properties dialog box appears.

Choose the business object type to be populated by the map.

Choose the map.

Choose an output variable.
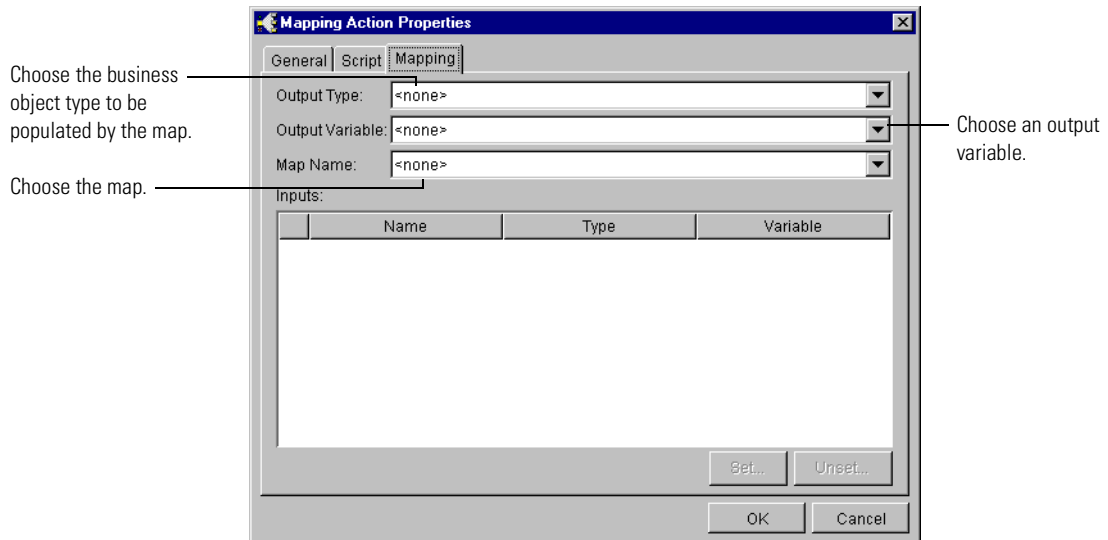
2 Choose the business object type to be populated by the map in the Output Type list.

3 Choose an output variable in the Output Variable list.

Partner Agreement Manager displays any business object variables of the same type as the map output.

4 Choose the map in the Map Name list.

Partner Agreement Manager lists any inputs for this map in the Inputs area.



Partner Agreement Manager lists all inputs for the map you choose.

Click to set the variable for this input.

Click to unset the variable for this input.

**5** Choose an input and click Set to set an input variable.

The Select Variable dialog box appears.



Double-click a variable to choose it.

**6** Choose the input variable and click OK.

The Mapping panel shows the name of the variable you choose.



The variable you chose appears here.

**7** Set any other variables as needed, and click OK.

You must set an input variable for each input into the map.

## Setting timer properties

A timer action inserts a pause into the private process. You might use a timer action to delay a process while it waits for an event to occur. For example, you might want to allow time for a system to update, or for more e-mail messages to arrive. The maximum wait time is 100 hours and 59 minutes.

**To set timer properties:**

1 Double-click a timer action to open its Properties dialog box.

The Timer Properties dialog box appears.

Set the wait time in hours and minutes.
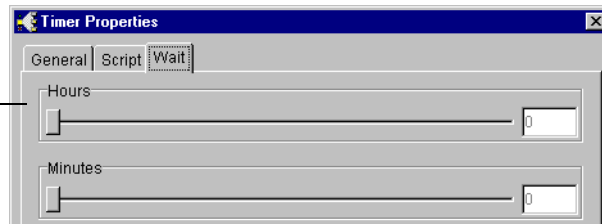
2 Set the wait time and click OK.

## Setting extension properties

Through an extension action, Partner Agreement Manager can communicate with an external application. The operations you can perform depend on the Partner Agreement Manager adapters that are installed. Partner Agreement Manager adapters can perform three types of operations: Get, Post, and Advanced.

- A Get operation obtains information from the enterprise system and uses it to populate a Partner Agreement Manager business object. You might, for example, get information from a purchasing system to populate a purchase order business object. A Get operation is uniquely defined by the business object it generates (its output).

- A Post operation takes information from a Partner Agreement Manager business object and saves it into the enterprise system. For example, you might create a new drawing document in a CAD folder that consists of the contents of a design drawing field in a business object. A Post operation is uniquely defined by the business object from which it obtains the information (its input).

- An Advanced operation performs a combination of Get and Post operations. You can use an Advanced operation to accomplish several tasks at once. For example, you might use an Advanced operation to get a purchase order number from the purchasing system and related parts information from the inventory system to produce a purchase order business object and an inventory business object. An Advanced operation is uniquely identified by its description.

When you set extension properties, you choose an adapter that has been registered with Partner Agreement Manager. You then set the appropriate properties for the adapter you choose.

Each adapter has four sets of properties: General, Inputs, Outputs, and Execution.

- General: Lets you choose the type of operation to be performed and the operation. The adapter instance you choose determines the operation types you can perform. The operation type, in turn, determines the operations you can choose. An adapter instance can have more than one operation type associated with it, and an operation type can have more than one operation. For a Get or Post operation, you specify a business object. For an Advanced operation, you choose a description of the action to be performed.

- Inputs and Outputs: Let you bind the inputs and outputs for the adapter instance you choose to local variables in the private process. The adapter you choose determines which input and output business objects must be bound to private or public process variables. How you have defined variables in the public and private processes determines which variables are available for binding.

- Execution: Lets you specify a time-out for the extension action. You can also set the extension action to wait until the action is completed. The advantage of setting a time-out is that you can capture time-out information in a special variable that you can then use to evaluate audit information about processes that do not run to completion.

**To set adapter extension properties:**

1 Double-click an extension action to open its Properties dialog box.

The Extension Properties dialog box appears.
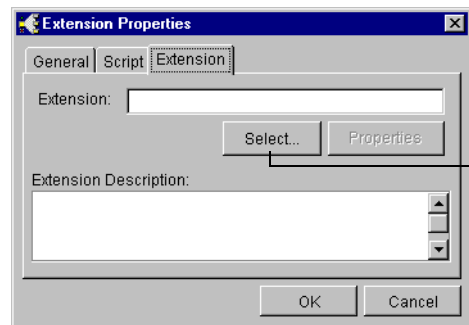


Click to select an extension.

**2** Click Select to choose an extension.

The Select Extension dialog box appears.
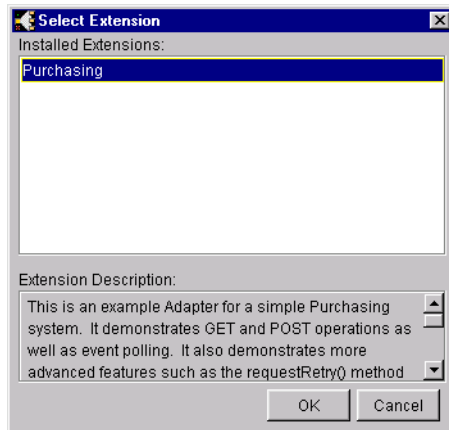


This list shows the extensions available at your location. The contents of your list might be different.

**3** Choose an extension from the list and click OK.

The adapter you choose determines the properties you can set.

**4** On the Extension Properties dialog box, click the Properties button.

The Properties dialog box for the adapter appears.
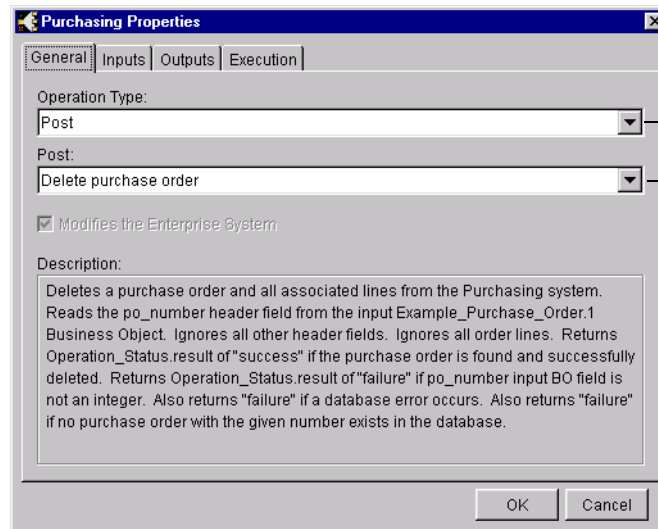


Choose an operation type.

Different operations appear here depending on the operation type you choose.

**5** Choose an operation type and a specific operation from the dropdown lists.

**6** Set the Inputs and Outputs properties for the operation type.

Depending on the operation type you chose on the General tab, you will see different versions of the Inputs and Outputs tabs and bind the input variables in different ways. Setting the properties is described in separate procedures for each operation type.

| If you chose this operation type | Follow the instructions in this section |
|---|---|
| Post | *Setting input and output properties for POST operations*, next. |
| Get | *Setting inputs and outputs options for GET operations* on page 180. |
| Advanced | *Setting inputs and outputs properties for Advanced operations* on page 181. |

**7** When you finish setting Inputs and Outputs properties, click the Execution tab and set Executions properties.

You can set the extension action to wait until it finishes its action or to time out after a specified period of time. If you specify a time-out interval, you can also set how the extension action terminates when the time-out is reached.



Specify the length of the time-out interval.

Set this option to terminate the private process if the extension action times out.

Set this option to terminate the extension action but keep the private process running.

Specify the name of a variable to be set when a time-out occurs.

| Use this setting | To |
|---|---|
| Terminate action after time-out | Terminate the extension action, but let the private process continue running. If you choose this setting, you can specify a variable in the private process to be set if a time-out occurs. You can then use a script action in the private process to test for a time-out and proceed accordingly. If you plan to use a time-out variable, you must define it in the private process. |
| Abort process after time-out | Terminate the private process if the extension action times out. |

**8** Click OK on the adapter instance's Properties dialog box. Click OK on the Extension Properties dialog box.

#### To set input and output properties for Post operations:

**1** Click Inputs to bind the adapter's input variables to private or public process variables.

The Inputs tab appears.



The adapter's business object type appears here.

Choose a value for the input variable.

A description of the business object type appears here.

**2** Choose a value for the input variable, and then click the Outputs tab.

The Outputs tab appears.



Choose a local variable in this list to hold status information.

**3** Choose a variable to hold status information.

When you finish setting options on the Inputs and Outputs tabs, continue with step 7 on page 178.

### Setting inputs and outputs options for GET operations

If you chose a Get operation in step 5 on page 177, continue with these steps.

**To set input and output properties for Get operations:**

**1** Click Inputs to bind the adapter's input variables to private or public process variables.

The Inputs tab appears.



Select an input variable.

Click to set the variable to a local variable.

A description of the selected variable appears here.

**2** Choose an input variable and click Set.

The Select Binding dialog box appears.

**3** Double-click a variable to bind it to the input variable.

Or select the variable in the list, and then click OK.



Choose a local variable in this list.

**4** Click the Outputs tab to bind the adapter's output variables to local variables in the private process.

The Outputs tab appears.



Choose a variable in this list to bind to the output variable.

Choose a local variable in this list to hold status information.
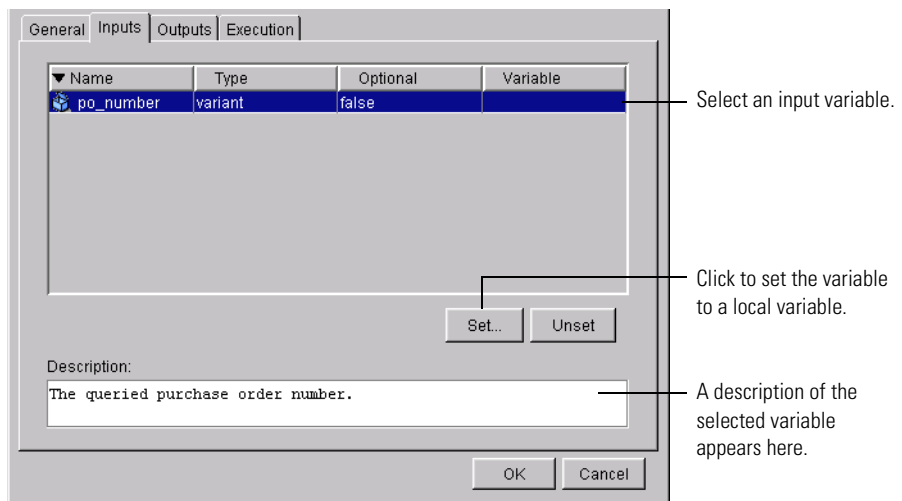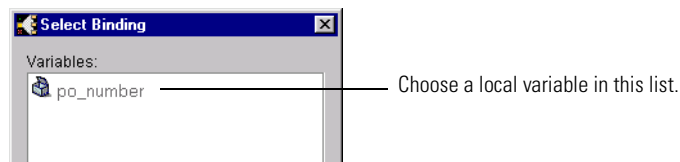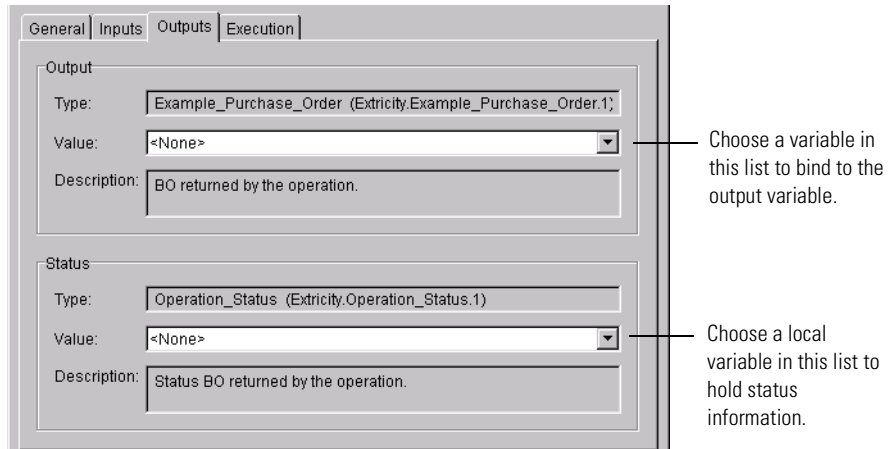
**5** Choose an output variable and choose a variable to hold status information.

When you finish setting options on the Inputs and Outputs tabs, continue with step 7 on page 178.

### Setting inputs and outputs properties for Advanced operations

If you chose an Advanced operation in step 5 on page 177, continue with these steps.

**1** Click Inputs to bind the adapter's input variables to private or public process variables.

The Inputs tab appears.



Select an input variable.

Click to set the variable to a local variable.

**2**  Select an input variable and click Set.

The Select Binding dialog box appears.

**3**  Double-click a variable to bind it to the input variable.

Or select the variable in the list, and then click OK.



Choose a local variable in this list.
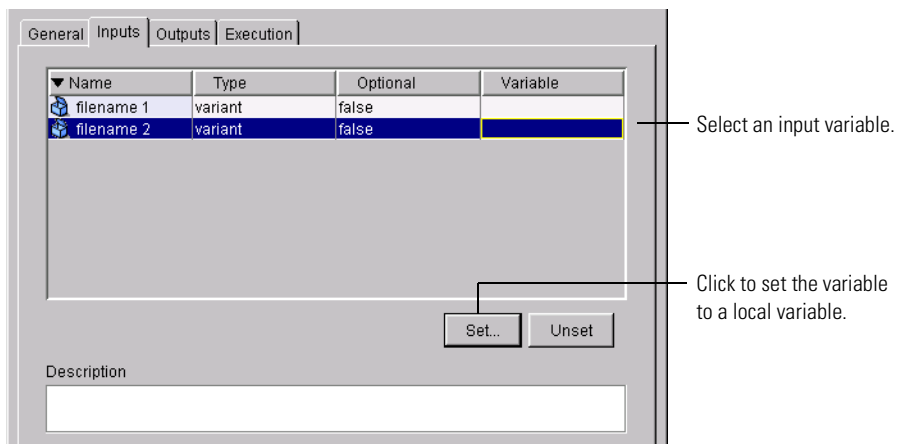
**4**  Click the Outputs tab to bind the adapter's output variables to local variables in the private process.
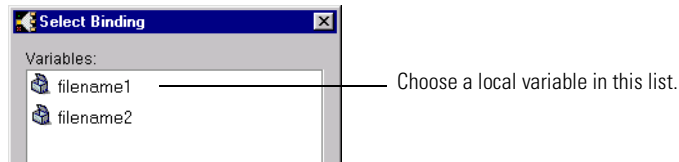


Select an output variable.

Click to set the variable to a local variable.
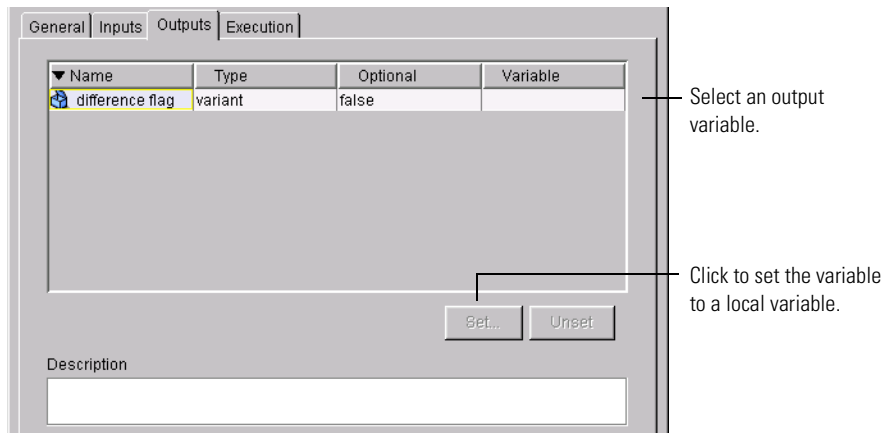
**5**  Choose an output variable and click Set.

The Select Binding dialog box appears.

**6**  Choose a variable in the Select Binding dialog box, and then click OK.

When you finish setting options on the Inputs and Outputs tabs, continue with step 7 on page 178.

## Setting subprocess properties

A subprocess action initiates another public process from within the current private process. With a subprocess action, you can reuse an existing process that already contains all of the steps you need. You can run any public process as a subprocess as long as your company is the first node in that public process. Because a subprocess is part of your private process, you can't run a subprocess that starts with one of your partners.



For this public process step...

...a private process uses a subprocess action...

...that calls this public process...

...and this private process.
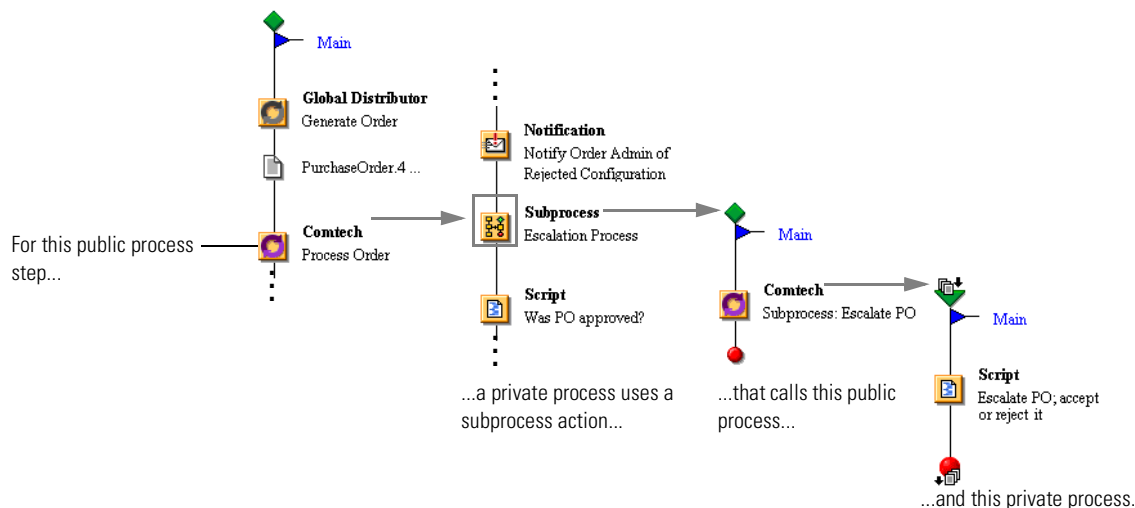
When you set the properties for a subprocess action, you choose the process that you want to run. With Partner Agreement Manager, you can choose from all installed public processes that start with your company. After you choose a process, you can map input and output variables from the subprocess to variables in the current public or private process. You can use variables to pass variables from the private process into the subprocess and to store the output from the subprocess. You can also specify how the subprocess knows which partner in a partner group will participate. You can specify the partner by name, or specify a variable that will pass the appropriate partner ID (or a combination of partner name and ID) into the subprocess when needed.

**NOTE:** Define any variables you need before you set subprocess properties, and be sure to include the partner ID in the variable.

You can also set up the conditions under which the subprocess runs. First, you can specify whether the private process must wait for the subprocess to finish running before continuing. If you want to wait, you can set how long—for a specific interval, or until it finishes, no matter how long. If you set an interval, you can also specify whether to terminate the subprocess after it times out.

**To set subprocess properties:**

1 Double-click a subprocess action to open its Properties dialog box.

The Subprocess Properties dialog box appears.
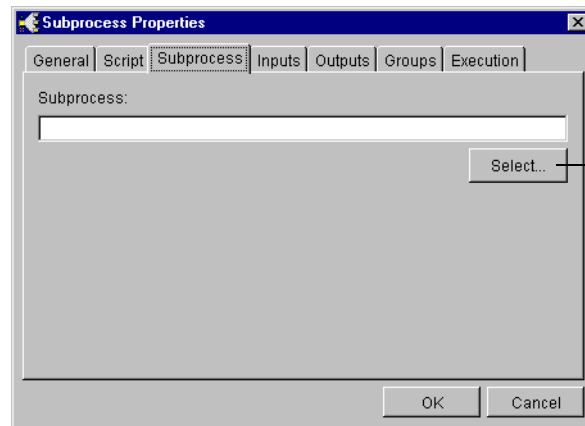


Choose the public process that you want to run as a subprocess.

2 Click the Select button to choose a public process.

The Select Process dialog box appears.
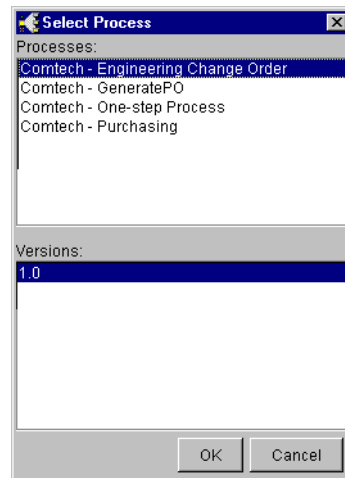


This list shows all installed processes for which your company has the first action.

This list shows available versions for each process.

**3** Choose a process and a version, and click OK.

**4** Click the Inputs tab to bind the input variables for the subprocess to variables in the private process.

The Inputs tab displays the input variables for the subprocess.



This list shows the input variables in the subprocess. Choose a variable to map it to a private process variable.
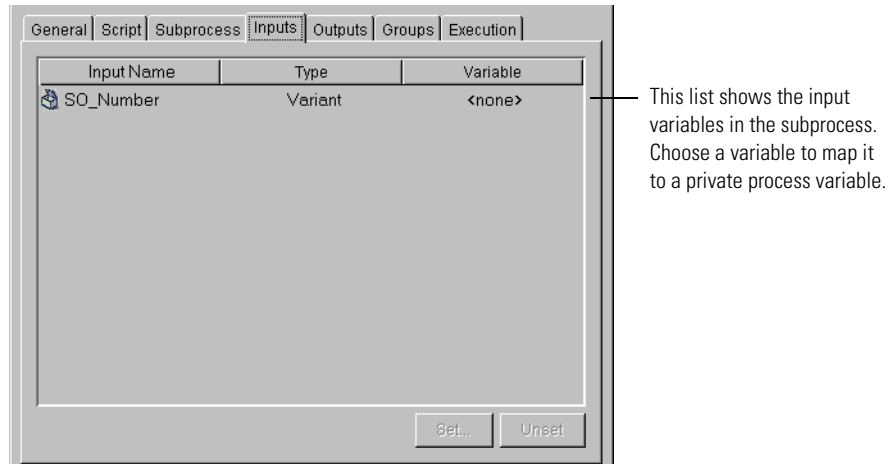
**5** To bind an input variable for the subprocess to a variable, choose the input variable and click Set.

The Select Variables dialog box appears.



This list shows the variables defined for the private process. that match the input variable you chose.
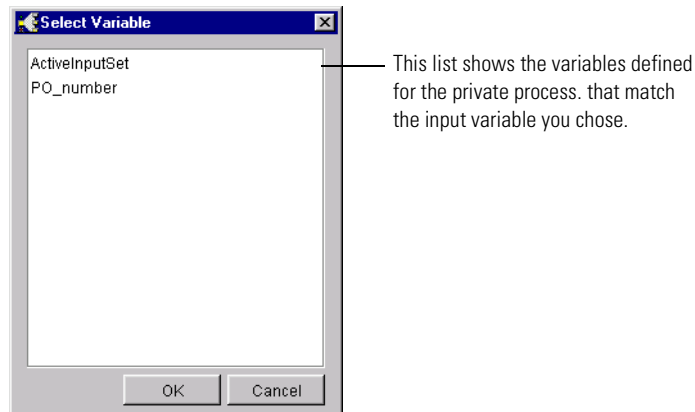
**6** Choose a variable and click OK.

**7** Click Outputs to bind a subprocess output variable, choose the variable, click Set, and double-click a variable in the Select Variables list.

**8** If an action in the subprocess is assigned to a partner group, click the Groups tab to specify a partner.

The Groups tab appears. You can either specify a partner ID, or choose a variable that will pass the partner ID into the subprocess when it runs.

You can specify a partner or a variable that will supply the partner ID when needed.
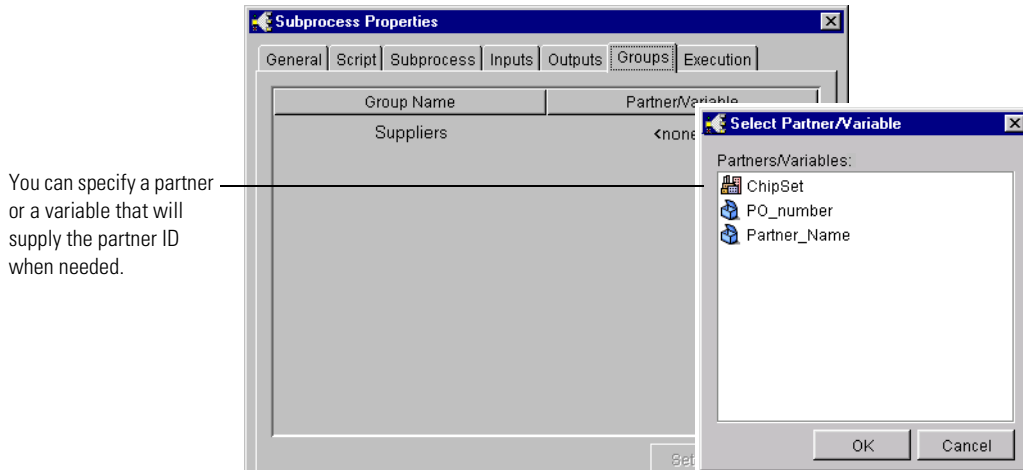


**9** Choose a group, click Set, and double-click a partner or variable in the Select Partner/Variable dialog box.

If the partner group contains only one partner, that partner is selected.

**10** Click the Execution tab to set the rules for executing the subprocess.

The Execution tab appears.



If you use this setting and the subprocess is later suspended, the subprocess will not start.

Use these settings to specify the time-out period.
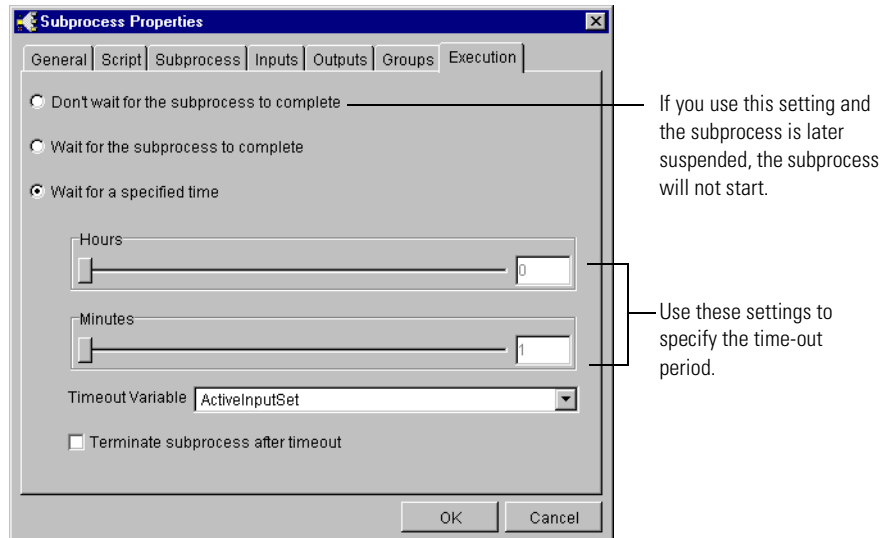
**11** Specify the appropriate options and set a time-out interval.

| To set Partner Agreement Manager to | Do this |
|---|---|
| Start the subprocess and continue with the next action | ■ Select Don't Wait for the Subprocess to Complete. |
| Start the subprocess and wait for it to finish before continuing | ■ Select Wait for the Subprocess to Complete. |
| Start the subprocess and wait a specific period of time for it to finish before continuing | ■ Select Wait for a Specified Time and use the sliders to set the time-out interval. |
| Stop the subprocess if it doesn't finish within the allotted time. | **1.** Select Wait for a Specified Time and use the sliders to set the time-out interval.<br>**2.** Select Terminate Subprocess after Timeout. |
| Let the subprocess finish running even though it has timed out | **1.** Select Wait for a Specified Time and use the sliders to set the time-out interval.<br>**2.** Clear the Terminate Subprocess after Timeout check box. |

**12** Click OK.

Tɪᴘ: To view the subprocess, right-click the subprocess action in the private process and choose View Subprocess from the menu that appears. Partner Agreement Manager opens a Process window and displays the process.

## Sᴇᴛᴛɪɴɢ ᴏᴜᴛᴘᴜᴛ ᴏʙᴊᴇᴄᴛ ᴘʀᴏᴘᴇʀᴛɪᴇꜱ

Output object is the action you use to map the output from a private process to a business object and path in the public process.

Each private process begins with an input set and ends with an output set. The input set—in the form of business objects—comes from the public process and serves as the starting point for the private process. The output set—also in the form of business objects—conveys the results of the private process back to the public process.

Partner Agreement Manager uses a combination of the business object type and path name to uniquely identify each business object in a public process.



**Script** — Was PO approved?

Yes | No

**Script** — Create PO Acknowledgment | **Script** — Create PO Rejection

**Output Object** — Send Acknowledgment | **Output Object** — Send Rejection

**Comtech** — Process Order

Order Accepted — PO_Acknowledg... | Order Rejected — PO_Rejection.1 -...

The business object type and path name uniquely identify this expected output from the private process.

The public process expects these two outputs: PO_Acknowledgment and PO_Rejection.

The private process must include at least one output object that maps to each output that the public process expects.

Similarly, each output object uses a business object variable and an output message path for its definition. The business object variable provides temporary storage for the information generated in the private process. The business object variable you use here must be the same type of object that the public process expects.



**Comtech** — Process Order

Order Accepted — PO_Acknowledg...

These settings...

Business Object Variables:
I1
HoldResultBO
PO_AckBO
PO_RejectionBO
EscalationResult

Type:
PO_Acknowledgment.1 - Comtech

Output Messages:
Order Accepted

...define this output object action.

**Script** — Was PO approved?

Yes

**Script** — Create PO Acknowledgment

**Output Object** — Send Acknowledgment

The public process expects an output object of the type PO_Acknowledgment on the Order Accepted path.

The output object's output message path corresponds to a path in the public process. Unless the public process contains several business objects with exactly the same name, there's usually only one path available for each business object variable.

You can place an output object anywhere in a private process, but it behaves as if it were the last action in the private process. If your public process contains branches, you must have an output object for each branch.

> **IMPORTANT:** You must define the business object variable in the private process before you can assign output objects to it. See *Creating a variable* on page 136 for more information.

**To set output object properties:**

1  Double-click an output object icon to open its properties.

The Output Object Properties dialog box appears.
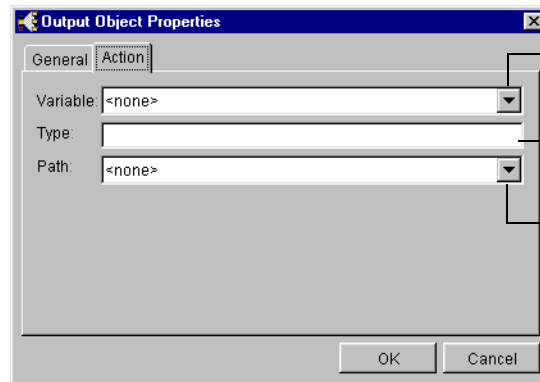


This list shows all available business object variables for the private process.

This is the type of business object the variable is based on.

This list shows all available output message paths that have a business object of the same type.

2  Choose a variable in the list of available Business Object Variables.

Partner Agreement Manager displays the business object type for the variable, as well as all paths that contain that type of business object.

3  If necessary, choose a path in the Path area.

4  Click OK.

## SETTING TERMINATION ACTION PROPERTIES

Within a process, there are some situations where it's appropriate to logically terminate a process according to your business rules. In a purchasing system, for example, you might want to stop a process if an enterprise system generates an incomplete configuration or a purchase order is corrupted. Or you might want to stop a process because approval has been denied. In each case, the cause for termination lies in the business rules, not in the Partner Agreement Manager implementation.

If you want, you can allow the process to run to completion and generate an error. Or you can anticipate the problem condition in your design and include a termination action that stops the process under a specific set of circumstances. As an added benefit, using a termination action provides audit information about the cause of the problem.

When you use a termination action, you specify the reason for terminating the process. Partner Agreement Manager is preset to include several typical reasons: denial of approval, missing data, data out of range, or formatting errors. You can add reasons tailored to your needs. See *Specifying additional termination reasons* on page 193.

When you run a process that ends with a termination action, Partner Agreement Manager displays the termination reason in the Auditor folders in the Process Manager window. You can sort messages by status to identify process instances that terminated for the same reason.

**TIP:** To sort messages by status, right-click the Status field in any process log entry, choose Sort by Status from the menu that appears, and choose Ascending or Descending sort order.

To view the message in the Process Audit window, double-click it in the Auditor folder. From the Process Audit window, you can click the Errors tab and double-click a message to display the message. To display more information, click the Show Details button.

The explanation for the termination action appears in the system message it generates.

Click to get more information about the error.

The reason you specify for the termination appears in the Process Audit window.
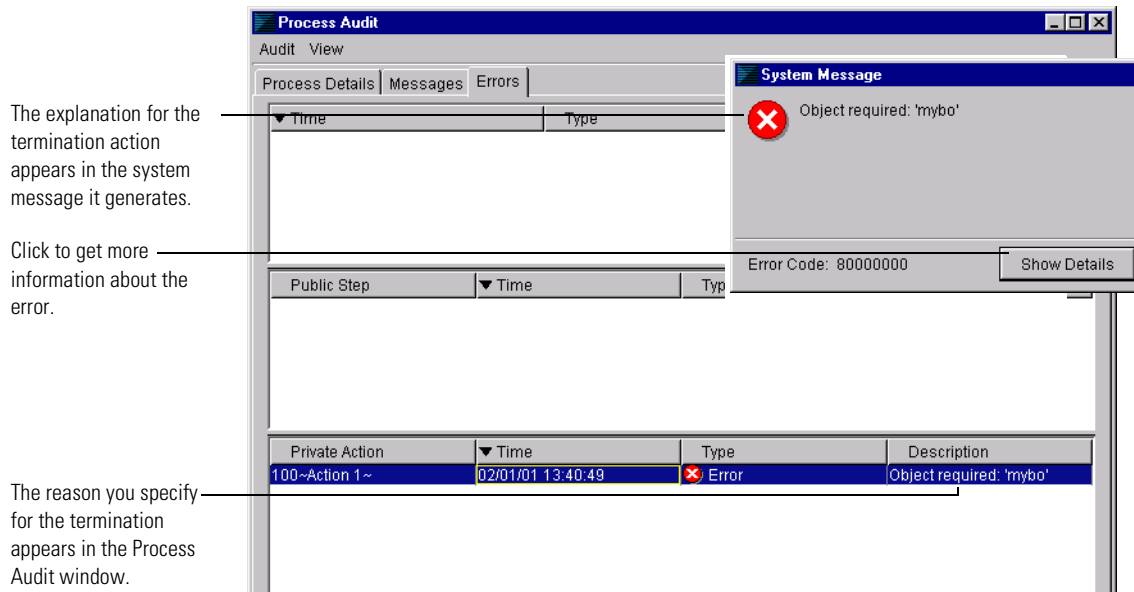


When you include a termination action in a private process, you can provide details that pertain to the termination by substituting business object field values for variables.

This information comes from a business object field.

This is the error code for the termination action.

This is the detailed information for the termination.
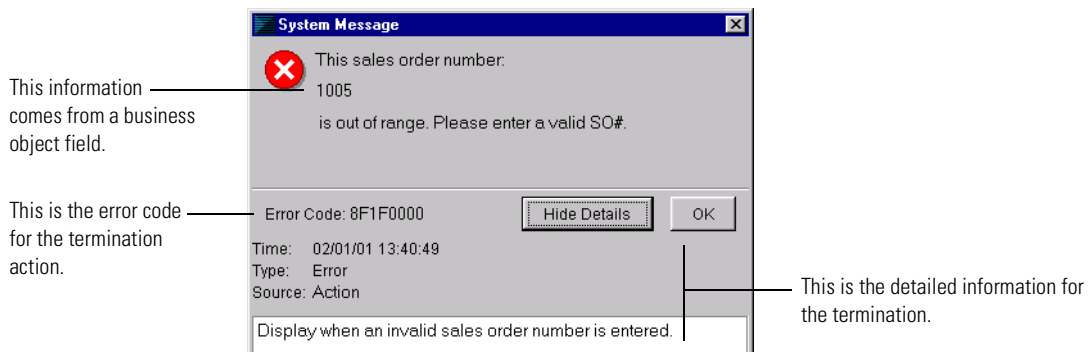
When you add a termination action, you specify the reason for termination, and add a description. Because a termination is only one of the possible outcomes of a process, the most common location for a termination action is on one of the branches of a conditional branch block. A script that precedes the termination action defines the condition under which the termination action is executed.

**TIP:** If you want to include other actions that are triggered by the same condition as the termination—a notification, for example—you can insert them between the script and the termination action.

**To set termination properties:**

1 Double-click the termination action to set its properties.

The Terminate Properties dialog box appears.



Choose a reason for termination.

Add an explanation for the termination.
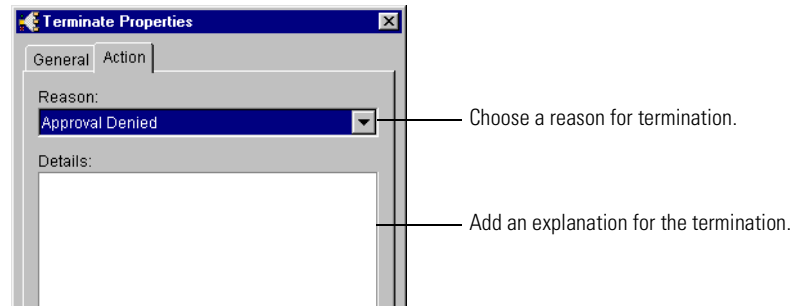
2 Choose a termination reason in the Reason list.

When you run the process and a termination occurs, the reason you choose appears as the status in the audit messages and in the status field in the Partner Agreement Manager Auditor.

3 Type an explanation for the termination in the Details field.

The description you enter appears as the explanation for the warning message generated when a process terminates. If you want, you can include data from a variant or business object variable in the Details field.



This is a field (SO_Number) in a business object variable (SONum). The $ indicates that it's a variable.
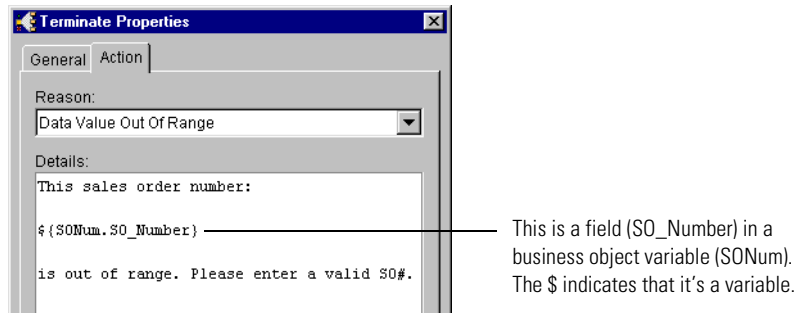
You can also click the General tab and label the action.

**4** Click OK.

### Specifying additional termination reasons

Partner Agreement Manager is preset to include a few of the most common reasons for terminating a private process: out-of-range or missing data, data format errors, or approval refusal. These reasons appear in the Termination Properties dialog box, in audit messages, and in the status field in the Partner Agreement Manager Auditor.

If you want, you can add other reasons that are specific to your Partner Agreement Manager installation.

**NOTE:** This task is typically performed by the Partner Agreement Manager administrator.

The reason codes are stored in a text file, Terminate.Properties, located in the PAM Resources folder. You can edit the contents of this file using a text editor, such as Notepad.

Each reason has a code and description. Partner Agreement Manager uses the code internally and displays the corresponding reason when you set properties or view an audit item. Reason codes are five-digit numbers starting with 32768. Both the codes and descriptions must be unique.

This is a sample of part of the Terminate.Properties file.

```
#-------------------------------------------------------
# The user-defined terminate reasons (> 32767)
#-------------------------------------------------------
32768=Missing Data In Business Object
32769=Data Incorrectly Formatted
32770=Data Value Out Of Range
32771=Approval Denied
```

If you want to include other reasons, you can add them to this list—as many as you need. Be sure to save the file as a text file. Users must restart both the Process Server and Process Manager to display the new reasons in the Termination Properties dialog box.

# SAVING A PRIVATE PROCESS

As you develop a private process, it's a good idea to save your work periodically. The first time you save a new private process, Partner Agreement Manager provides a default name (Private1, Private2, and so forth) and stores the private process in the corresponding public process folder.

Each subsequent time you save a private process, Partner Agreement Manager increments the last digit of the version number. You can also save a new version of a private process (Partner Agreement Manager increments the first digit of the version number).

**NOTE:** After a private process has been activated, you must create a new version (or activate another private process in its place) if you want to make any changes to the private process.

**To save a private process:**

1 Choose Save from the Process menu or click the Save Process button in the Command toolbar.

If you're saving a private process associated with a public process, Partner Agreement Manager saves the private process and lists it in the folder for the public process. The default name for the private process is Private1 (the number increments for each new private process you create for the same public process).

If you're saving a private process that is not yet part of a public process, the Save Private Process As dialog box appears.
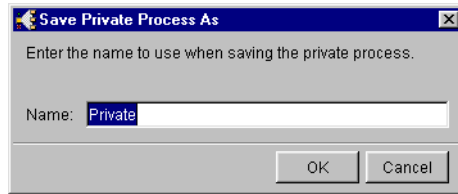


**2** Type a name for the private process and click OK.

Partner Agreement Manager adds the private process to the Private Process Library.

## Saving a new private process version

As you work on a private process, you might want to save your changes in a new version, rather than overwriting an existing version. You might, for example, want to try alternate methods for achieving the same effect.

**To save a new private process version:**

▶ Choose Save As New Version from the Process menu.

- If you're saving a new version of a private process associated with a public process, Partner Agreement Manager increments the first digit of the version number.

- If you're saving a new version of a library process, Partner Agreement Manager increments the version number and adds the new version to the Private Process Library.

## Adding to the Private Process Library

Although you can create a private process specifically for the Private Process Library, you can also add versions of other private processes to the library. You might, for example, want to make actions in an active private process available to be copied into other private processes.

**To add to the Private Process Library:**

**1** Choose Save In Library from the Process menu.

The Save Private Process As dialog box appears.
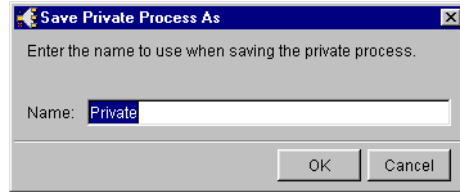


**2** Type a new name for the private process and click OK.

Partner Agreement Manager adds the private process to the Private Process Library.

## Renaming private processes

When you save a private process associated with a public process, Partner Agreement Manager provides a default name (Private1). The name of the process remains Private1, however, the version number increments each time you save the private process. You can change the default name for a private process if you want.

**To rename a private process:**

**1** Right-click anywhere in the Private Process window and choose Properties.

You can also choose Properties from the Process menu, or click the Properties button in the Command toolbar. The Private Process Properties dialog box appears.



Type a name for the private process.

Add a description if you want.
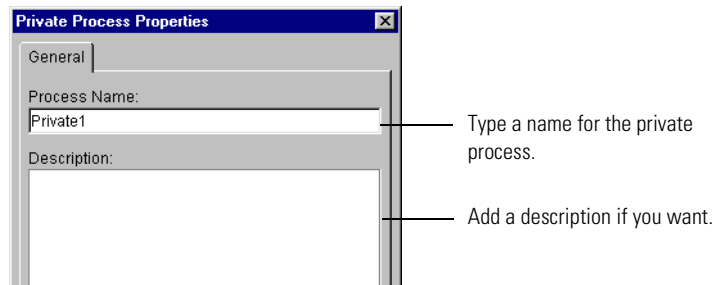
**2** Type a new name and an optional description, and then click OK.

# Inserting a private process

Partner Agreement Manager provides a Private Process Library as a handy way to store private processes that you might want to use for more than one public process. You can create generic private processes and customize them as appropriate, or reuse a private process as is. You can also reuse an inactive private process that is associated with another public process.

**Important:** When you insert a private process, Partner Agreement Manager replaces any actions that might already appear in the Private Process window.

**To insert a private process:**

1 In the Process window, choose the step for which you want to insert a private process.

2 Open the Private Process window, and choose an Insert command from the Edit menu.

- Choose Insert Private Process to insert a private process attached to a public process.

- Choose Insert Library Process to insert a private process stored in the Private Process Library.

The Select Private Process dialog box appears.

Choose a public process...

...a public process version...

... a private process...

...and a private process version.

This dialog box appears when you insert a private process.

This dialog box appears when you insert a library process.

**Select Private Process**

Public Processes:
- Comtech - Configurable Ordering
- Comtech - Escalation Subprocess
- Comtech - PO with Designs

Public Versions:
- 1.0
- 1.1
- 1.2

Private Processes:
- Process Order

Private Versions:
- 1

**Select Private Process**

Processes:
- Update Product Availability

Versions:
- 1

OK    Cancel

**3** Choose the private process you want to insert (including the version number), and click OK.

Partner Agreement Manager inserts the private process you choose, replacing any actions that might already be in the Private Process window.

## ACTIVATING A DIFFERENT VERSION OF A PRIVATE PROCESS

As you develop new versions of a private process, you might want to activate the newer version in place of an older one. You can even activate a new private process for a public process that is already installed. Partner Agreement Manager picks up the new private process the next time it runs the public process.

**To activate a different version of a private process:**

**1** In the Public Process window, right-click the step and choose Activate Private Process.

The Activate Private Process dialog box appears.
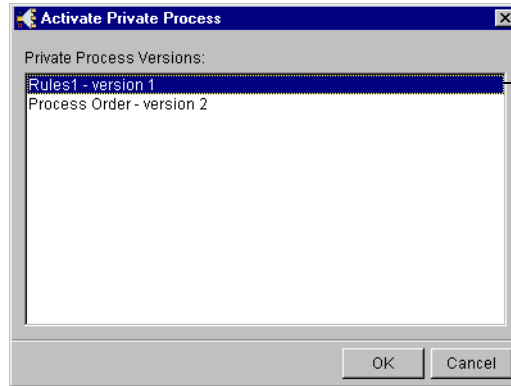


This list contains all versions of the private process that can be activated for this step in the public process.

2 Choose a private process in the list and click OK.

## Verifying a private process

With Partner Agreement Manager, you can check a private process to make sure that there are no errors that prevent the private process from running. Partner Agreement Manager also looks for undefined elements that don't prevent the private process from running, but that might prevent you from getting the expected outcome.

Typical problems include undefined business objects or messages, paths that don't produce a valid output object, undefined or unassigned recipients, undefined wait time, invalid action type, or unspecified subprocesses. Typical warnings include undefined approval subjects, messages, or notification subjects.

After you verify the private process, you can keep the Verify Private Process window open as you continue to work on the private process.

**To verify a private process:**

1 Open the private process and click the Verify Private Process button in the Command toolbar.

You can also choose Verify from the Tools menu.

The Verify Private Process window appears.

The total number of errors appears here.

The total number of warnings appears here.

Icons show the type of element where the error occurs (in an action or in the flow of the private process).

Individual errors and warnings are listed in the order in which they occur.

Click a message to see which action contains the error.

Click to verify the private process.

**Verify Private Process**

Results:

❌ Errors (3)
- ⬜ SendDoc: Select a business  object type for this action, or de
- ⬜ SendDoc: Select a public process path from the available list
- ⬜ The following business objects do not exist: " SONum".  Cha

⚠️ Warnings (0)

[ Verify ]   [ Close ]

**2** To see which action is responsible for a specific error or warning, click the error or warning in the Verify Private Process window.

Partner Agreement Manager highlights the offending action in the Private Process window.

**3** To recheck the private process, click Verify.

As Partner Agreement Manager checks the private process again, it removes any errors or warnings that have been resolved and adds any new ones that have occurred.

**4** Click OK to close the Verify Private Process window.

## ABOUT PRIVATE PROCESS VARIABLES

Depending on the structure of the public process, Partner Agreement Manager creates default variables in the private process to be used for input and output. If the public process includes compound branches, the private process that merges the branching then contains default input variables (I1, I2, I3, and so forth) for each branch that starts in the private process. Likewise, the private process for a step that initiates several branches can contain variables for each possible output from the private process (you can either reuse existing variables or create your own).

For example, this public process has three branches that result in three possible input sets for a private process. Partner Agreement Manager creates a different input variable for each possible input set. Each variable is of the same type as the corresponding output object its path expects.



This step in a public process has three possible outputs.

Reject PO

PO_Rejection.1 -...

Accept PO

Back ordered

Shipped

Backorder_Notic...

Shipment_Notic...

The input variable for this path is of the same type as this business object.

For this step in a public process...

Comtech

...the associated private process has these default input variables.

| Name | Type | Usage | Description |
|------|------|-------|-------------|
| ActiveInputSet | Variant | | |
| I1 | PO_Rejection.1 - Comtech | Input | |
| I2 | Backorder_Notice.1 - Co... | Input | |
| I3 | Shipment_Notice.1 - Co... | Input | |

The purpose of these variables is to provide a handy annotation for input variables in scripts.

# MANAGING PROCESSES

Read this chapter for information about distributing, installing, activating, and deactivating processes.

This chapter includes these sections:

# About managing processes

Although most Partner Agreement Manager processes involve the exchange of information between at least two partners, only one partner is generally responsible for developing a public process. When development is complete, all partners involved must approve the public process, implement their own private processes, test the public process, and then install it.

These are the steps that take a public process from draft status to installation for production.
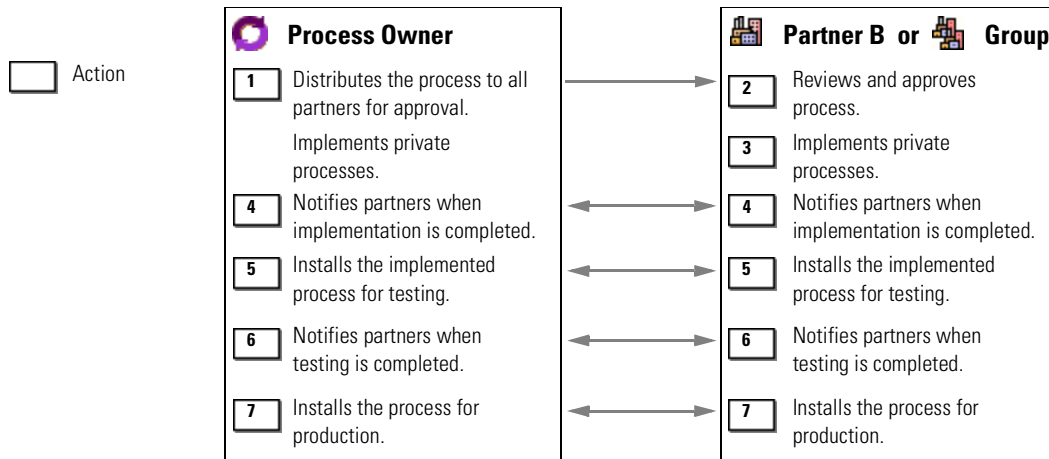


1 You start with the draft process and distribute it to all participating partners for approval.

2 Your partners receive the public process, review it, and either approve or disapprove it.

3 As soon as a partner has approved a process, that partner can implement the process by creating its private processes.

4 After all of the partners have created their private processes, the public process is ready to be installed for testing.

5 Each partner tests it, and when all partners have indicated that the process passes, it is ready to be installed for production.

6 When each partner has activated the process, it is ready to run.

   **NOTE:** If you want to make changes to a process after it has been installed, you can deactivate it, save it as a new version, make any necessary changes, and take it through distribution and approval again.

# About the Process Distribution Manager

Because tracking several processes in varying states of approval and testing can be cumbersome, Partner Agreement Manager provides a Process Distribution Manager to alert you when action is required and to help you take the appropriate action at each step in the distribution-to-activation process.

A check mark shows that a stage is completed for all partners involved.

Click a plus sign (+) to display the names of participating partners or partner groups.

Click to apply the current action.

Click to view the state of this process for all partners.



This list shows all actions that are currently available for the chosen state.

Gray type shows states that are not yet available.

Bold type shows all states that are currently available.

This area displays the status for an applied action or a description of the action chosen.

At any point in the distribution-to-installation process, you can open the Process Distribution Manager and review the status of the process. Partner Agreement Manager uses check marks and hourglasses for the various states to indicate status.
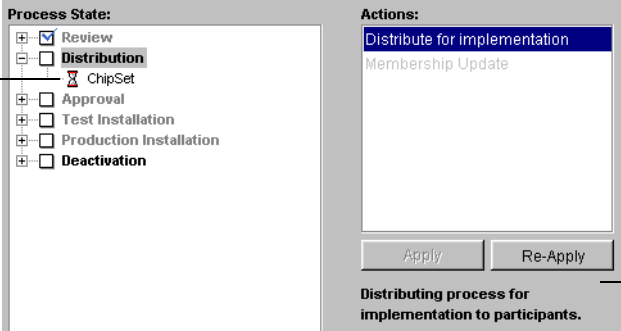
A check mark for the top-level state shows that all partners involved have reached the state.



A check mark beside the name of a group indicates that all group members have reached the state.

A check mark beside the name of a group member indicates that the group member has reached the state.

An hourglass indicates that Partner Agreement Manager is waiting for the partner to receive a message or the process it is attempting to distribute.

An hourglass indicates that Partner Agreement Manager is waiting for the partner to receive a message or distributed process.



The status area shows the step Partner Agreement Manager is attempting.

You can also click the Partner State button to view each partner's status for the process.

The Partner State dialog box shows the current state of the process for each partner.

Click to display the status for partner group members.



The partner icons indicate each partner's role in the process.

These are the states that a process can pass through and the actions available for each state.

| In this state | You can |
| --- | --- |
| Review | Send the process to all participants so that they can review it. It's a preliminary step you can take before starting the formal distribution of a process. Because no action is required from your partners, a process sent for review does not appear in your partners' Processes Manager folder. |
| Distribution | Distribute the process to all participants so that they can approve it and start implementing it. If you add a member to a partner group after you distribute a process, you can also update the group membership. See *Distributing a process* on page 208.<br><br>All participating partners can begin implementing the process as soon as they receive it. They cannot begin testing it until all partners approve and implement it. See *About implementing a process* on page 212. |
| Approval | Approve the process. All partners must approve the process before you can begin testing it. See *Approving a process* on page 210. |
| Test Installation | Install the process for testing purposes. The process can be run in test mode, but not in production mode. After you install a process for testing, you can suspend it. If a process has been suspended, you can resume testing it. See *Installing a process for testing* on page 212. |
| Production Installation | Install the process for production. The process can now be run in production mode. After you install a process for production, you can suspend it. If a process has been suspended, you can resume running it. See *Activating a process* on page 214. |
| Deactivation | Deactivate the process (prevent it from running). You can either deactivate the process immediately (all running instances of the process are halted and no new instances can start), or request deactivation from the partner who owns the process. See *Deactivating a process* on page 216.<br><br>Note that deactivating a process immediately is a very extreme step that should be reserved for only the most severe circumstances. |

**To view the Process Distribution Manager:**

▶ In the Process Manager window, right-click the process and choose Process Distribution Manager from the menu that appears.

You can choose a process in a partner's Processes folder or in the Processes Manager folder.

▶ Open the Public Process window and click the Process Distribution Manager button in the Command toolbar.

You can also choose Process Distribution Manager from the Tools menu.

# Distributing a process

After you create a process, you distribute it to your partners for their approval. You might want to send copies of the process to your partners in advance, so that you can review and discuss it informally before actually starting the formal approval cycle (this is the Review state).

**To distribute a process:**

**1** Open the Process window and click the Process Distribution Manager button in the Command toolbar.

You can also right-click the process in the Process Manager window and choose Process Distribution Manager from the menu that appears. The Process Distribution Manager appears.

Distributes a review copy of a process to partners.

Click to distribute the process to all partners for their approval.



Click to apply the highlighted action.
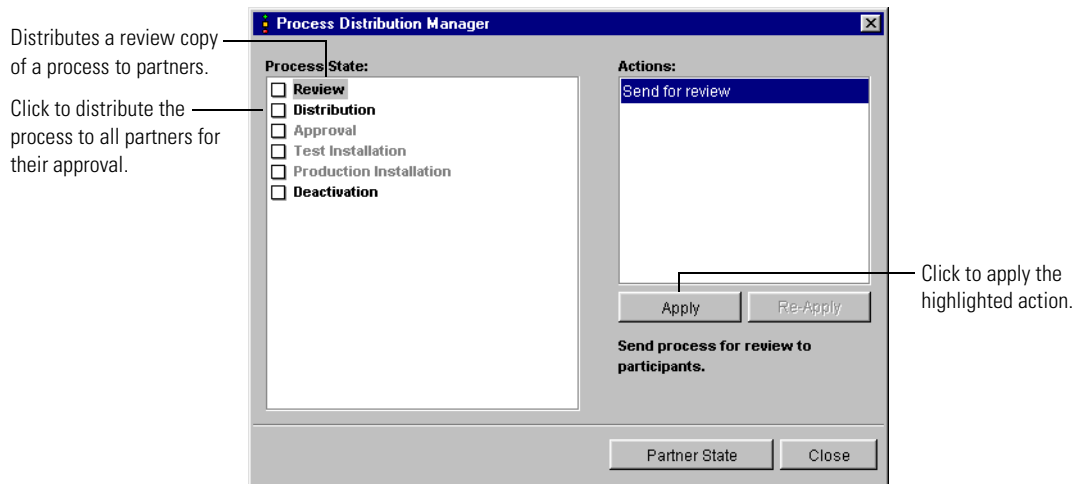
**2** Choose the action you want to take and click Apply.

| To | Choose this state | And choose this action |
|---|---|---|
| Distribute a review copy of the process as a courtesy to partners | Review | Send for review |
| Distribute the process to partners so that they can install and test it | Distribution | Distribute for implementation |
| Update distribution after adding or removing partner group members | Distribution | Membership Update |

The Process Distribution Manager shows the new state for the process.

A check mark shows that the process has been distributed to all partners involved.

This is the status for the current action.

The distributed process appears in this folder.

Choose this state to cancel the distribution. If you deactivate a process, you must save a new version before you can distribute it again.
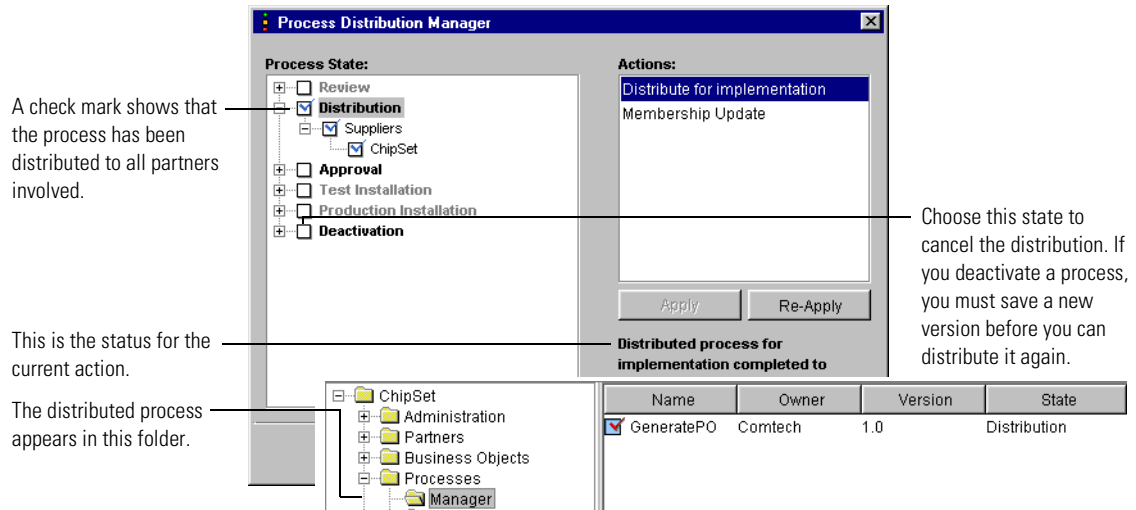
After a process has been distributed and before it is installed, you can effectively cancel the distribution by deactivating the process. If you do so, Partner Agreement Manager notifies your partners that the process has been deactivated, and your partners can no longer approve the process.

If you want to redistribute a deactivated process, you must first save a new version of the process and then distribute the new version.

WARNING: Deactivating a process that has started running (it has been installed for testing or production) can have severe consequences. This extreme step should be reserved for only the most severe circumstances.

ABOUT LOCAL OR ONE-STEP PROCESSES

A local or one-step process is one in which you are the owner and sole partner. You might create a local process to perform some internal data manipulation, to get data from an enterprise system, or to act as a controlling process to call a subprocess.

Because you are the only partner involved, distributing and installing a local process is streamlined. You begin by distributing the process just as you do any process. Partner Agreement Manager immediately marks the process as approved, and you can then implement it by adding the appropriate private processes. The next stage is installation for testing, followed by installation for production.

Notice that there are no plus signs (+) because there are no additional partners involved.
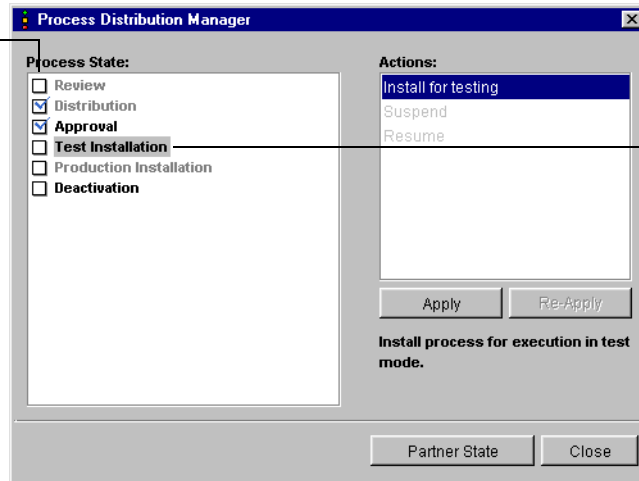
After you distribute and implement a local process, you can install it for testing.



## Approving a process

The next step in the distribution of a public process is approval. After the process owner distributes the process, it becomes an action item for all of the other partners. By the time a process gets to the approval stage, you and your partners will have undoubtedly discussed it at length, but it's always a good idea to review it carefully before giving your approval.

**Note:** Only the partners who receive a process must approve it. Partner Agreement Manager approves the process for the partner who distributed it.

**To approve a process:**

**1** Choose the Manager folder in the Process Manager window and double-click a process awaiting action to open it.



Choose the Manager folder to display all processes that are awaiting action.

Double-click a process to open it.
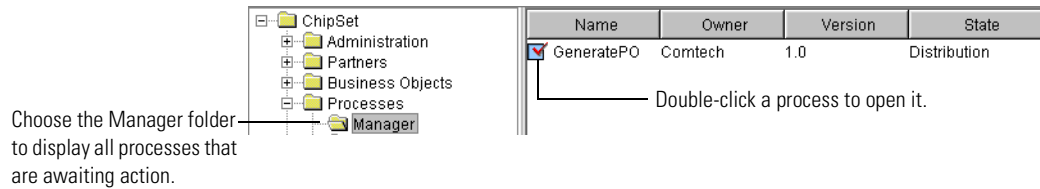
**2** Review the process as necessary.

**3** Click the Process Distribution Manager button in the Command toolbar.

You can also right-click a process in the Process Manager window and choose Process Distribution Manager from the menu that appears.

The Process Distribution Manager appears.



After you review the process, you can choose Approval.
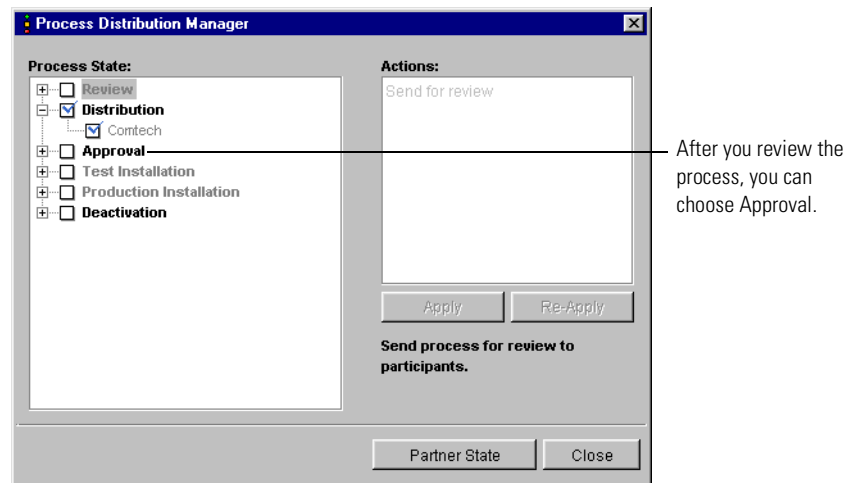
**4** If the process is acceptable, choose the Approval process state and the Approve action. Click Apply.

A check mark appears for the Approval state, and the Test Installation state becomes available.

The check mark indicates that the process has been approved by all partners.



The status area indicates that you have approved participation in the process.
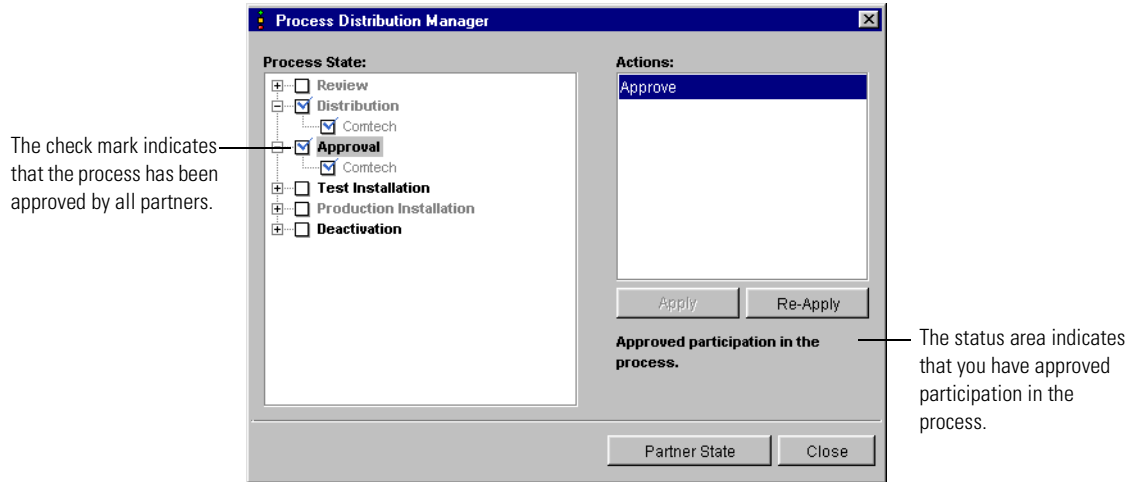
The state displayed in the Process Manager window changes to Approval.

The state displayed in the Process Manager window is Approval.
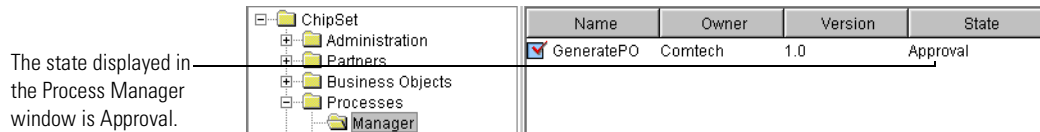


### About implementing a process

Before you can progress to the next state—Test Installation—you must create, verify, and save a private process for each of your steps in the public process. See *Creating a new private process* on page 150.

**NOTE:** Implementation can also include developing any necessary interfaces with external applications using special extension actions. See the *Partner Agreement Manager Adapter Developer's Guide* for information about developing adapters to interface with external systems.

## Installing a process for testing

When you install a process for testing, all partners in the process can participate in test mode. This assumes that you have created the appropriate test environment as part of your Partner Agreement Manager implementation.

**NOTE:** The partner who initiates a process installs it for testing after all other partners have done so. When using partner groups, the initiator can install for testing as soon as any member of each partner group has installed for testing.

During the testing phase, all partners in the public process—starting with the partner who initiates the first step—run the process to make sure that it performs correctly. At the end of a successful testing phase, each partner indicates that the process passes the testing milestone by installing it for production. At any point during the testing phase, any partner can suspend the process. See *Suspending and resuming a public process* on page 233.

**To install a process for testing:**

1 Choose the Manager folder in the Process Manager window and open a process awaiting action.

2 Click the Process Distribution Manager button in the Command toolbar.

You can also right-click a process in the Process Manager window and choose Process Distribution Manager from the menu that appears.

Check marks in the Process Distribution Manager show the current state of the process.
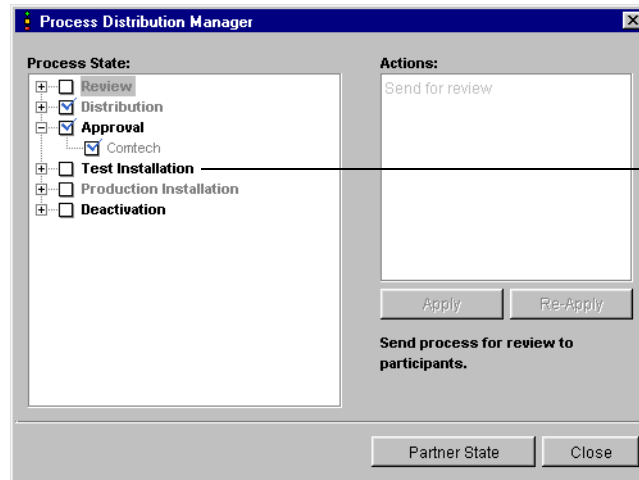


After all partners have installed a process for testing, you can begin testing as a group.

3 Choose Test Installation in the Process State list and choose Install for Testing in the Actions list. Click Apply.

A check mark appears for the Test Installation state, and the Production Installation state becomes available. Also, the Suspend action becomes available.

The check mark indicates that all partners have installed the process for testing.

| Process State: | Actions: |
|---|---|
| ☐ Review | **Install for testing** |
| ☑ Distribution | Suspend |
| ☑ Approval | Resume |
| ☑ **Test Installation** | |
| ☑ Comtech | |
| ☐ **Production Installation** | |
| ☐ **Deactivation** | |
| | Apply     Re-Apply |
| | **Installed process for execution in test mode.** |
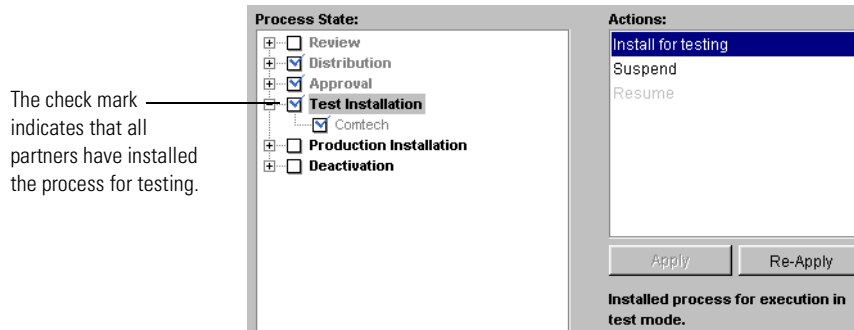
Partner Agreement Manager requires the owner of the process to be the last partner to install the process for testing.

**TIP:** To test a process, the partner assigned to the first step must start it. To start a public process, right-click it in the Processes folder in the Process Manager window, and choose Start from the menu that appears.

## Activating a process

After all partners successfully test the process, you're ready to install it for production and begin using it. The partner who owns the process is the last one to install it.

**To activate a process:**

**1** Choose the Manager folder in the Process Manager window and open a process awaiting action.

**2** Click the Process Distribution Manager button in the Command toolbar.

You can also right-click a process in the Process Manager window and choose Process Distribution Manager from the menu that appears.

Check marks in the Process Distribution Manager show the current state of the process.
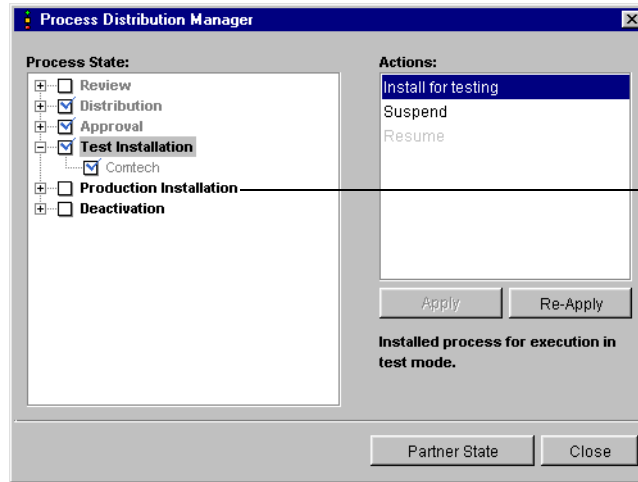


After testing is complete, you can choose Production Installation.

**3** Choose Production Installation in the Process State list and choose Install for Production in the Actions list. Click Apply.

A check mark appears for the Production Installation state. Also, the Suspend action remains available.
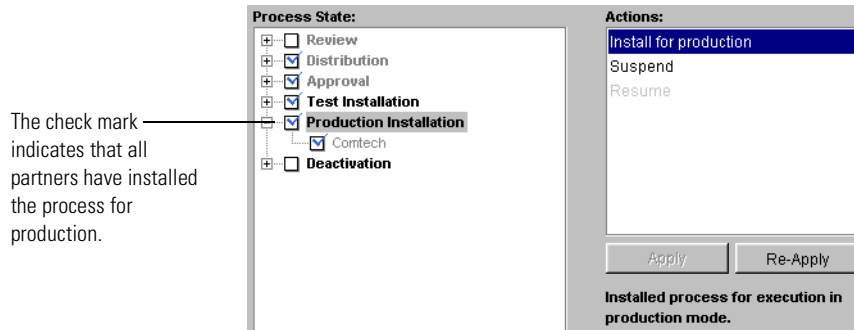


The check mark indicates that all partners have installed the process for production.

# Deactivating a process

After you begin distributing a process, you can deactivate it at any point. Deactivating a process during the early stages—before you install it for testing or production—is risk-free. Before the process starts running, deactivating it recalls it from distribution and no further action can be taken. If you want to resume the distribution of a deactivated process, you must first save the process as a new version and distribute the new version. You might, for example, want to make minor changes to the process or refine it in some way.

**Warning:** Deactivating a process immediately is a very extreme step that should be reserved for only the most severe circumstances.

After you have installed a process for testing or production, deactivating it can have severe consequences, depending on which deactivation action you choose.

The two deactivation actions are:

- Request Deactivation

  Requesting deactivation is similar to suspending a process and is the more benign of the two deactivation actions. When you request deactivation, Partner Agreement Manager notifies the process owner, who relays the request to all partners who might initiate the process. When all partners acknowledge the request, and when all running instances of the process finish, the process is deactivated.

- Deactivate Immediately

  Deactivating a process immediately stops the process, and no further interaction between partners occurs. If an instance of the process is running, Partner Agreement Manager continues to list it as a running instance until you stop and restart the Process Server. You might use this action if your association with a partner terminates abruptly.

**To deactivate a process:**

1 Choose the Manager folder in the Process Manager window and open the process you want to deactivate.

2 Click the Process Distribution Manager button in the Command toolbar.

You can also right-click a process in the Process Manager window and choose Process Distribution Manager from the menu that appears.

The Process Distribution Manager shows the current state of the process.
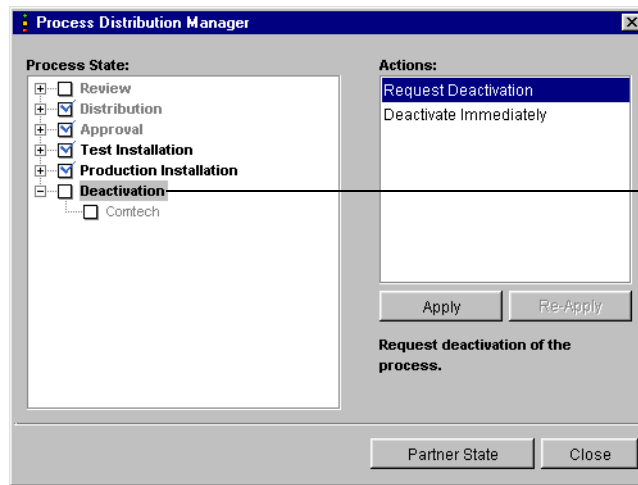


Any partner in a process can deactivate it.

**3** Choose Deactivation in the Process State list and choose either Request Deactivation or Deactivate Immediately in the Actions list. Click Apply.

Partner Agreement Manager notifies partners as necessary and deactivates the process. A check mark appears for the Production Installation state. Also, the Suspend action remains available.

**NOTE:** The owner of a process that has been deactivated must save the process as a new version before it can be distributed again.

## SHOWING PARTNER STATE FOR A PROCESS

In a single convenient location—the Partner State window—Partner Agreement Manager lets you view the current implementation and execution state of a process for all partners involved. You can leave the Partner State window open as you work and refresh its contents whenever you want.

**To show partner state for a process:**

**1** Right-click the process in the Process Manager window and choose Process Distribution Manager from the menu that appears.

You can also open the Public Process window and click the Process Distribution Manager button in the Command toolbar. The Process Distribution Manager appears.
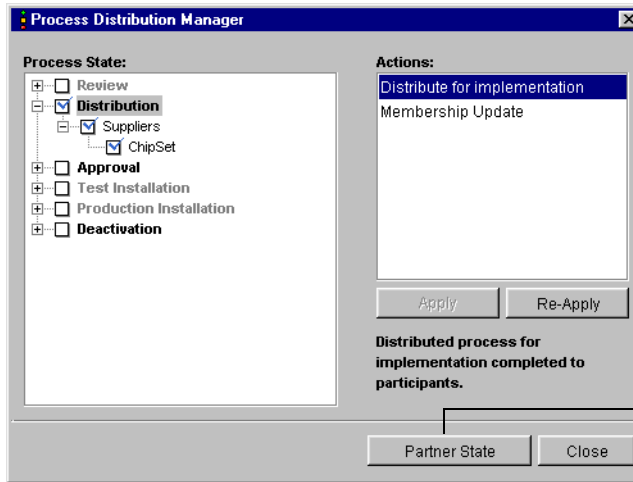


Click to view the state of this process for all partners.

**2** Click Partner State.

The Partner State window appears. Partner Agreement Manager displays the current state of this process for each partner.



Click to display the status for partner group members.
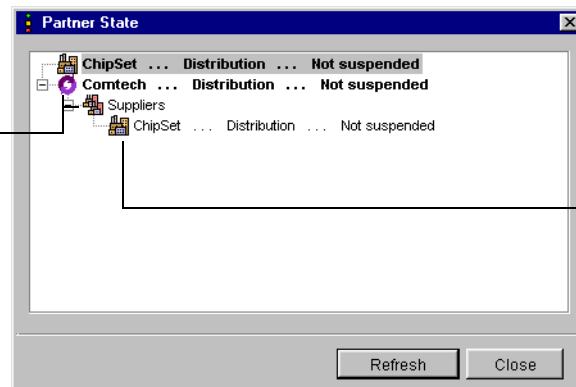
The partner icons indicate each partner's role in the process.

**3** To update state information, click Refresh.

Partner Agreement Manager updates the Partner State window to display any information that has changed.

# Starting a process

There are three ways to start a process. You can:

- start it manually.
- schedule it to start at a specific time.
- set it to be triggered by an event.

When you start a process manually or schedule it, you can choose the execution mode it runs under: Test or Production. If the process is installed for testing, Test is the only mode available. If it's activated for production, you can run it in either test or production mode.

## Starting a process manually

Partner Agreement Manager makes it easy to start any installed process directly from the Process Manager window. When you start a process, you choose an execution mode. You can also specify a value for any variant input variable defined in the process. And if the process includes a partner group, you specify which member of the group is participating in this instance of the process. The process then runs just as it would if it had been scheduled or triggered by an event.

**Note:** You can only choose a variant variable as input for a process, not a business object variable.

**To start a process manually:**

1 Open the Process folder in the Process Manager window and navigate to the process you want to start.

2 Right-click the process and choose Start from the menu that appears.

The Start Process dialog box appears.



Specify an execution mode.
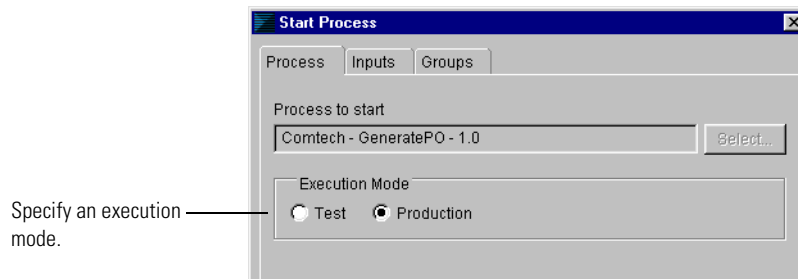
**3** Select the appropriate execution mode (test or production).

If a process is installed for testing, Test is the only available mode.

**4** If there are input variables defined for the process, click the Inputs tab.

The Inputs tab appears.



Type a value for the chosen input variable.
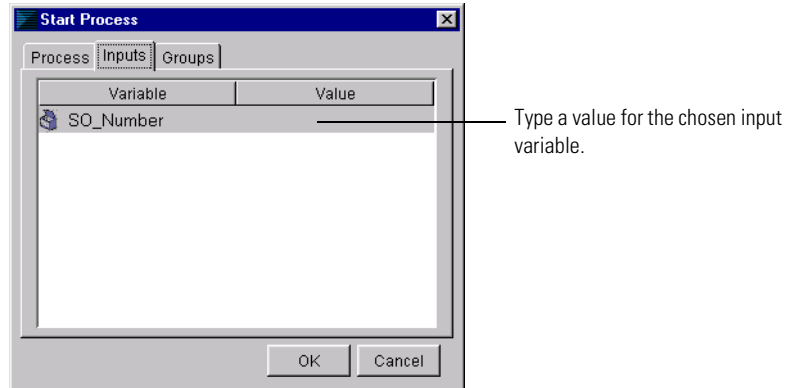
**5** Choose a variable, and type a value for it in the Value column.

The value you enter becomes the input value for that variable at the start of the process.

**6** If a partner group has been defined for the process, click the Groups tab to specify which partner is participating.

The Groups tab appears.



Choose a partner name.
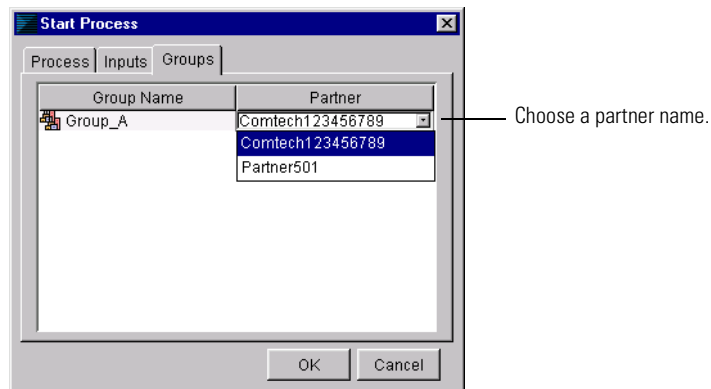
**7** Choose a partner from the Partner list for the group.

**8** In the Start Process dialog box, click OK.

Partner Agreement Manager starts the process using the input values you specify, and directs it to the partner you specify.

## Scheduling a process

With the Partner Agreement Manager scheduling feature, you can start a process at a specific time. You can also set a process to run on a limited or open-ended recurring schedule. You can set a process to repeat by year, month, week, day, hour, or even minute-by-minute.

When you schedule a process, you also select the mode it runs under (test or production).

**To schedule a process:**

1 Choose New from the Actions menu, and then choose Process Schedule.

The Process Scheduler appears.



Choose a process to be scheduled.

After you choose a process, you can specify the mode it runs under.
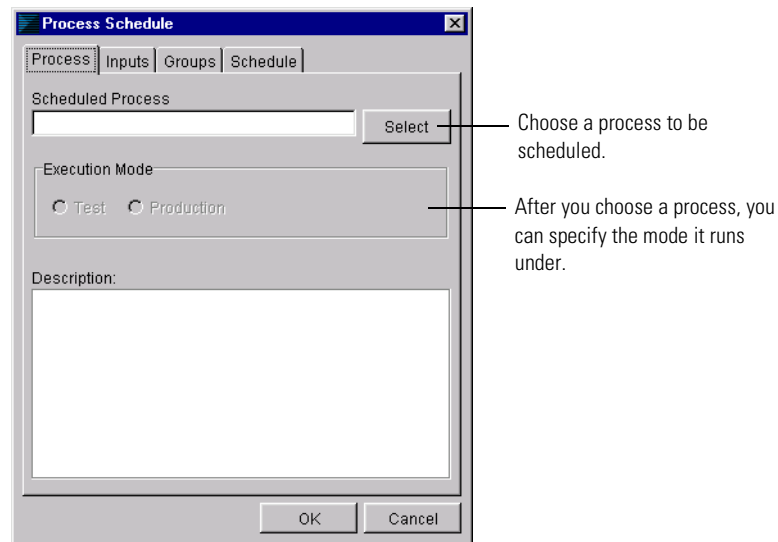
2 Click Select to choose the process you want to schedule.

The Select Process dialog box appears.



This list shows all installed processes for which you have the first action.

This list shows all installed versions of a process.
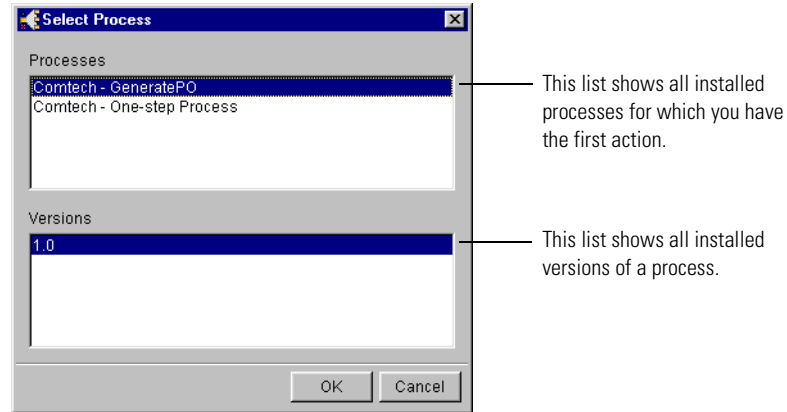
**3** Choose the process you want to schedule, choose the version, and click OK.

**4** Choose the execution mode for the process.

If the process is installed for production, you can select either test or production mode. If the process is installed for testing, Test mode is the only one available.

**5** If there are input variables defined for the process, click the Inputs tab.

The Inputs tab appears.



Choose an input variable.

Type a value.
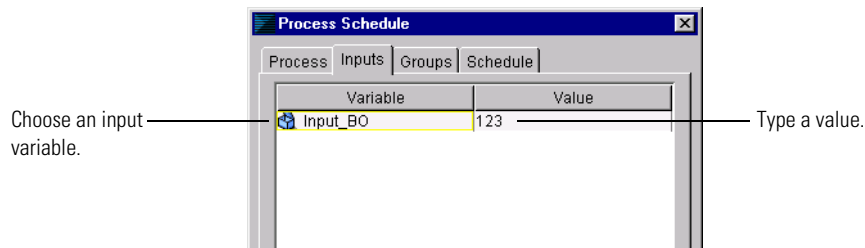
**6** Choose a variable and enter a value for it in the Value column.

The value you enter becomes the input value for that variable at the start of the process.

**7** If a partner group has been defined for the process, click the Groups tab to specify which partner is participating.

The Groups tab appears.



Choose a partner name.
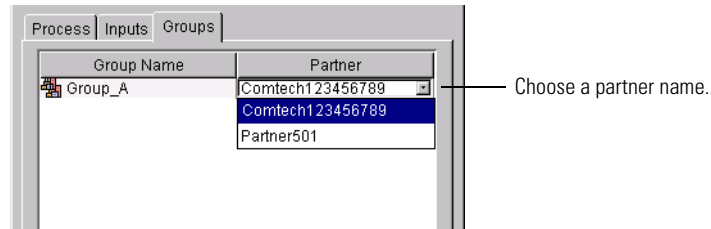
**8** Choose a partner from the Partner list for the group.

**9** Click the Schedule tab to set start parameters and frequency.

The Schedule tab appears.



Set the starting date and time.

Type an interval to repeat the process (0 does not repeat).

Set the process to repeat continuously or a set number of repeats.

Type the number of times to repeat the process.
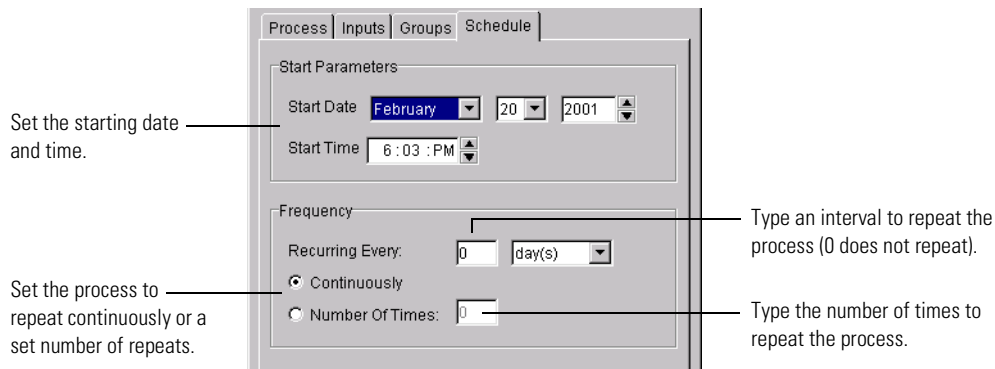
**10** Select the month, day, and year for the start date, and set the start time.

**11** Set the process to repeat, if necessary.

| To repeat a process | Do this |
|---|---|
| One or more times | **1.** Set a Recurring Every interval (for example, 3 days, or 5 minutes). <br> **2.** Click Number of Times, and enter the number of times you want to repeat the process. |
| Once only (no repeat) | **1.** Set the Recurring Every interval to 0. <br> **2.** Click Number of Times, and enter 1 as the value. |
| At regular intervals | **1.** Set a Recurring Every interval (for example, 3 days, or 5 minutes). <br> **2.** Click Continuously. |

**12** In the Process Schedule dialog box, click OK.

# About events

One of the ways you can start a Partner Agreement Manager process is to trigger it when information arrives from an external source. The information might be from an enterprise system, a web server, or from some other source. You might, for example, want to start a Partner Agreement Manager process when a purchase order is entered into an enterprise system, or when inventories fall below a certain level.

You can use Partner Agreement Manager to define different types of events that can trigger processes. After you create an event type, you can register processes with the event so that Partner Agreement Manager starts the processes when an event of this type occurs.

For each event, you specify the type of data it passes to triggered processes, the list of processes triggered by events of that type, and the input variable in each process that receives the event data. You can also specify how to resolve events of that type if a pending process is suspended when the event occurs.

Then, when an event occurs, Partner Agreement Manager triggers an instance of each process you registered for that event type. The input variables you specify receive the event data.

Partner Agreement Manager stores event types in an Events folder located in the Processes folder in the Process Manager window. This folder contains both predefined events and the ones you create.



This folder contains both predefined events and the ones you create.
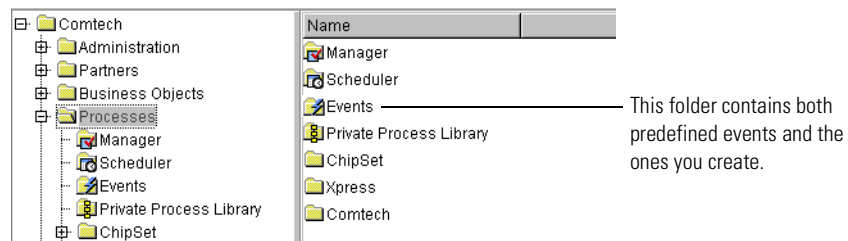
## How events enter Partner Agreement Manager

Events can enter Partner Agreement Manager in two ways:

- through a method that is built into adapters and that polls an external system by way of the Adapter Server.

- through a socket in the Process Server that is set to listen for events.

When you build an adapter, you can include a checkForEvents method that polls an external system at regular intervals for events. In order for polling to be active, the Adapter Server and the adapter must both be running, and polling must be enabled. When polling is active, Partner Agreement Manager generates events according to the criteria defined in the checkForEvents method.

With polling, the adapter polls the business application for events.
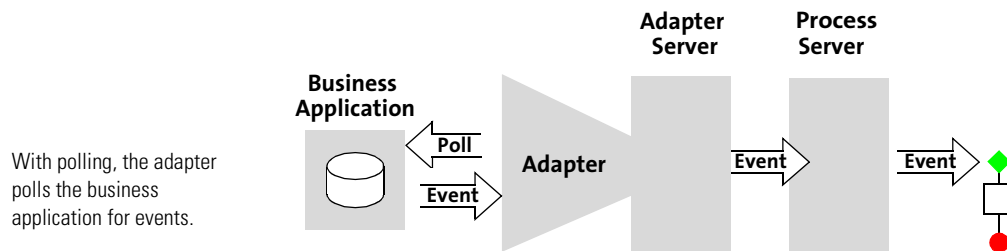


Another way to send an event to the Process Server is to use the HTTP POST mechanism to push an event into the Process Server. See *Using event push* on page 229.

See the *Partner Agreement Manager Adapter Developer's Guide* for information about adapters.

## Creating an event type

When you create an event type, you can register the processes to be triggered at the same time, or you can create event types and return to them later to register processes. After you create an event type, you can register or unregister processes to be triggered, or change the input variable binding, but you can't change the event type name or data type.

**To create an event type:**

1 Choose New from the Actions menu, then choose Event.

The Create New Event Type dialog box appears.



Select the type of data this event will contain.
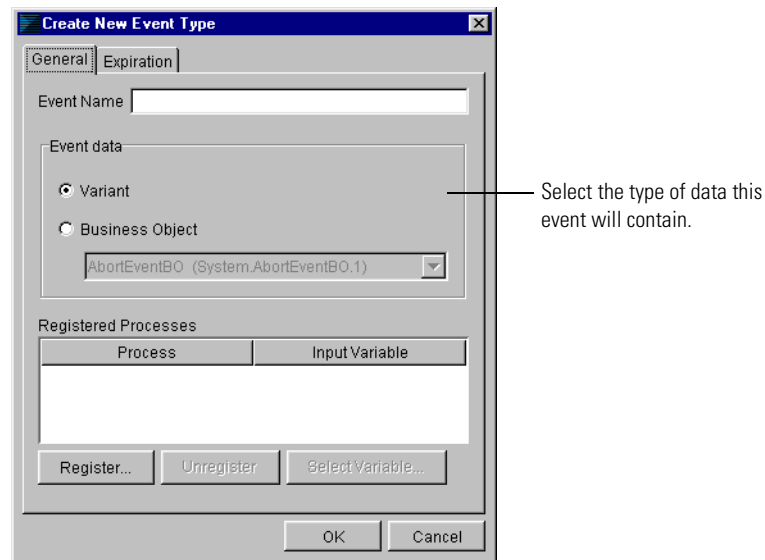
2 Type a name for the event and select the type of data that this event will pass to Partner Agreement Manager.

3 Click Register to choose the processes to be triggered by this event type.

The Register Process dialog box appears.



Choose a process to be triggered by this type of event.

Choose a process version.
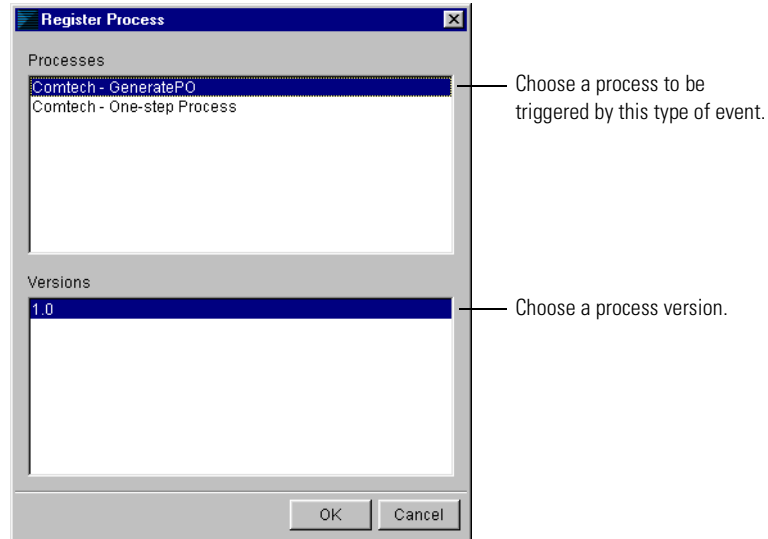
**4** Choose a process and a version, and then click OK.

To trigger additional processes, you can add them to the Registered Processes list. To remove a process, choose it and click Unregister.

**5** In the Create New Event Type dialog box, choose a registered process and click Select Variable to specify the input variable that will receive event data when the process is triggered.

The Select Variable dialog box appears. It lists the variables for the selected process that match the type of data you specified for the event.



Variables in this list match the type of data specified for this type of event.
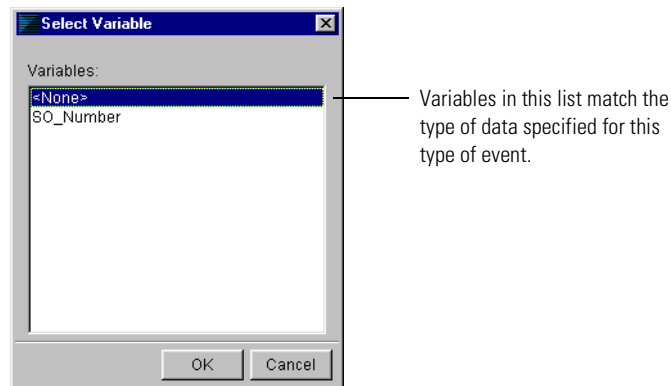
**6** Choose a variable and click OK.

You can choose an input variable for each process in the list. To change the binding to an input variable, choose the process and click Select Variable.

**7** In the Create New Event Type dialog box, click the Expiration tab to set time-out properties for the event.

The Expiration tab appears.



These settings specify the action to take if a process triggered by the event type is suspended.

**8** Select an expiration setting.

The expiration specifies the action to take if a process triggered by the event has been suspended. For more information on process suspension, see *Suspending and resuming a public process* on page 233.

| Use this setting | To: |
| --- | --- |
| No expiration | Wait indefinitely for the process to be resumed, and then trigger an instance of that process with event data. |
| Expires immediately | Do not trigger a process if it is suspended. |
| Expires in | Wait for the specified period of time for the process to be resumed. You can specify a combination of days, hours, and minutes. Disregard the event after it expires. |

**9** Click OK.

# Using event push

Partner Agreement Manager includes an event push feature, which uses the HTTP POST mechanism to push events into the Process Server and start an Partner Agreement Manager process.

When you use event push, a port on the Process Server is set to listen for events in the form of HTTP POST messages. The default port is 8486. When a message is detected, Partner Agreement Manager uses the information in the message to create an event.

With event push, a socket on the Process Server listens for events.
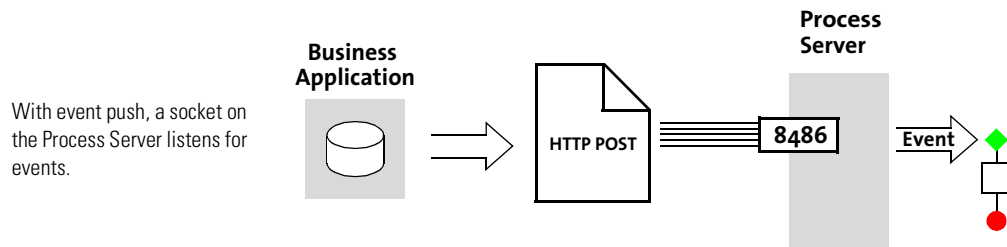


**NOTE:** If you have already assigned port 8486 or reserved it for another purpose, you can change the listening port by adding a setting to your Partner.properties file in the WebSphere PAM Partners folder for your specific partner. The format is:

```
CRStation.ExternalEvent.Port=port
```

Partner Agreement Manager reads an HTTP POST message off the socket, responds with a 200 OK (and additional data to help in debugging and for browser push) or a 400/401/404 error message, and then closes the socket.

The HTTP POST message must be in this format:

```
POST (Whatever/)Partner Agreement ManagerEvent HTTP/1.0
Content-Type: application/x-www-form-urlencoded
Content-Length: length
Authorization: Basic base64-encoded(userID:password)


URL-encoded-event-content


event-content contains the following fields:


event-name=name of the event (aka event type id)
event-id=ID of the event (Ignored - for future use)
event-source=The event source (string)
event-mode=test | production
data=variant or business object
```

For more information, see the HTTP specification.

## TESTING A PUSHED EVENT

With the following HTML form, you can use event push in a test environment. The form creates an HTTP POST message that goes to the default event-push listening port on your Process Server. When it receives the POST message, the Process Server prompts the browser to input the Partner Agreement Manager user ID and password. If the user ID and password are valid, the POST message is accepted and the POST message generates an event.



Enter the appropriate information in these fields and click Push Event.

This is the HTML code that generates the event tester form:

```
<HTML>

<Head>
<b>EventTester</b>
</Head>

<Body>
<Form method="POST" action="http://Processserver:8486/
Partner_Agreement_ManagerEvent">
   Event Name <Input name="event-name">
   <P>
   Event ID (Ignored) <Input name="event-id">
   <P>
   Event Source <Input name="event-source">
   <P>
   Event Mode <Select name="event-mode">
       <Option value=test>Test Mode
       <Option value=production>Production Mode
   </Select>
   <P>
   Event Data <TextArea name="data" rows=3> </TextArea>
   <P>
   <center>
     <Input type="submit" value="Push Event">
   </Center>
</Form>
</Body>

</HTML>
```

**NOTE:** If you use this code to create an event tester, be sure to enter the name of your Process Server in the **Form method** line. If you use a port other than the pre-set one, enter the correct port number.

# Using a process abort event

When a process ends unexpectedly, Partner Agreement Manager produces a predefined event, Process Abort Event, to indicate an unexpected termination. You can use this event as the trigger for any error handling processes you might want to run. For example, you might want to run another process that sends e-mail notifications to individuals based on the nature of the problem. Or you might want to run another process that accomplishes other tasks.

When a Process Abort Event occurs, it contains an AbortEventBO business object, which in turn contains fields for the process name, process owner, process version, process ID, and aborting partner. You can use the values in any of these fields to trigger actions in any registered process (a process triggered by the event).

**To use a process abort event:**

1 Open the Events folder and double-click the Process Abort Event.

The Event Process Registration dialog box appears.



Choose a process to be registered.

Choose a process and click to unregister it.
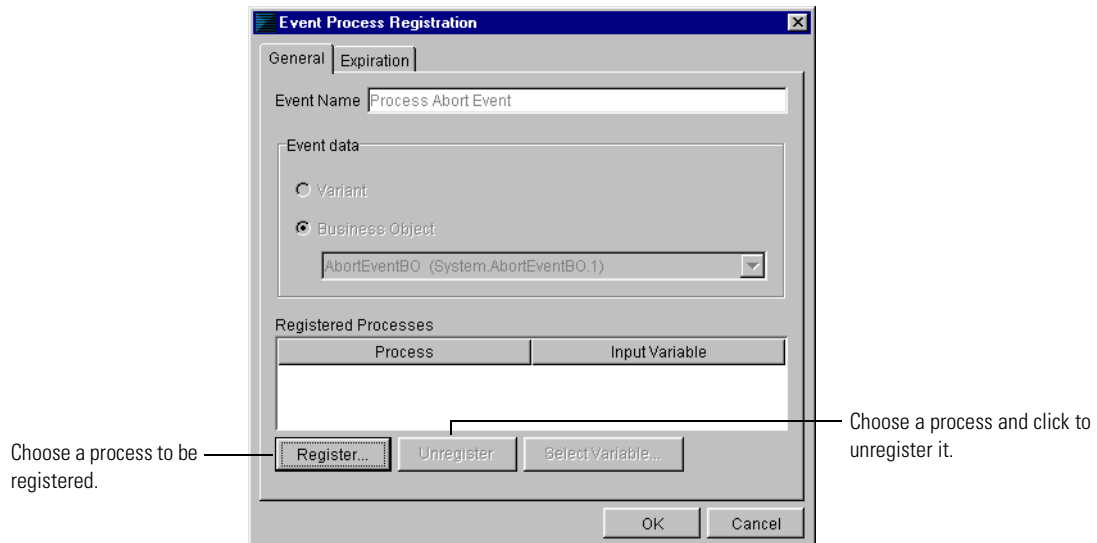
2 Click Register and choose the processes to be triggered by the abort event.

3 For each process, select the input variable that will be initialized with the AbortEventBO.

To select the variable, first select the process in the Event Process Registration dialog box, click Select Variable, choose an input variable in the dialog box, and then click OK.

**4** Click OK in the Event Process Registration dialog box.

# Suspending and resuming a public process

After you install a process for testing or production, the Suspend action becomes available. Suspending a process means that no new instances of the process can be started. Instances of the process that are currently running continue to run until they are completed.

As soon as you request suspension, Partner Agreement Manager immediately notifies the process owner. The owner, in turn, automatically relays the request to all partners who might initiate the process. When all initiating partners acknowledge the request, and when all running processes finish, the process is suspended. Until that point, the Apply button remains available to indicate that the process is not yet suspended.

**NOTE:** Partner Agreement Manager manages all of this communication automatically. No additional action is required from partners.

When a process has been suspended, the partner who suspended it can ask that it be resumed. If more than one partner suspends a process, each partner must resume the process before it can be run again. Again, all partners must be notified before an instance of the process can actually be started. If a partner attempts to start a suspended process, Partner Agreement Manager alerts the partner that another partner has suspended the process.

**To suspend a process:**

**1** Right-click the process in the Process Manager window and choose Process Distribution Manager from the menu that appears.

You can also open the Public Process window and click the Process Distribution Manager button in the Command toolbar. The Process Distribution Manager dialog box appears.

**2** Choose Test Installation or Production Installation in the Process State list and choose Suspend in the Actions list. Click Apply.

Partner Agreement Manager indicates that it is suspending process execution.



The Suspend action remains available until the process is successfully suspended.

The status area indicates that Partner Agreement Manager is suspending the process.
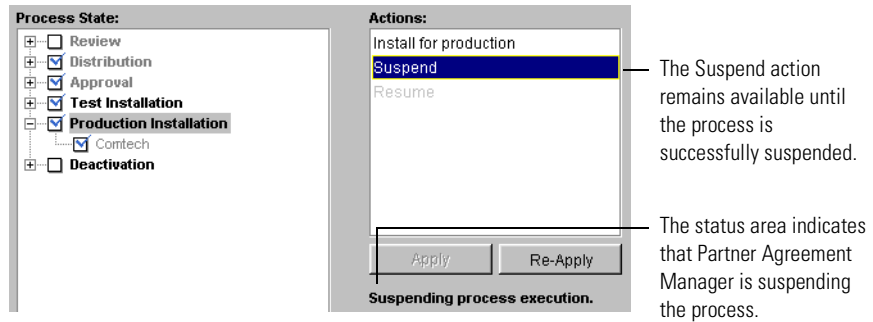
**TIP:** If there seems to be a long delay between when you ask to suspend a process and when it actually stops running, you might want to display the partner state to see each partner's current status. If necessary, you might want to contact a partner off-line to coordinate suspending a process.

If an error occurs while transmitting your suspend or resume request to your partners, Partner Agreement Manager lets you reapply the request.

**To resume a suspended process:**

▶ Click Resume in the Process Distribution Manager.

If you have suspended a process, you must resume it—a partner can't clear your suspension. All partners who suspend a process must resume it before the process can start running again. To determine if additional partners have also suspended the process, open the Process Distribution Manager for the process and click the Partner State button. See *Showing partner state for a process* on page 217.

# A

# Notices

This information was developed for products and services offered in the United States. IBM may not offer the products, services, or features discussed in this information in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this information. The furnishing of this information does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the information. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this information at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you. Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM United Kingdom Laboratories,
Mail Point 151,
Hursley Park,
Winchester,
Hampshire,
England
SO21 2JN.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

# Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AIX
DB2
IBM
MQSeries
SupportPac
WebSphere

Pentium is a registered trademark of Intel Corporation in the United States and/or other countries.

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

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

# GLOSSARY

**action**—a task performed as part of a private process. A private process action is the equivalent of a step in a public process. See the following terms in this glossary for more information about the action types you can include in a private process:

- approval action
- extension action
- mapping action
- notification action
- output object action
- script action
- subprocess action
- termination action
- timer action

See also *private process.*

**adapter**—the software bridge between Partner Agreement Manager processes and specific end-system and business-application interfaces. Adapters manage interactions between business applications and the Adapter Server. They allow private processes to interact with external business applications while a process is running, and they allow PAM to start public processes based on events that occur in external business applications. See also *adapter implementation*, *adapter instance*, *adapter type.*

**adapter implementation**—the implementation declaration for an adapter type. It specifies the name and location of the Java source file that defines the application logic used to communicate with a specific end system through that end system's interface. The application logic is specified in the form of properties. See also *adapter*, *adapter instance*, *adapter type*.

**adapter instance**—an instance of an adapter implementation. The adapter instance is used in a private process extension action and provides the specific values to be used for the properties declared in the adapter implementation. See also *adapter*, *adapter implementation*, *adapter type, extension action*.

**adapter type**—a definition that is stored in XML format and specifies the adapter's properties as well as the operations and events it supports. A single adapter type can have multiple implementations, and each implementation can have multiple instances. See also *adapter*, *adapter implementation*, *adapter instance*.

**approval action**—a private process action that you use to ask for a response from a user before letting the process continue to run. You can use an approval action, for example, to ask for an OK when a purchase order exceeds a predetermined amount. See also *private process*.

**business object**—a message transmitted as part of a public process. Business objects take the form of purchase orders, acknowledgments, requests for clarification, and so on. See also *business object type*.

**business object type**—a definition that determines the types of information a message can contain. It has three properties: the top-level element in its element definition set, its key field, and whether instances of it return audit information for non-repudiation purposes. The name of the business object type is the name of the element you select as its top-level element. See also *business object*, *element definition set*, *non-repudiation*.

**business object variable**—one of the two types of variables used in Partner Agreement Manager to store information within a process. Business object variables create an instance of a business object type. They can be used to store, for example, the outputs from extension actions, the inputs for map actions, or the inputs and outputs for subprocesses. See also *business object*, *business object type*, *extension action*, *variant variable.*

**CA**—see *certificate authority*.

**certificate**—a security document that binds a public encryption key to an entity (an individual or organization) known as the principal. The security document (a digital certificate) is signed by another entity known as the issuer. A digital certificate for which both the principal and issuer are the same entity is known as a self-signed certificate. A certificate for which the principal and issuer are different entities is issued by a certificate authority (CA) like VeriSign and is known as a CA-issued (or third-party-signed) certificate. Partner Agreement Manager supports both self-signed and CA-issued certificates. PAM also supports the binding of certificates to be used for signature authentication, message encryption, and SSL authentication for channels other than Partner Agreement Manager. See also *certificate authority*, *SSL*.

**certificate authority**—a trusted third-party organization or company that issues digital certificates used to create digital signatures and public-private key pairs. The role of the certificate authority, or CA, is to authenticate the entities (individuals or organizations) involved in electronic transactions. CAs are a critical component in data security and electronic commerce because they guarantee that the two parties exchanging information are really who they claim to be. See also *certificate*.

**channel**—a communications mechanism that encapsulates all the processing information needed to send messages to a partner's system, as well as to translate data received from a partner into Partner Agreement Manager messages. PAM provides channels for RosettaNet, EDI, cXML, and other systems and protocols. See also *message*.

**digital certificate**—see *certificate.*

**DTD**—Document Type Definition. A type of file associated with SGML and XML documents that defines how the formatting tags should be interpreted by the application presenting the document. In Partner Agreement Manager, a DTD file contains the complete description of a business object type's element definition set. See also *business object*, *business object type*, *element definition set*.

**element definition set**—a collection of data fields (or elements) or groups of data fields that defines the structure and meaning of a business object type. See also *business object*, *business object type*.

**encryption certificate**—see *certificate.*

**event**—a piece of information that comes into Partner Agreement Manager as a message from another source (an enterprise system or business application, for example) and triggers a public process. See also *message*.

**event push**—a method that uses the HTTP POST mechanism to push events into Partner Agreement Manager as a way to trigger processes. A port on the Process Server is set to listen for events in the form of HTTP POST messages. When a message is detected, PAM uses the information in the message to generate an event. See also *event*.

**extended enterprise**—a business model under which companies that work together as partners function as efficiently as a single organization through the implementation of automated communication technologies.

**extension action**—a private process action that communicates via an adapter with an external application that is registered with Partner Agreement Manager. You can use an extension action, for example, to launch a spreadsheet application, perform calculations, and update the enterprise system, or to get information from an enterprise system or listen for an event in the enterprise system. See also *adapter*, *private process*.

**LDAP**—Lightweight Directory Access Protocol. LDAP provides a standard method for accessing information from a central directory. After user authentication is set up in the LDAP directory, applications that use the LDAP protocol can retrieve the information from that directory. An authenticated user can log in to any application that supports the LDAP protocol with the same user name and password.

**linked certificate**—see *certificate*.

**map**—a Java Script or VBScript that inserts data into fields in an output business object type generated by a private process. The map specifies which fields in the output business object type receive data, and it identifies the information source.

**map method**—a reusable logical block of code that inserts data into a particular type of element or element sequence in a business object type. Within a map method, you can write the expressions that map individual input and output fields in the sequence. Or you can create a submap and drag input fields to output fields and have Partner Agreement Manager create the appropriate mapping expressions. See also *map*, *submap*.

**mapping action**—a private process action that you use to call a map. The map specifies the fields in a business object type that will receive data extracted from another source. You use a mapping action when you want to extract data from one business object type and insert it in a different business object type. For example, you use a mapping action to transform a purchase order generated by your inventory system into a sales order in a format that your partner expects. See also *map*, *private process*.

**message**—a structured communication used to pass information and control to another partner in a public process. The action in the process passes to the partner who receives the message. The content of a message is determined by its business object type. A message can be transmitted via synchronous or asynchronous methods, as determined by its communication service type. See *business object type*.

**non-repudiation**—a business object security feature that authenticates instances of a business object type and maintains an audit record to verify that they were received by the intended recipient. For business object instances that you receive, Partner Agreement Manager authenticates each instance and maintains an audit record to verify that the instance actually originated with the stated originator. If you disable auditing for a business object type, non-repudiation support is disabled for all messages that contain instances of that business object type.

**notification action**—a private process action that you use to send an e-mail, fax, or pager message to addressees that you specify. You use a notification action to inform someone inside or outside your organization that an event has occurred. For example, you can use a notification action to alert the order entry department when a purchase order arrives from a customer. See also *private process*.

**output object action**—a private process action that you use to bind a business object to the expected output object and path in a public process. You use an output object action at the point in a private process when you are ready to send a business object to the associated public process. This is typically the last action in the private process. See also *private process*.

**partner group**—a group of partners that perform the same role in a process at different times. Instead of duplicating a public process and substituting a different partner name, you can set up a partner group for the public process and then designate a specific partner as the participant when you start an instance of the process. For example, you might design a generic purchasing process that works equally well with any of your suppliers and then designate the appropriate partner when you start the process.

**partner profile**—information that identifies an organization, specifies a contact person in that organization, lists the communication services the organization supports, and defines the organization's security profile. When partners agree to participate in a public process, they must exchange profile information as a way to ensure authenticity before they can proceed.

**PIP**—Partner Interface Process. RosettaNet PIPs are specialized system-to-system XML-based dialogs that define business processes between supply-chain partners and provide models and documents for the implementation of e-commerce standards. Each PIP includes a technical specification based on the RosettaNet Implementation Framework (RNIF), a message guideline document with a PIP-specific version of the business dictionary, and an XML message guideline document. See also *RosettaNet*.

**post method**—the last block of code that is executed when a mapping action runs. Its only parameter is the output business object. You use the post method when you need to perform post-processing on the output business object. For example, you might use the post method to set the value of a summary field based on the number of line items in the output business object, or to examine a range of dates in a repeated group, extract the most recent date, and post that date in a header field. See also *mapping action*, *pre method*.

**pre method**—the first block of code that is executed when a mapping action runs. The pre method's parameters are the map inputs. You use the pre method to access a map's inputs and set global variables based on their content. See also *mapping action*, *post method*.

**private process**—a task or set of tasks that business partners participating in a public process perform at points where they need to take action internally. Partners participating in a public process must implement a private process for each public process step that they own. A private process begins with input from the public process and ends with output that feeds back into the public process. The input can be the receipt of a business object from a partner, or it can be a triggering event from an internal system. The output is the business object that transfers control back to the public process. See also *action*, *process*, *public process*.

**private process action**—see *action*.

**process**—the flow of actions and the exchange of business information between partners in an extended enterprise. A process operates on two levels, public and private. See *extended enterprise*, *private process*, *public process*.

**public process**—the step-by-step flow of messages, events, and actions between two or more business partners. Public processes are set up by agreement between partners, and each step in a public process has a private process associated with it. A public process is developed by one partner, and all the partners who participate in it must review and approve it before it can be implemented. The partner who designs a public process is its owner. See also *private process*, *process*.

**RosettaNet**—a consortium of major information technology, electronic components, and semiconductor manufacturing companies that is working to create and implement industry-wide, open e-business process standards. See also *PIP*.

**script action**—a private process action that consists of a script written in VBScript or JavaScript and is designed to manipulate information or set up conditional actions based on input. You use a script to establish decision-making criteria for branches or loops, to set variables, or to calculate values that are used elsewhere in the private process. See also *private process.*

**security certificate**—see *certificate.*

**self-signed certificate**—see *certificate.*

**signature certificate**—see *certificate.*

**SSL**—Secure Sockets Layer. The SSL protocol is a security protocol that provides for communications privacy and reliability over the Internet. The protocol allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery.

**submap**—a secondary level map that is called by a map method to insert data into an output element other than the top-level element. See *map*, *map method.*

**subprocess action**—a private process action you use to call an existing public process. You can call any public process in which your organization owns the first partner action. For example, you can use a subprocess to get a quote approved by a third-party supplier before responding to a customer. See also *private process.*

**termination action**—a private process action that you use to stop a process at a predetermined point for a reason that you specify. You can use a termination action to deal with errors in data that might prevent a process from completing successfully. For example, you might want to stop a process in cases where an enterprise system passes incomplete or corrupted information to it. See also *private process.*

**third-party-signed certificate**—another name for a CA-issued certificate. See *certificate.*

**timer action**—a private process action that you use to insert a pause. You can use a timer action to specify the period of time you want to elapse before the next action in the process starts. See also *private process.*

**variant variable**—single field variables. Variant variables store text strings—the type of information contained in a single field element. You can use variant variables to store the input for actions, to set flags (such as the time-out flag for an approval action), to move information within scripts, or to store the results of an approval action. See also *business object variable.*

subprocesses
    exporting  133
    importing  134
synchronous communication services for messages
  112
Syntax Checker, in Map Script Editor  84
System folder, described  31

## T
termination actions
    adding  189
    described  148, 189
    in Auditor folder  190
    in Process Audit window  191
    specifying reasons  193
Test mode, for public processes  219
thumbnails, viewing in Public Process window  97
time-outs, in approval actions  171
timer action
    adding  174
    described  147
Timer Properties dialog box  175
toolbars
    hiding and showing  95
    in Element Definition Set Editor, described
        36
    in Map Editor, described  71
    in Private Process window, described  143
    in Public Process window, described  95
top-level map, viewing  81

## U
undefined business objects
    assigning in actions  114
    replacing  130
undefined partners
    assigning in actions  111
    replacing  130

## V
variables
    business object variables, described  134
    creating  136
    deleting  137
    for registered processes  227
    in private processes  200
    in public processes  134
    variants, described  134
variables panel, in Public Process window  135
variant variables
    creating  136
    described  134
VBScript, selecting  157
Verify dialog box
    Private Process  200
    Public Process  129
version control
    for element definition sets  48
    for maps  90
version numbers
    in private processes  145
    in public processes  96

## X
XOR logic in branches  117

## Z
zoom controls, described  97, 146