

PROCESSANALYST

# Getting Started

POWERDESIGNER 6.1

December 1997

Copyright © 1991-1997 Sybase, Inc. and its subsidiaries.  
All rights reserved.  
Printed in the United States of America.

Information in this manual may change without notice and does not represent a commitment on the part of Sybase, Inc. and its subsidiaries.

The software described in this manual is provided by Powersoft Corporation under a Powersoft License agreement. The software may be used only in accordance with the terms of the agreement.

No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc. and its subsidiaries.

Sybase, Inc. or its subsidiaries claim copyright in this program and documentation as an unpublished work, revisions of which were first licensed on the date indicated in the foregoing notice. Claim of copyright does not imply waiver of other rights of Sybase, Inc. and its subsidiaries.

ClearConnect, Column Design, ComponentPack, InfoMaker, ObjectCycle, PowerBuilder, PowerDesigner, Powersoft, S-Designor, SQL SMART, and Sybase are registered trademarks of Sybase, Inc. and its subsidiaries. Adaptive Component Architecture, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Warehouse, AppModeler, DataArchitect, DataExpress, Data Pipeline, DataWindow, dbQueue, ImpactNow, InstaHelp, Jaguar CTS, jConnect for JDBC, MetaWorks, NetImpact, PowerDynamo, Optima++, Power++, PowerAMC, PowerBuilder Foundation Class Library, Power J, PowerScript, PowerSite, Powersoft Portfolio, Powersoft Professional, PowerTips, ProcessAnalyst, Runtime Kit for Unicode, SQL Anywhere, The Model For Client/Server Solutions, The Future Is Wide Open, Translation Toolkit, UNIBOM, Unilib, Uninull, Unisep, Unistring, Viewer, WarehouseArchitect, Watcom, Watcom SQL Server, Web.PB, and Web.SQL are trademarks of Sybase, Inc. or its subsidiaries. Certified PowerBuilder Developer and CPD are service marks of Sybase, Inc. or its subsidiaries. DataWindow is a patented proprietary technology of Sybase, Inc. or its subsidiaries.

AccuFonts is a trademark of AccuWare Business Solutions Ltd.

All other trademarks are property of their respective owners.

# Contents

<b>About This Book</b> .....	<b>v</b>
<b>About the PAM Tutorial</b> .....	<b>1</b>
About the tutorial .....	2
Domain of analysis .....	2
What you will do? .....	3
How long it will take? .....	3
What you will learn? .....	3
<b>Creating the Root Process</b> .....	<b>5</b>
Open a new model.....	6
Create and define a process.....	8
Create and define external entities .....	10
Create data flows between objects.....	12
<b>Decomposing the Root Process</b> .....	<b>15</b>
Decompose a process .....	16
Create and connect processes .....	17
Display the process tree .....	22
Create a control flow .....	23
<b>Creating Data Items and Domains</b> .....	<b>25</b>
Create a data item .....	26
Create a domain .....	29
Attach check parameters to a data item .....	32
Apply a domain to a data item .....	34
Attach data items to data flows.....	36
<b>Storing Data Items</b> .....	<b>43</b>
Create a data store .....	44
Create an object creation flow .....	46
Store data items in the data store .....	48
Connect the data store to the rest of the model.....	52
Check the model.....	55

<b>Managing the Display</b> .....	<b>57</b>
Add a title box .....	58
Change the color of the window.....	61
Align symbols.....	62
Arrange flows.....	64
Display or hide flow text.....	66
Center the model on the page .....	68
Print the model.....	70
Exit ProcessAnalyst .....	71
<b>Glossary</b> .....	<b>73</b>
<b>Index</b> .....	<b>75</b>

# About This Book

**Subject**

This book contains a tutorial designed to introduce you to ProcessAnalyst, PowerDesigner's functional model design module. It does not cover all ProcessAnalyst functions in detail, but shows you how to use the basic functions you need to start.

This tutorial will not make you an expert in ProcessAnalyst; only experience creating real-world models can do that. It will, however, give you hands-on experience and provide a foundation for continued growth.

**Audience**

This book is intended for users who are new to ProcessAnalyst, but who are familiar with the Windows environment.

## CHAPTER 1

# About the PAM Tutorial

This tutorial is a series of five lessons in which you learn how to use ProcessAnalyst to create a ProcessAnalyst Model (PAM).

You learn by building a PAM based on a real-life example of an information system. As you develop this model, you will create all the basic elements of a PAM. You can then apply your knowledge to creating a PAM to meet your needs.

### What is a PAM?

A PAM offers a functional analysis of an information system. It analyzes the functions performed by the various processes that make up the system.

The PAM is complementary to the Conceptual Data Model (CDM). The PAM shows the dynamics that exist between the elements of the system. The CDM describes the static facts at the heart of these elements.

ProcessAnalyst shows the flow of data from one element to another and the transformations the data undergoes.

## About the tutorial

This tutorial will teach you how to create a new PAM.

ProcessAnalyst supports four methodologies:

- ◆ The OMT functional model
- ◆ Yourdon/DeMarco (data flow diagram)
- ◆ Gane & Sarson (data flow diagram)
- ◆ SSADM (data flow diagram)

The OMT functional model is considered the most recent and, consequently, the most powerful version of data flow diagram (DFD) structural data analysis tools. The OMT functional model also offers you more ways of representing data flows than the other methods.

The model you will build provides you with a concrete example of a functional model, that you can then relate to the real world to create models based on your needs. Before starting to create the model graphics, you must define the domain of analysis.

### Domain of analysis

The tutorial model concerns the publishing industry. Between the writer's manuscript and the book on the shelf in a bookstore, there is a transformation process called publication.

For example, a publishing company may want to document its activity by identifying precisely the various stages of publication. Analyzing its activity in this way makes it possible to identify and better define all the functions of the publication process, as well as all the flows of data which circulate between these functions.

The functions and data can then be represented graphically in a PAM.

## What you will do

Chapter 2	First you will create a new PAM. You will learn how to create the root process: the highest level process in the PAM.
Chapter 3	You will learn how to decompose the process to show the functions involved in the root process. You will also learn how to create a control flow between two processes.
Chapter 4	You will create data items and domains with check parameters, and attach data items to the various flows in your model.
Chapter 5	You will store the information described in the model by creating a data store, and an object creation flow going towards it.
Chapter 6	You will use the graphic management functions to improve the presentation of your model. You will check the model to ensure that the PAM you have built is methodologically correct, then save and close the PAM.

## How long it will take

You can do the tutorial in one sitting in about 1½ hours. Or you can stop after any chapter and continue at another time.

## What you will learn

You will learn the basic techniques for creating a PAM:

- ◆ How to create processes, external entities, and data stores
- ◆ How to create different types of links between objects
- ◆ How to decompose a process
- ◆ How to create data items and domains

When you finish this tutorial, you can compare your PAM with the final version (PAMAFTER.PAM), delivered with the ProcessAnalyst module.



## Creating the Root Process

The first step in creating your model is to create the **root process**. This is the function that groups together all the other functions within the domain of analysis. In this example, the root process is the Publish function. This function groups together other functions that will be broken down at another level of analysis of the domain.

The Publish function represents the transformation of data that circulates between an author and a bookstore. In this model, Author and Bookstore are designated as external entities. An **external entity** is an actor that relates to a process either as a source of data to be processed, or as a receiver of data that has already been processed.

An external entity has a more passive role than a process. Whereas the process affects the data by transforming it, and external entity only transmits or receives this data. In this example, the Author external entity is a data transmitter and the Bookstore external entity is a data receiver.

To transmit data between external entities and processes, you need to create data flows. A **data flow** is the transmission vector between the various objects in the model.

In this exercise you will:

- ◆ Open a new model
- ◆ Create and define a process
- ◆ Create and define external entities
- ◆ Create data flows between objects

**How long will it take?**

About 15 minutes.

## Open a new model

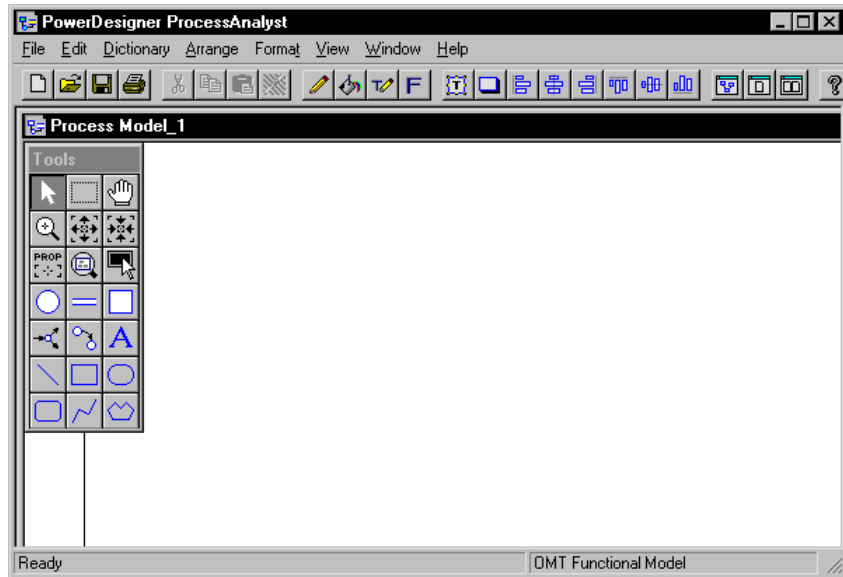
**Where you are**

Chapter 2 Creating the Root Process

- ➔ Open a new model
- Create and define a process
- Create and define external entities
- Create data flows between objects

- 1 Double-click the *ProcessAnalyst* program icon.

The ProcessAnalyst window appears.



Before you begin the exercises, you will save the new model.

**Your screen looks different**

All the screen captures in this book were taken in a Windows 95 environment, with an 800x600 screen resolution. The appearance and proportions of the images which appear on your screen may be slightly different.

- 2 **Select *File*►*Save As*.**

The File Save As dialog box appears.

- 3 **Type *TUTORIAL.PAM* in the File name box.**

- 4 **Click *OK*.**

You can now begin the exercises.

## Create and define a process

**Where you are**  
Chapter 2 Creating the Root Process  
Open a new model  
➔ Create and define a process  
Create and define external entities  
Create data flows between objects

A **process** is a function which has been defined in the domain of analysis. A process transforms the data sent to it.

You will create a process called Publish.

 1 Click the *Process* tool.

2 Click the model workspace.

A process symbol appears at the click position. The process has the name *Prce\_n*, where *n* is the number of the object in the order of creation of objects.

The number that appears above the process name is the process identification number. It indicates the creation order of the process.



**Process number**

You can modify this number from the process property sheet. You can also renumber objects by selecting Dictionary➤Renumber Objects.


3 Click the *right mouse button* to release the Process tool.

4 Double-click the process symbol you have just created.

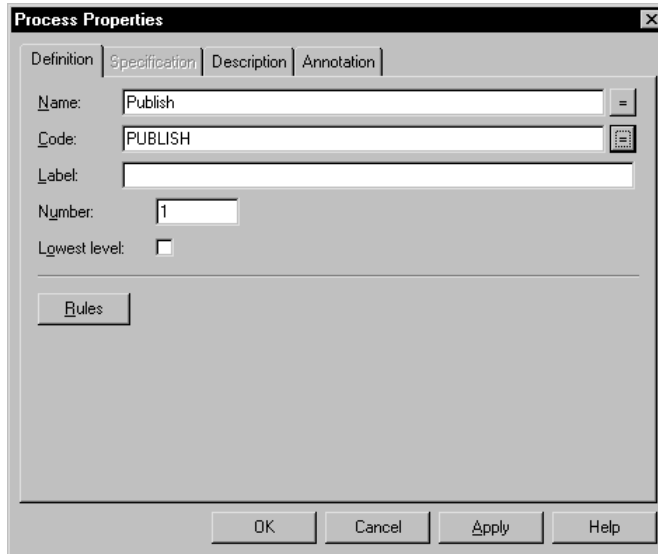
The process property sheet appears.

5 Type *Publish* in the Name box.

This is the name of the process.

- 6 Click the  button at the end of the Code box.

This sets the code equal to the name.



- 7 Click *OK*.

Publish appears in the process symbol.



## Create and define external entities

**Where you are**  
Chapter 2 Creating the Root Process  
Open a new model  
Create and define a process  
➔ Create and define external entities  
Create data flows between objects

The Publish process transforms data it receives from the Author external entity and directs the data to the Bookstore external entity.

You will create the Author external entity.



**1 Click the *External entity tool*.**

**2 Click the model workspace.**

An entity symbol appears at the click position.




**3 Click the *right mouse button* to release the External entity tool.**

**4 Double-click the *external entity* symbol you have just created.**

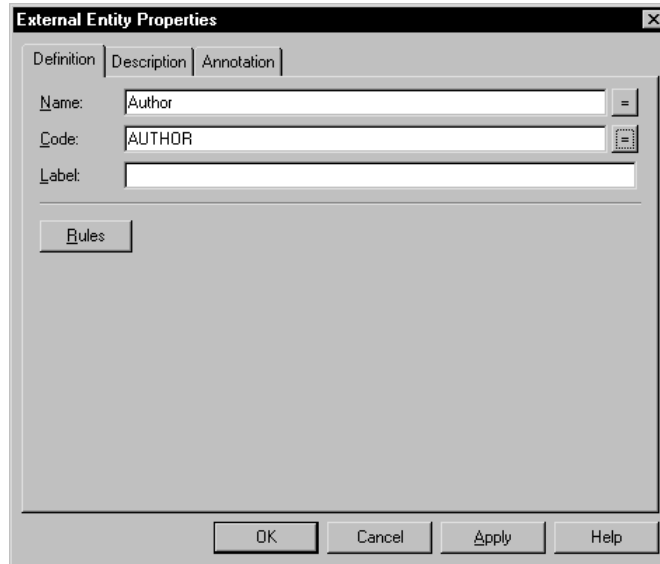
The external entity property sheet appears.

**5 Type *Author* in the Name box.**

This is the name of the external entity.

- 6 Click the  button at the end of the Code box.

This sets the code equal to the name.



- 7 Click *OK*.

Author appears in the external entity symbol.



- 8 Repeat steps 1 to 7 to create another external entity:

Name	Code
Bookstore	BOOKSTORE

## Create data flows between objects

**Where you are**  
Chapter 2 Creating the Root Process  
Open a new model  
Create and define a process  
Create and define external entities  
➔ Create data flows between objects

You will link objects in the root process. In this way you indicate the route taken by the data in the model.

You will create a data flow between the Author external entity and the Publish process.

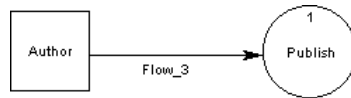
1 Click the *Flow* tool.



2 Click the *Author* external entity and hold the mouse button.

3 Drag the flow to the *Publish* process and release the mouse button.

A flow symbol appears between the two objects.



4 Click the *right mouse button* to release the *Flow* tool.


5 Double-click the flow symbol you have just created.

The data flow property sheet appears.

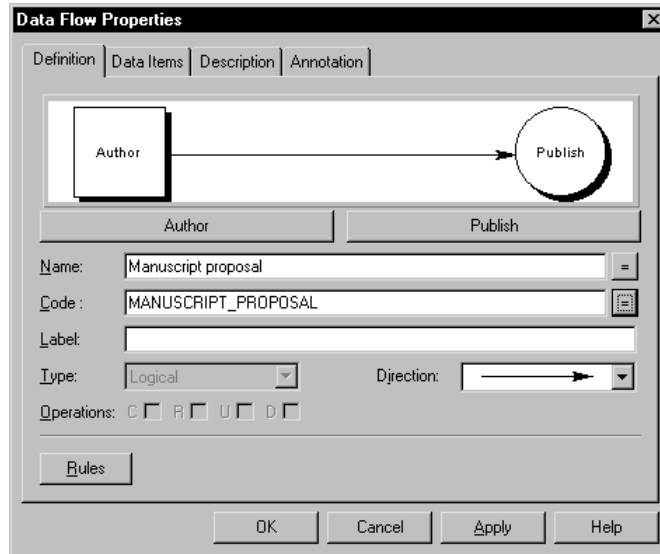
6 Type *Manuscript proposal* in the **Name** box.

This is the name of the flow. The name explains the nature of the link between Author and Publish: an author proposes a manuscript for publication.



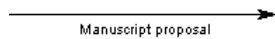
- Click the  button at the end of the Code box.

This sets the code equal to the name.



- Click *OK*.

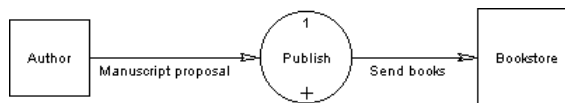
The name *Manuscript proposal* appears next to the flow.



- Repeat these steps to create another flow from the *Publish* process to the *Bookstore* external entity.

Name	Code
Send books	SEND BOOKS

The two flows appear in the PAM.



## Decomposing the Root Process

The **root process** is the process at the top of the hierarchy of processes. It is a process that encompasses other processes. To identify these processes, you must make a more detailed analysis of the root process.

In the example model, the Publish process encompasses two processes:

- ◆ Select only the manuscripts that will be published
- ◆ Print the selected manuscripts

These two processes are at a lower graphical and analytical level than the Publish process. The root process is decomposed. The act of decomposition creates a subprocess. At the subprocess level, you must create two processes: Select and Print. There are two types of flows between these processes:

- ◆ Logical data flow transfers data
- ◆ Control flow communicates an order

In this exercise you will:

- ◆ Decompose a process
- ◆ Create and connect processes
- ◆ Display the process tree
- ◆ Create a control flow

**How long will it take?**

About 15 minutes.

## Decompose a process

### Where you are

#### Chapter 3 Decomposing the Root Process

- ➔ Decompose a process
- Create and connect processes
- Display the process tree
- Create a control flow

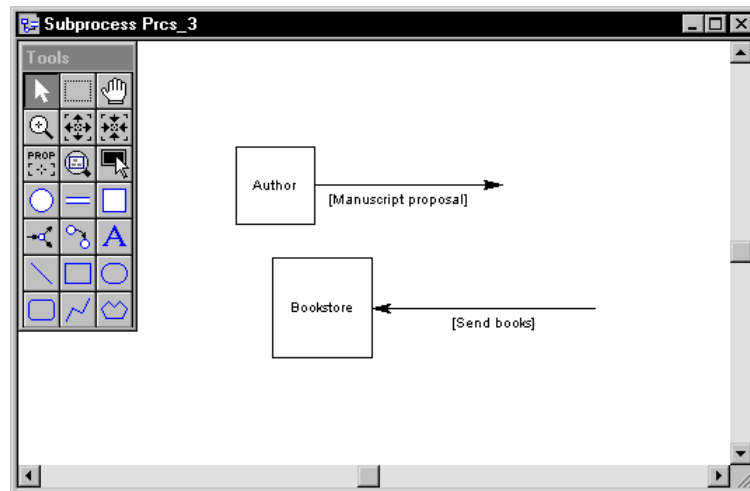
You will decompose a process and create a subprocess.



1 Click the *Decomposition* tool.

2 Click the *Publish* process in the model.

The Publish subprocess window appears.



You have decomposed the Publish process.

In the subprocess window, Send books and Manuscript proposal are represented in a particular way: they are migrated flows.

You will learn how to manage migrated flows in the next section.

## Create and connect processes

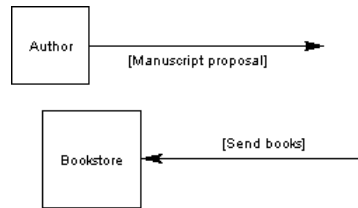
### Where you are

#### Chapter 3 Decomposing the Root Process

#### Decompose a process

- ➔ Create and connect processes
- Display the process tree
- Create a control flow

When you decompose the root process, the links are transferred to the subprocess in the form of migrated flows. **Migrated flows** are links to the upper level process.



A migrated flow is made up of:

- ◆ A symbol at one end, that represents the connected object in the upper level process.
- ◆ An arrow which represents a flow of information

Square brackets around the flow name indicates that it is a migrated flow.

You will now create two processes: Select and Print. You will then connect them to the root process using the migrated flows.



1 Click the *Process* tool.

2 Click the model workspace.


The process symbols appears at the click position.

3 Click the *right mouse button* to cancel the *Process* tool.

4 Double-click the process symbol you have just created.

The process property sheet appears.

5 Type *Select* in the Name box.

6 Click the  button at the end of the Code box.

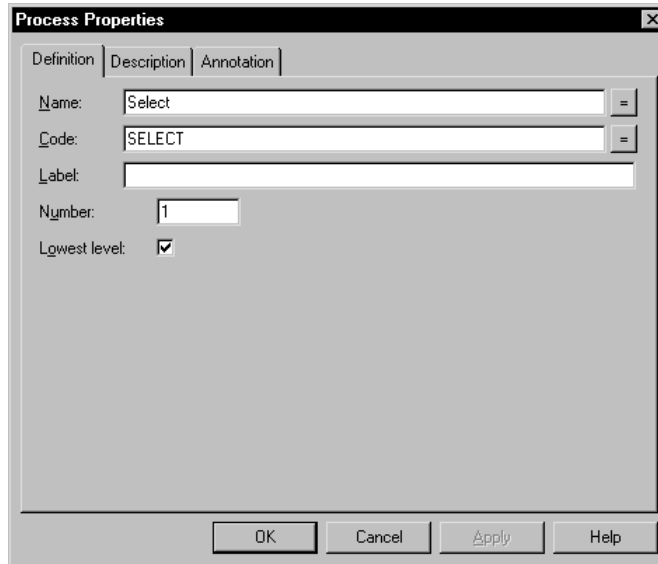
This sets the code equal to the name.

7 Select the *Lowest Level* checkbox.

When you select the Lowest Level checkbox in the process property sheet, you prevent the selected process from being decomposed into subprocesses.

**Check process decomposition**

When a process is not at the lowest level (checkbox not selected) and it is not decomposed, a message appears when you check your model. This reminds you to decompose the process.



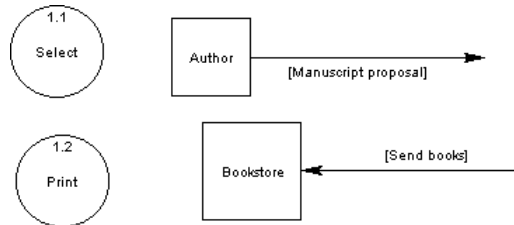
8 Click *OK*.

The Select process appears.

## 9 Repeat steps 1 to 8 to create another process.

Name	Code	Lowest level
Print	PRINT	✓

The subprocess now contains two processes and two connectors. You will link these objects to one another.

10 Click the *Flow* tool.11 Click the *Select* process and hold the mouse button.12 Drag the flow towards the *Print* process and release the mouse button.


The flow symbol appears between the objects.

13 Click the *right mouse button* to release the *Flow* tool.

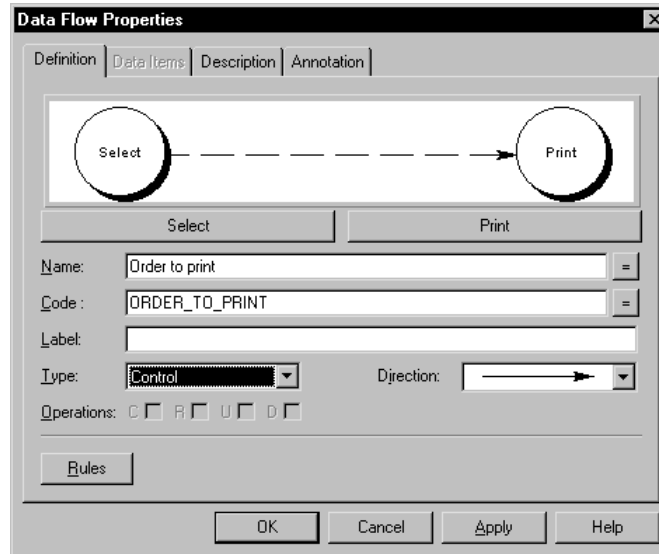
## 14 Double-click the flow you have just created.

The flow property sheet appears.

15 Type *Send direct* in the *Name* box.

- 16 Click the  button at the end of the Code box.

This sets the code equal to the name.

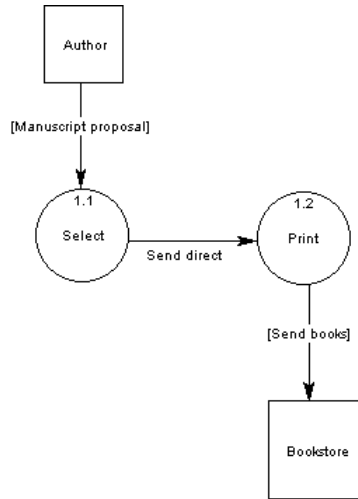


- 17 Click *OK*.

- 18 Move the objects by selecting them and dragging them to the positions shown below.

**Moving migrated flows**

You move an migrated flow by selecting and dragging one end at a time.



You have connected the objects in the subprocess to the root process.



## Display the process tree

**Where you are**

Chapter 3 Decomposing the Root Process

Decompose a process

Create and connect processes

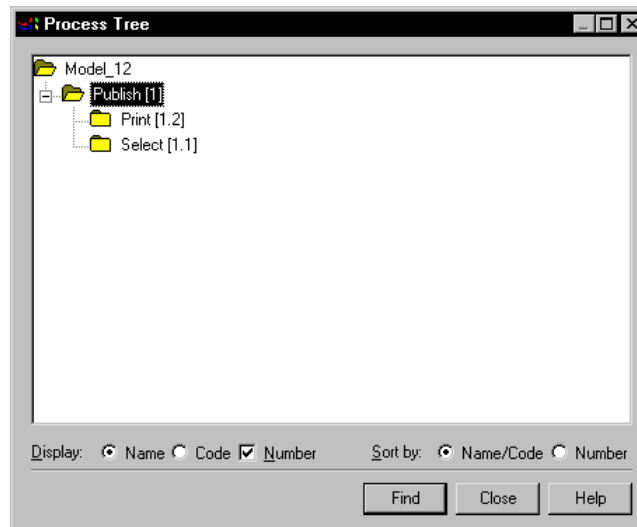
➔ Display the process tree

Create a control flow

When you decompose a process you create subprocesses. To find your way around the process hierarchy and to keep a global view of your model, you can use the Process Tree.

- 1 Select *Dictionary* ➤ *Subprocesses* ➤ *Process Tree*.

The Process Tree window appears.



The process tree gives you a global view of the processes and enables you to find a process in the diagram using the Find button.

- 2 Click *Close*.

The Process Tree window closes.

## Create a control flow

**Where you are**Chapter 3 Decomposing the Root Process

Decompose a process

Create and connect processes

Display the process tree

➔ Create a control flow

In the subprocess, you have created a data flow between the two processes Select and Print. This data flow transmits information about the format, number of copies, and the collection to which the printed book belongs.

There is another flow of information between Select and Print. This flow transmits the following order: "a manuscript which has been selected must be printed." This is not a data flow, but a control flow. A **control flow** transmits an order which starts another process. It can only connect two processes.



1 Click the *Flow* tool.

2 Click the *Select* process and hold the mouse button.

3 Drag the flow towards the *Print* process and release the mouse button.


The flow symbol appears between the objects. By default, the flow is a logical data flow. You will modify the type of this flow.

4 Click the *right mouse button* to release the *Flow* tool.

5 Double-click the flow symbol you just created.

The data flow property sheet appears:

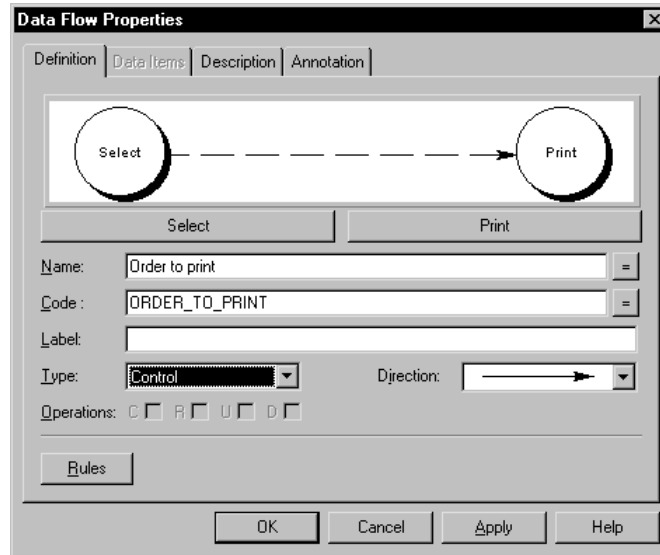
6 Type *Order to print* in the *Name* box.

7 Click the  button at the end of the *Code* box.

This sets the code equal to the name.

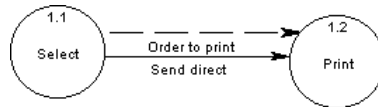
8 Select *Control* from the Type dropdown listbox.

This transforms the data flow into a control flow. You can only do this because the flow does not transmit data, and it links two processes.



9 Click *OK*.

The Order to Print control flow is represented by a dashed line.



## Creating Data Items and Domains

A **data item** is a basic piece of information in the PAM. In this PAM, data items are the format of a book, the date of publication, and the address of a bookstore.

A **domain** identifies the types of data items in your project. It defines the set of values for which a data item is valid. Domains make it possible to standardize the characteristics of data items and make global modifications to the model when necessary.

You can specify more precisely what values are allowed for a data item by attaching check parameters to it. **Check parameters** define minimum, maximum, or accepted values.

In this exercise, you will learn how to:

- ◆ Create a data item
- ◆ Create a domain
- ◆ Attach check parameters to a data item
- ◆ Apply a domain to a data item
- ◆ Attach data items to data flows


**How long will it take?**

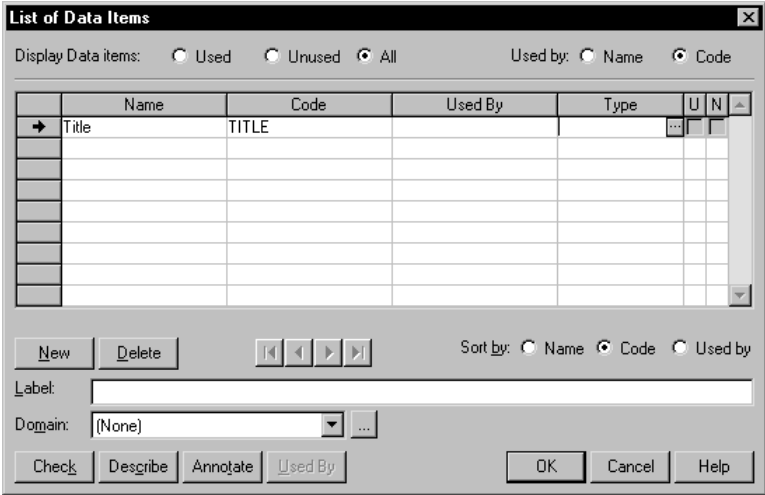
About 20 minutes.

## Create a data item

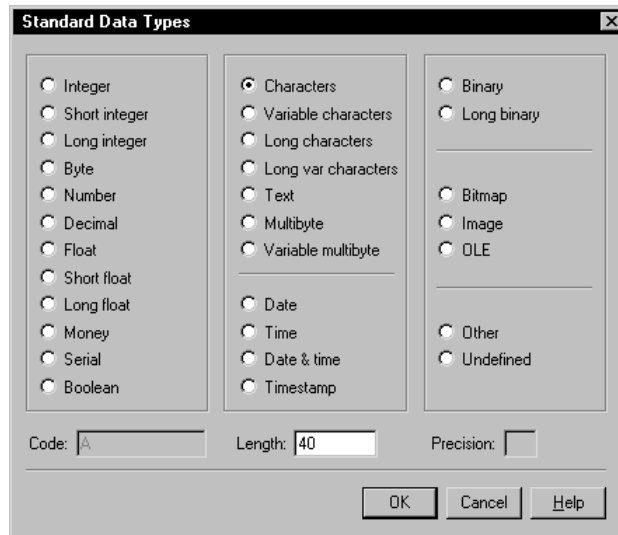
**Where you are**  
Chapter 4 Creating Data Items and Domains  
➔ Create a data item  
Create a domain  
Attach check parameters to a data item  
Apply a domain to a data item  
Attach data items to data flows

You will create the Title data item for the title of the book, then you will create the other data items in the PAM.

- 1 **Select *Dictionary* ➤ *List of Data Items*.**  
The List of Data Items appears.  
There is an arrow at the beginning of the first line.
- 2 **Type *Title* in the Name column.**  
This is the name of the data item.
- 3 **Click the *Code* column and click the  button to set the code equal to the name.**



- 4 Click the *Type* column and click the  button that appears in the column. The List of Standard Data Types appears.
- 5 Click the *Characters* radio button.
- 6 Type *40* in the Length box to limit the length of the title to 40 characters.



- 7 Click *OK*.  
The value *A40* appears in the *Type* column of the *Title* data item.

8 Now create the following data items:

Name	Code	Data type	Type code
Price	PRICE	Money Length 5 Precision 2	MN5,2
Collection	COLLECTION	Variable characters Length 30	VA30
Issue date	ISSUE_DATE	Date	D
Publication deadline	PUBLICATION_DEADLINE	Date	D
Format	FORMAT	Number Length 5 Precision 2	N5,2
ISBN	ISBN	Number Length 20	N20
Bookstore name	BOOKSTORE_NAME	Characters Length 80	A80
Author name	AUTHOR_NAME	Characters Length 80	A80
Special price	SPECIAL_PRICE	Money Length 5 Precision 2	MN5,2
Number of copies	NUMBER_OF_COPIES	Number Length 6	N6

**Creating a new item in a list**  
Click the New button to go to a new line in the list.

9 Click *OK*.

## Create a domain

**Where you are**

Chapter 4 Creating Data Items and Domains

Create a data item

➔ Create a domain

Attach check parameters to a data item

Apply a domain to a data item

Attach data items to data flows

A **domain** identifies the type of data items in your project. You use a domain to define the values for which a data item is valid. When you apply a domain to a data item, the data item inherits the characteristics of the domain.

You will create two domains in your model: Date and Name.

1 Select *Dictionary* ➤ *List of Domains*.

The List of Domains appears.

2 Type *Date* in the Name box.

3 Click the *Code* column and click the  button that appears in the column.

4 Click the *Type* column and click the  button that appears in the column.

The List of Standard Data Types appears.

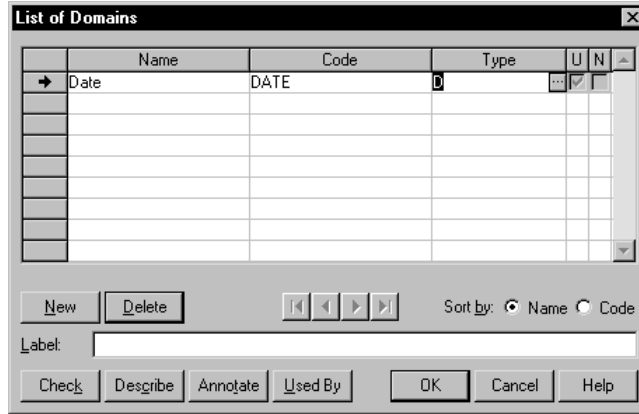
5 Click the *Date* radio button.

This applies the Date data type to the Date domain.



6 Click *OK*.

The value D appears in the Type column. When you apply this domain to any data item used to record a date, the data item will inherit the data type of the domain.

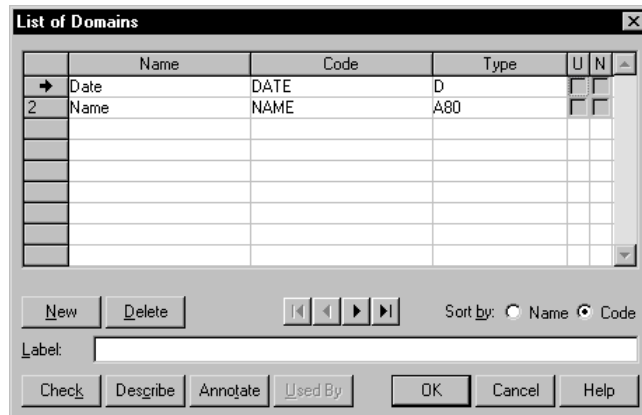


7 Now create the following domain:

Name	Code	Data type	Length
Name	Name	Characters	80

**Creating a new item in a list**  
 Click the New button to go to a new line in the list.

The new domain appears in the list of domains.



8 Click *OK*.

#### Default data type

If you do not apply a data type to a domain, it automatically takes the default data type defined in the Model Options dialog box.

## Attach check parameters to a data item

**Where you are**

Chapter 4 Creating Data Items and Domains

Create a data item

Create a domain

➔ **Attach check parameters to a data item**

Apply a domain to a data item

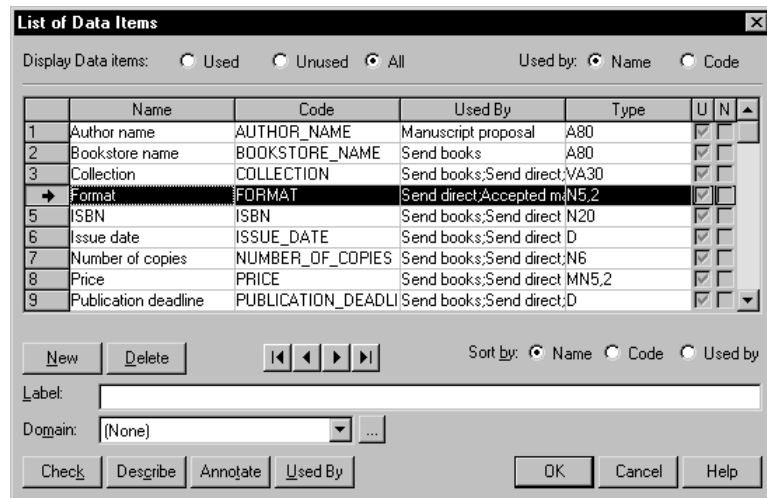
Attach data items to data flows

You will attach check parameters to the Format data item to define the different types of format the book can have. These check parameters ensure that the format does not exceed the minimum and maximum values.

1 **Select Dictionary** ➤ *List of Data Items*.

The List of Data Items appears.

2 **Select Format**.



**3 Click the *Check* button in the List of Data Items.**

The Properties dialog box for the Format data item opens at the Standard Parameters page.

**4 Type *10* in the Minimum box.**

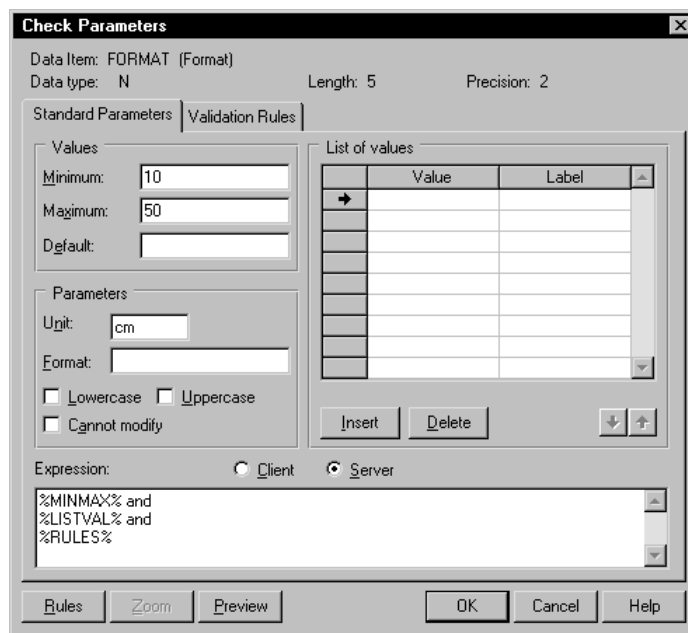
This means that the minimum size is 10 cm.

**5 Type *50* in the Maximum box.**

This means that the maximum size is 50 cm.

**6 Type *cm* in the Unit box.**

The values appear in the dialog box.



**7 Click *OK* in each of the dialog boxes.**

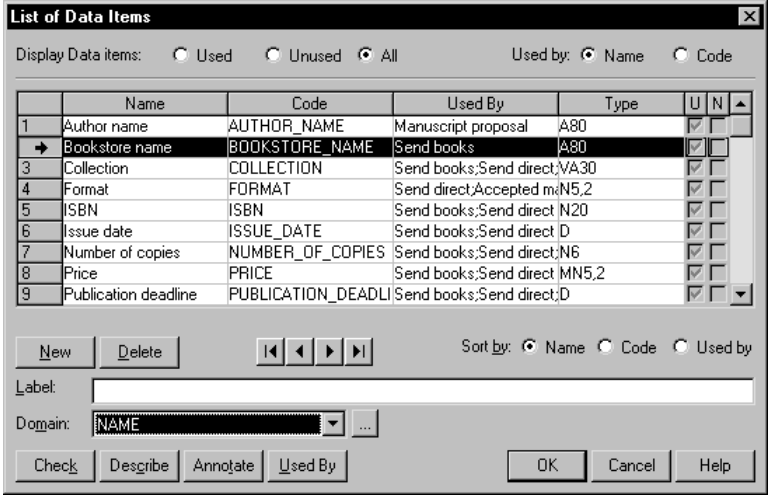
## Apply a domain to a data item

**Where you are**  
Chapter 4 Creating Data Items and Domains  
Create a data item  
Create a domain  
Attach check parameters to a data item  
➔ Apply a domain to a data item  
Attach data items to data flows

You will learn to apply a domain to a data item. In your model, you will apply the Name domain to the Bookstore name data item.

The Name domain specifies the data type A80. All the data items to which you apply this domain will inherit this data type.

- 1 **Select *Dictionary* > *List of Data Items*.**  
The List of Data Items appears.
- 2 **Select *Bookstore name*.**  
An arrow appears at the beginning of the line.
- 3 **Select *Name* from the Domain dropdown listbox at the bottom of the dialog box.**  
Name appears in the domain box.



- 4 Repeat steps 2 to 5 to apply the following domains to the following data items:

<b>Data item</b>	<b>Domain</b>
Author name	Name
Issue date	Date
Publication deadline	Date

- 5 Click *OK*.

## Attach data items to data flows

**Where you are**  
Chapter 4 Creating Data Items and Domains  
Create a data item  
Create a domain  
Attach check parameters to a data item  
Apply a domain to a data item  
➔ Attach data items to data flows

There are different types of flows in ProcessAnalyst. Data flows, which are the most common, are used to transfer data items from one object to another.

You will learn to attach data items to the data flows you created in the root process. Before you do this, you must move up a level to the root process.

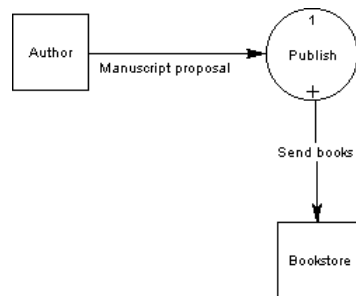
- 1 **Select** *Dictionary* ➤ *Subprocesses* ➤ *Up One Level*.

*or*



- Double-click the** *Decomposition tool*.

You move up to the root process.



- 2 **Select** *Dictionary* ➤ *List of Objects*.

The List of Objects opens to the Processes page.

- 3 **Click the** *Data Flows* **tab**.

The list of flows appears. The first flow in the list is selected by default.

This list only displays flows in the root process. The Send Manuscript and Order to Print data flows do not appear in the list because they belong to a lower level.

**Local and global objects**

External entities and data stores are global objects. They appear in the List of Objects at all process levels. Flows and processes are local objects. They appear in the List of Objects for a particular level.

**4 Select the *Send books* data flow.**

The screenshot shows the 'List of Objects' dialog box with the 'Data Flows' tab selected. The table below shows the data flow 'Send books'.

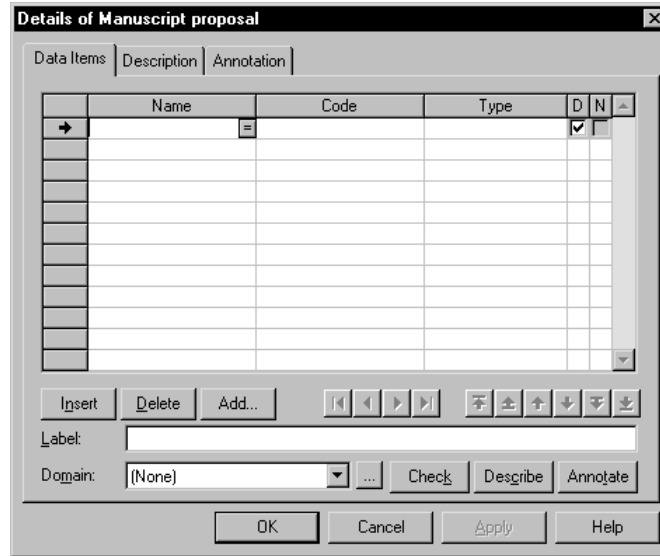
	Name	Code	D	M	N	
1	Manuscript proposal	MANUSCRIPT_PROPOSAL	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
→	Send books	SEND_BOOKS	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Below the table, the 'Object 1' is 'Publish' and 'Object 2' is 'Bookstore'. The 'Type' is 'Logical' and the 'Direction' is '→'. The 'Operations' are 'C', 'R', 'U', and 'D'.



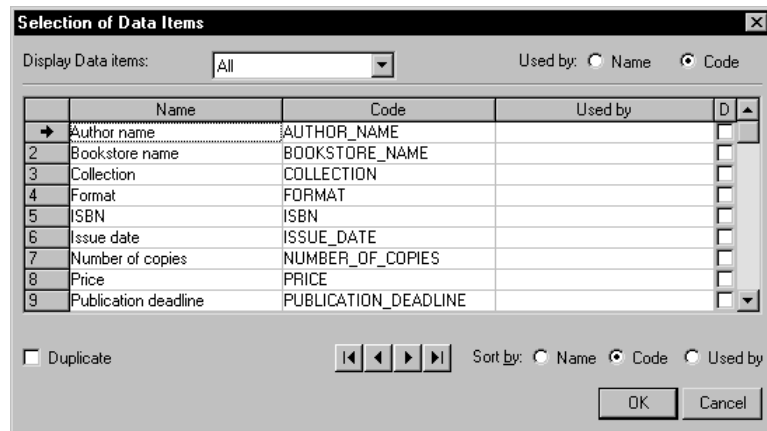
5 Click the *Details* button.

The data flow Details sheet appears.



6 Click the *Add* button.

The Selection of Data Items dialog box appears.



By default, the dialog box lists all the data items.

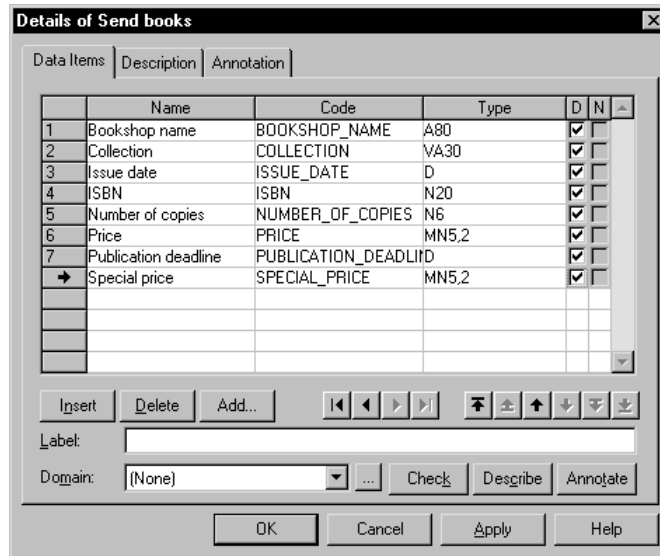
- 7 Select *Bookstore name, Collection, ISBN, Issue date, Number of copies, Price, Publication deadline, and Special price.*

**Multiple selection in a list**

You can select several items in a list by pressing CTRL and clicking in the number column of the data item you want to select.

- 8 Click *OK.*

The data items appear in the data flow Details sheet.



- 9 Click *OK.*

- 10 Repeat steps 4 to 7 to apply the following data items to the Manuscript proposal data flow:

Data flow	Data items
Manuscript proposal	Author name Title

- 11 Click *OK* in each of the dialog boxes.

The list of objects closes.

You will now use another method to attach data items to the data flow in the subprocess.

**12 Click the *Decomposition Tool*.**

**13 Click the *Publish process*.**

The subprocess appears.

**14 Double-click the *Send direct flow*.**

The flow property sheet appears.

**15 Click the *Data Items tab*.**

The list of data items attached to the flow appears. The list is empty.

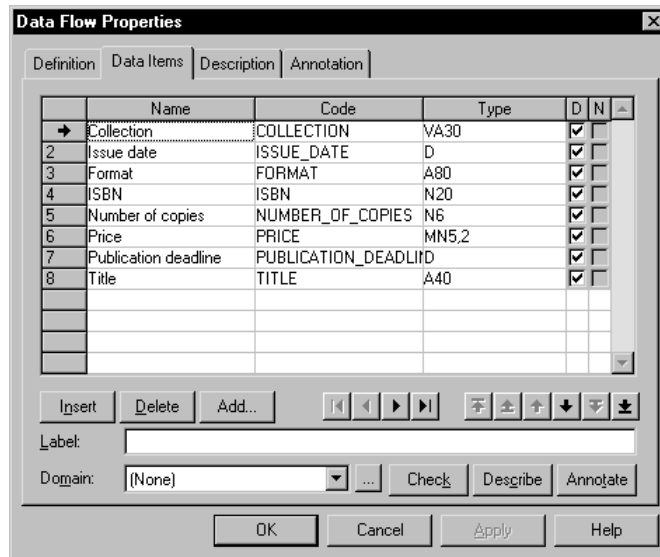
**16 Click the *Add button*.**

The Selection of Data Items dialog box appears.

**17 Select *Collection, Format, ISBN, Issue date, Number of copies, Price, Publication deadline, and Title*.**

**18 Click OK in the selection dialog box.**

The selected data items appear in the list.

**19 Click OK.**

The data items are attached to the Send direct data flow.

## Storing Data Items

Two things can happen to a data item in the model:

- ◆ It can be transformed directly by the processes
- ◆ It can be temporarily stored in a zone where it will be easy to retrieve.

When the data item is not used directly, it passes into a data store. The **data store** is an object that stores the data items in the model.

Type of object	What it does
Data store	Stores and allows access to data items
External entity	Sends or receives data items

You created the Select and Print processes in the Publish subprocess. Between these two processes there is a control flow which indicates that the selected manuscripts are sent directly for printing.

In reality, things are not that simple and it is necessary to refine the analysis at this level. What may happen, is that the manuscripts that go directly to the printer belong to popular authors whose books may become best-sellers. Other manuscripts may be stored somewhere while they are waiting to be printed.

In this exercise, you will learn to:

- ◆ Create a data store
- ◆ Create an object creation flow
- ◆ Store data items in the data store
- ◆ Connect the data store to the rest of the model
- ◆ Check the model

**How long will it take?**

About 20 minutes.

## Create a data store

**Where you are**  
Chapter 5 Storing Data Items  
➔ Create a data store  
Create an object creation flow  
Store data items in the data store  
Connect the data store to the rest of the model  
Check the model

A data store is a passive object that stores data for future access.

You will create the Manuscripts Pending data store in the Publish subprocess.

Before you begin

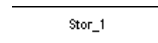
Before you begin make sure that you are still at the subprocess level.



**1 Click the *Data store* tool.**

**2 Click the model workspace.**

A data store symbol appears at the click position.



**3 Click the *right mouse button* to release the **Data store** tool.**

**4 Double-click the data store symbol you just created.**


The data store property sheet appears.

**Data store number**

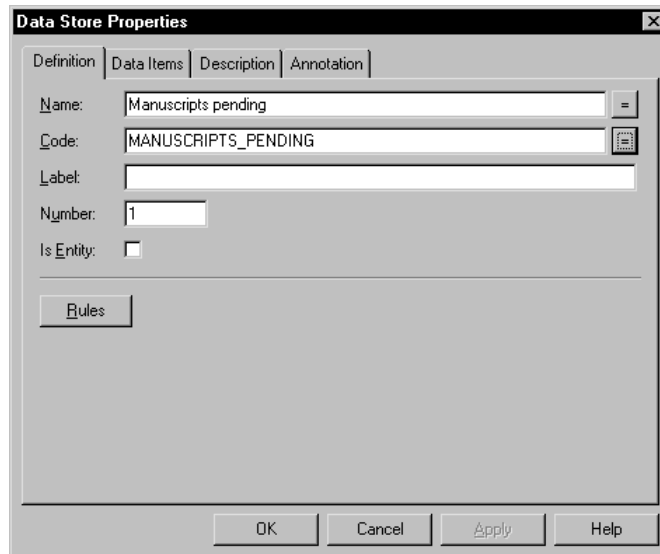
Data stores have an order number which appears on the data store property sheet. You can modify this number directly on the property sheet, or select Dictionary ► Renumber Objects.

**5 Type *Manuscripts pending* in the Name box.**

This is the name of the data store.

- Click the  button at the end of the Code box.

This sets the code equal to the name.



- Click *OK*.

Manuscripts pending appears in the data store symbol:

Manuscripts pending

## Create an object creation flow

**Where you are**

Chapter 5 Storing Data Items

Create a data store

➔ Create an object creation flow

Store data items in the data store

Connect the data store to the rest of the model

Check the model

An object creation flow creates an object intended for future use, in a data store.

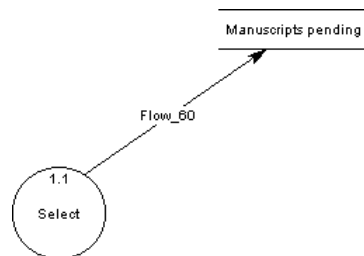
You will create an object creation flow called Accepted Manuscript between the Select Process and the Waiting Manuscripts data store. This creates an Accepted Manuscript object in the data store for each selected manuscript.



1 Click the *Data flow* tool.

2 Click the *Select* process, drag the flow towards the *Manuscripts pending* data store, and release the mouse button.

The flow appears between the two objects.




3 Click the *right mouse button* to release the Flow tool.



- 4 Double-click the flow symbol you have just created.

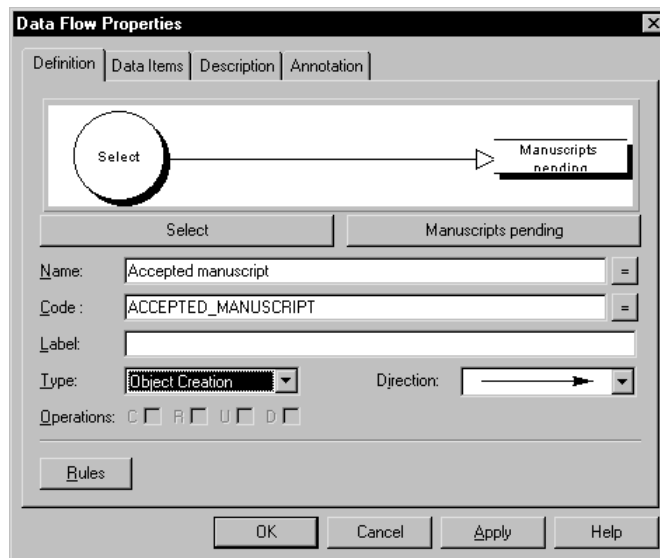
The data flow property sheet appears.

- 5 Type *Accepted Manuscript* in the Name box.

- 6 Click the  button at the end of the Code box.

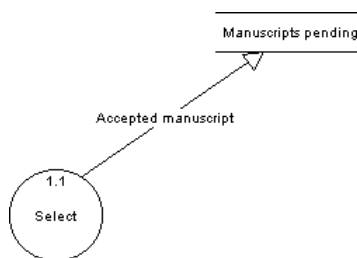
This sets the code equal to the name.

- 7 Select *Object Creation* from the Type dropdown listbox.



- 8 Click *OK*.

Accepted manuscript appears in the diagram.



## Store data items in the data store

Where you are  
Chapter 5 Storing Data Items  
Create a data store  
Create an object creation flow  
➔ Store data items in the data store  
Connect the data store to the rest of the model  
Check the model

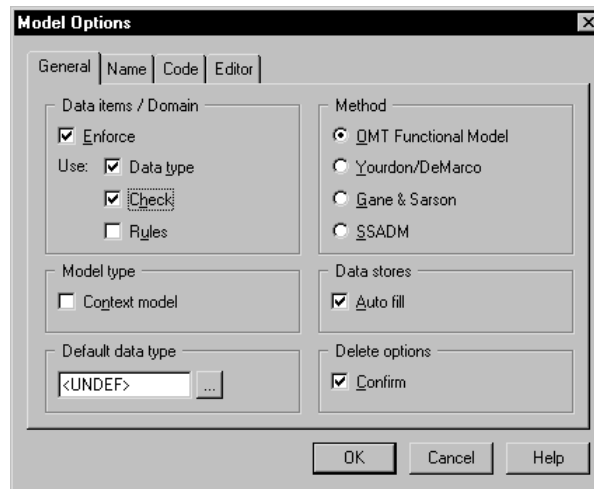
A data store is a reservoir of data items. The data items in this PAM are sent to the Manuscripts Pending data store by the Accepted Manuscript object creation flow.

- 1 Select *File* > *Model Options*.

The Model Options dialog box appears.

- 2 Select the *Auto fill* checkbox in the *Data stores* groupbox.

This ensures that the data transmitted by a flow automatically appears in the target data store.

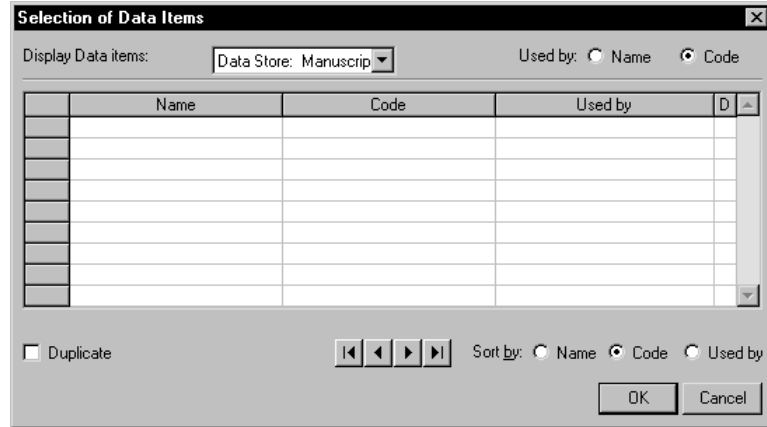


- 3 Click *OK*.



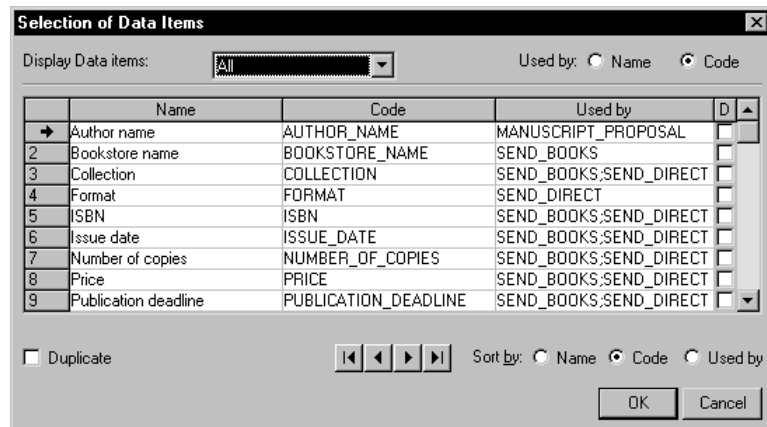
6 Click the *Add* button.

The data item selection window appears. Because the flow is attached to the Manuscripts Pending data store, the window lists the data items contained in the data store. At the moment, this list is empty.



7 Select *All* from the Display Data Items dropdown listbox.

The data item selection window displays all the data items in the model.



8 Select *Collection*, *Format*, *Number of copies*, *Publication deadline*, and *Title*.

9 Click *OK*.

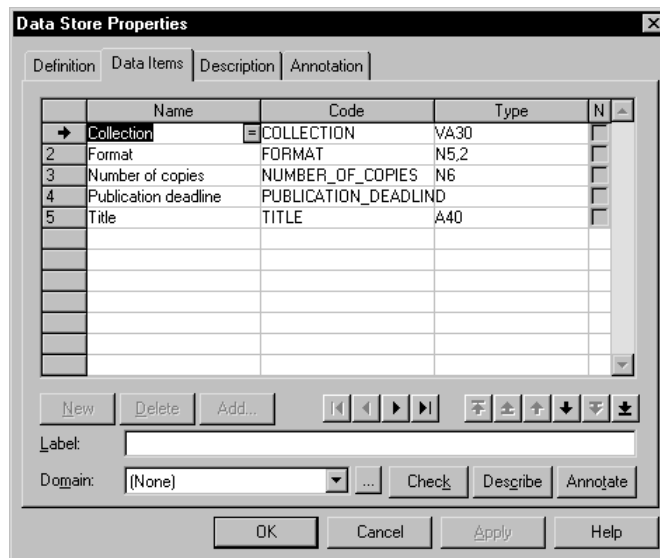
The selected data items appear in the Data Item page of the flow property sheet.

10 Click *OK* to close the flow property sheet.

The data items are directly transmitted to the data store.

11 Double-click the *data store* to display its property sheet.

12 Click the *Data Items* tab.



The data items appear in the list because the Auto fill checkbox in the Model Options dialog box is selected.

#### Filling the data store


When the Auto fill checkbox is not selected, you must use the Add button in the data items page of the data store property sheet to attach the data items to the data store.

13 Click *OK*.

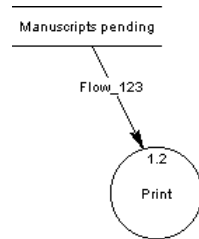
## Connect the data store to the rest of the model

**Where you are**  
Chapter 5 Storing Data Items  
Create a data store  
Create an object creation flow  
Store data items in the data store  
➔ **Connect the data store to the rest of the model**  
Check the model


You can now finish connecting your data store to the rest of the model by creating a flow between the data store and the Print process. This flow represents the transfer of manuscripts from the place they are stored to the printer.

- 1  Click the *Flow* tool.
- 2 Click the *Manuscripts pending* data store, drag the flow to the *Print* process, and release the mouse button.

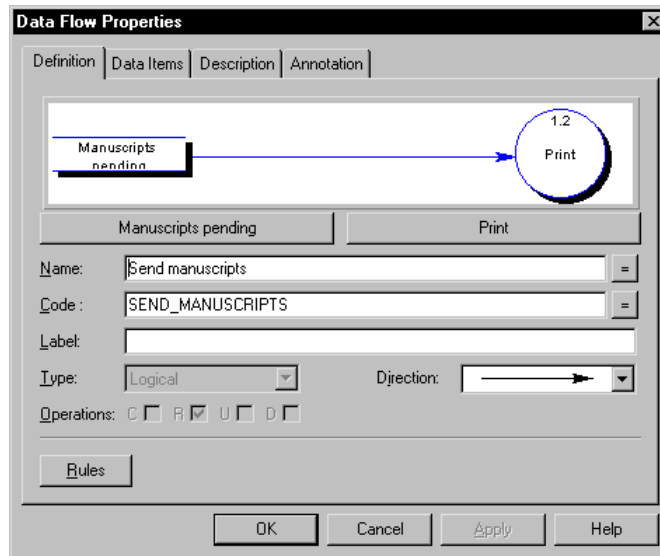
The flow appears between the two objects.



- 3 Click the *right mouse button* to release the Flow tool.
- 4 Double-click the flow symbol to display its property sheet.
- 5 Type *Send manuscripts* in the Name box.

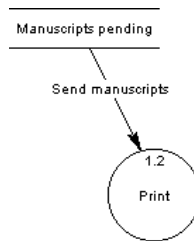
- 6 Click the  button at the end of the Code box.

This sets the code equal to the name. The flow type remains Logical.



- 7 Click *OK*.

Send manuscripts appears in the diagram.



You must now attach data items to this flow so that they can be processed by the Print process.

- 8 Double-click the flow.

The data flow property sheet appears.

**9 Click the *Data Items* tab.**

**10 Click the *Add* button.**

The list of data items appears. It displays the list of all the data items contained in the data store.

**11 Select *Collection, Format, Number of copies, Publication deadline, and Title*.**

**12 Click *OK* in each of the dialog boxes.**

The data items in the data store are transmitted to the Print process.



## Check the model

**Where you are**

Chapter 5 Storing Data Items

Create a data store

Create an object creation flow

Store data items in the data store

Connect the data store to the rest of the model

➔ Check the model

ProcessAnalyst can check if a PAM is methodologically correct. Error and warning messages indicate problems in the PAM.

**1** Select *Dictionary* ➤ *Check Model*.

A dialog box shows the progress of the check operation. It displays error and warning messages. The last message indicates whether or not the model is correct.

**2** Click *OK*.

**Display error messages**

You can display the error messages by selecting *Dictionary* ➤ *Display Messages*.

## CHAPTER 6

# Managing the Display

When you finish creating the PAM, you can organize the model using the display options.

In this exercise, you will learn to:

- ◆ Add a title box
- ◆ Change the color of the window
- ◆ Align symbols
- ◆ Arrange flows
- ◆ Display or hide flow text
- ◆ Center the model on the page
- ◆ Print the model
- ◆ Exit ProcessAnalyst

**How long will it take?**

About 10 minutes.

## Add a title box




<b>Where you are</b>
<u>Chapter 6 Managing the Display</u>
➔ Add a title box
Change the color of the window
Align symbols
Arrange flows
Display or hide flow text
Center the model on the page
Print the model
Exit ProcessAnalyst

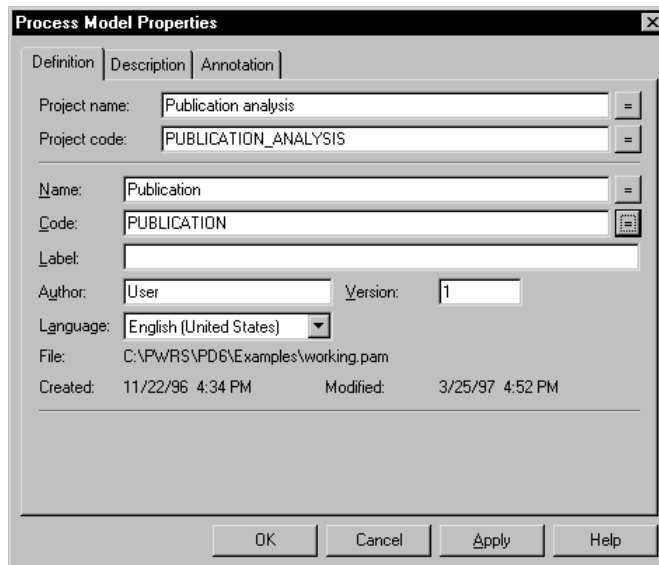
You will add a title box to the model. The title box contains the essential information about the model.

- 1 **Make sure that you are at the root process level by selecting *Window*➤*Process Model\_1*.**
- 2 **Select *Edit*➤*Add Title*.**

A title box appears in the middle of the model.

Process Model		
Project :		
Model : Model 1		
Author :	Version:	09/08/1996

- 3  Select the *Pointer* tool and double-click the title box.  
The model property sheet appears:
- 4 Type *Publication analysis* in the Project name box.
- 5 Click the  button at the end of the Project code box to set the project code equal to the project name.
- 6 Type *Publication* in the Name box.
- 7 Click the  button at the end of the Code box to set the code equal to the name.
- 8 Type your name in the Author box.  
This indicates that you are the owner of the model.
- 9 Type *1* in the Version box.  
This indicates that this is the first version of the model.



**Process Model Properties**

Definition | Description | Annotation

Project name:  =

Project code:  =

Name:  =

Code:  =

Label:

Author:  Version:

Language:

File: C:\P\WRS\PD6\Examples\working.pam

Created: 11/22/96 4:34 PM Modified: 3/25/97 4:52 PM

OK Cancel Apply Help

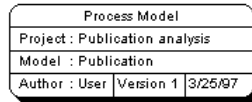
**10 Click *OK*.**

These details appear in the title box and the name of the model appears in the banner of the process window.

**11 Select the title box.**

**12 Select *Format* ► *Shadow***

A shadow appears behind the title box.



**13 Drag and drop the title box at the top of the model page.**

**14 Select *Format* ► *Background color*.**

**15 Click a color in the palette.**

**16 Click *OK*.**

The title box background changes to the selected color.

**Default colors**

You can set the default colors for symbols by selecting *File* ► *Display Preferences*, clicking on the *Symbols* tab, and selecting the colors you want to attribute to each type of symbol.

## Change the color of the window

**Where you are**

Chapter 6 Managing the Display

Add a title box

➤ Change the color of the window

Align symbols

Arrange flows

Display or hide flow text

Center the model on the page

Print the model

Exit ProcessAnalyst

You can change the background of your model by changing the background color of the ProcessAnalyst window.

1 Select *Window* ➤ *Window Color*.

2 Click a color in the palette.

3 Click *OK*.

The model background changes to the selected color.

## Align symbols

**Where you are**  
Chapter 6 Managing the Display  
Add a title box  
Change the color of the window  
➔ Align symbols  
Arrange flows  
Display or hide flow text  
Center the model on the page  
Print the model  
Exit ProcessAnalyst

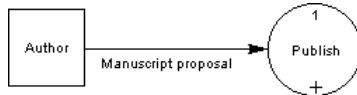
You will learn how to vertically and horizontally align the symbols in the root process.

- 1 Select the *Pointer* tool and draw a selection rectangle encompassing Author, Manuscript Proposal and Publish.

**Selecting several symbols at once**  
You can also select more than one symbol at a time by holding **SHIFT** and clicking each symbol in turn.

- 2 Click the *Align vertical* tool.

This tool enables you to align objects vertically on a horizontal axis. The Manuscript proposal flow becomes horizontal.

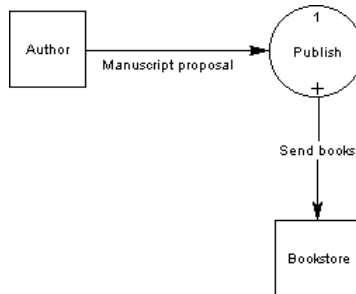


- 3 Draw a selection rectangle encompassing Publish, Send books, and Bookstore.

 4 Click the *Align horizontal* tool.

This tool enables you to align objects horizontally on a vertical axis. The Send books flow becomes fully vertical.

The symbols are now all fully aligned, either vertically or horizontally.





## Arrange flows

**Where you are**  
Chapter 6 Managing the Display  
Add a title box  
Change the color of the window  
Align symbols  
➔ **Arrange flows**  
Display or hide flow text  
Center the model on the page  
Print the model  
Exit ProcessAnalyst

Flows are readily modifiable. You can correct their alignment and modify their attach points.

You will learn how to do this using the Order To Print and Send Direct flows in the subprocess. Before you start, make sure that you are at the level of the subprocess.

Both data flows are parallel. You will learn how to separate them so that you do not confuse them.

- 1 **Select *Window* ➤ *Subprocess Publish*.**
- 2 **Click the *Order to Print control flow*.**
- 3 **Select *Arrange* ➤ *Disposition* ➤ *Arrange Attach Points*.**

The attach points of the flow move to the center of gravity of the two processes.

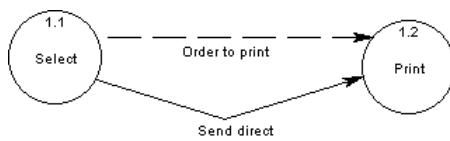
### **Moving a flow towards another object**

You can modify the source or target of a flow by clicking the flow to select it, and holding the CTRL key, while you move one of the attach points to another object in the diagram.

- 4 **Click the *Send direct data flow*.**
- 5 **Press the CTRL key while you click the center of the flow.**

A handle appears on the flow.

**6 Drag the handle to create a bend in the flow.**



This represents the flows more clearly and makes them easier to distinguish.

## Display or hide flow text

**Where you are**  
Chapter 6 Managing the Display  
Add a title box  
Change the color of the window  
Align symbols  
Arrange flows  
➔ Display or hide flow text  
Center the model on the page  
Print the model  
Exit ProcessAnalyst

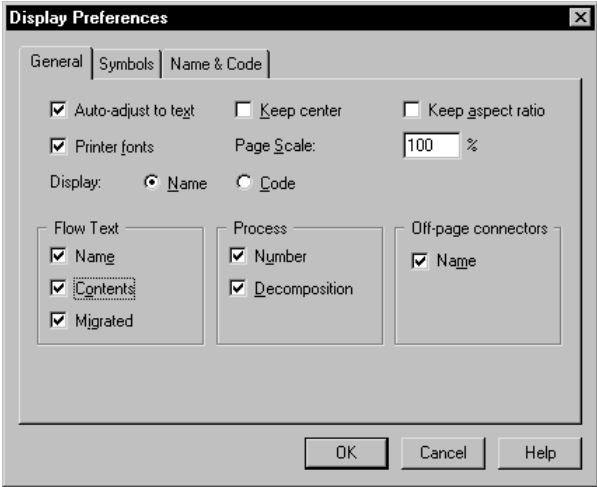
You can modify flow text so that it displays either:

- ◆ Names of the flows
- ◆ Names of the data items transported by the flows.

1 Select *File* ➤ *Display Preferences*.

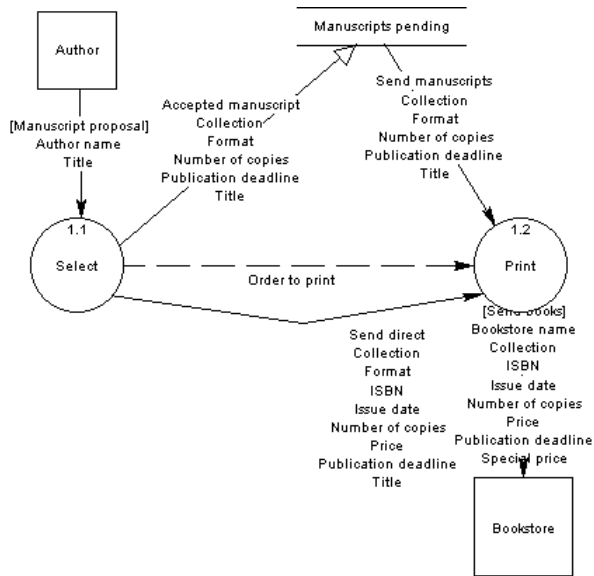
The Display Preferences dialog box appears.

2 Select the *Contents* checkbox to display the data items transported by the flow.



3 Click OK.

The names of the data items transported by the flows are displayed in the diagram.



4 Select *File* ► *Display Preferences*.


5 Clear the *Contents* checkbox.

6 Click OK.

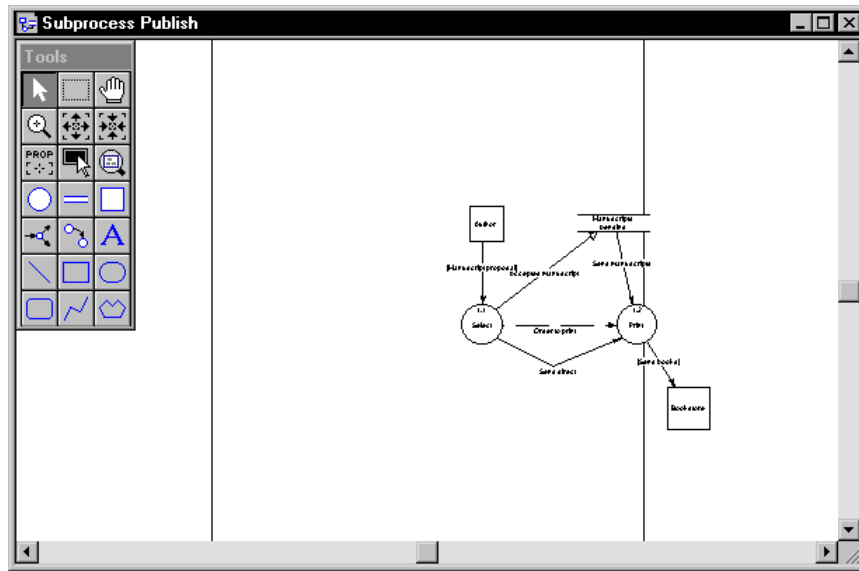
The names of data items no longer appear.

## Center the model on the page

- Where you are
- Chapter 6 Managing the Display
- Add a title box
- Change the color of the window
- Align symbols
- Arrange flows
- Display or hide flow text
- ➔ Center the model on the page
- Print the model
- Exit ProcessAnalyst

-  1 Click the *Current page* tool.

This displays the entire current page in the work area.

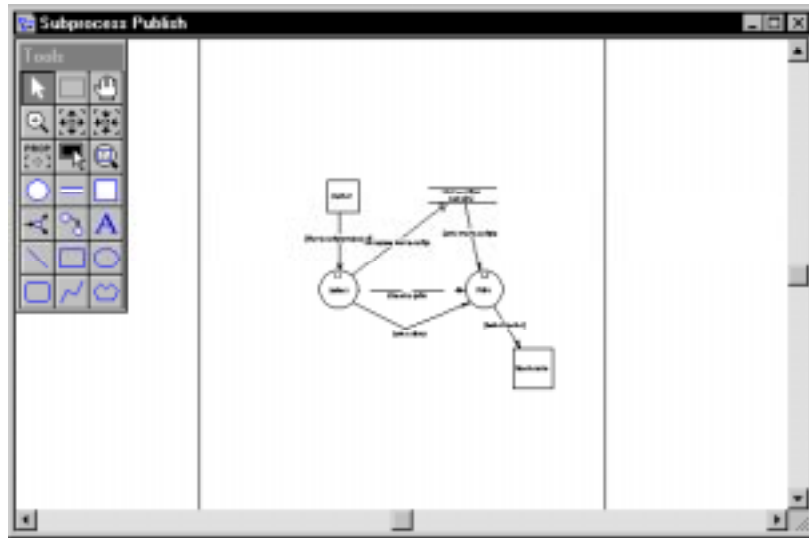


If the model runs over the page grid you can recenter it.

-  2 Select the *Grabber* tool, click the model, and hold the left mouse button.

This selects the whole model.

- 3 Move the model to the center of the page.



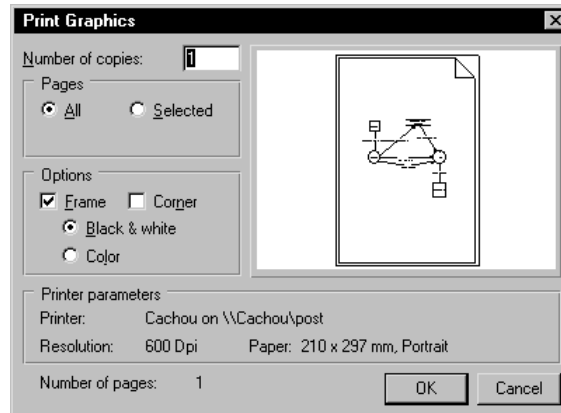
## Print the model

**Where you are**  
Chapter 6 Managing the Display  
Add a title box  
Change the color of the window  
Align symbols  
Arrange flows  
Display or hide flow text  
Center the model on the page  
➔ **Print the model**  
Exit ProcessAnalyst

You can now print the root process or the subprocess. When you print a model, you print the process which is currently displayed on screen.

- 1 Select *File* ➤ *Print Graphics*.

The Print Graphics dialog box appears.



**Select pages to print**  
When a model is spread over several pages you can deselect and select pages for printing by clicking on them. Pages selected for printing have a turned down corner. Pages not selected for printing have a simple frame.

- 2 Click *OK*.

## Exit ProcessAnalyst

Where you are  
Chapter 6 Managing the Display  
Add a title box  
Change the color of the window  
Align symbols  
Arrange flows  
Display or hide flow text  
Center the model on the page  
Print the model  
➔ Exit ProcessAnalyst

You will close the generated PAM and exit ProcessAnalyst.

- 1 Select *File*➤ *Save* from the menu bar.

This saves the PAM.

- 2 Select *File*➤ *Close* from the menu bar.

This closes the PAM window.

- 3 Select *File*➤ *Exit* from the menu bar.

You exit the ProcessAnalyst application.

You have now completed the PAM tutorial.



# Glossary

<b>business rule</b>	Written statement specifying what the information system must do or how it must be structured to support business needs.
<b>check parameters</b>	Range of data and validation rules that can be applied to data items and domains.
<b>Conceptual Data Model (CDM)</b>	Entity-relationship diagram that models the information system without considering the details of physical implementation.
<b>control flow</b>	Data flow that transmits an order that will start another process. This type of data flow can only connect two processes.
<b>data item</b>	Elementary piece of information. A data item is transformed by a process, transported by a data flow, and stored in a reservoir.
<b>data store</b>	Unit that stores the data items circulating in the model.
<b>decomposition</b>	Action of breaking down the various functions making up a process, and describing and representing these functions graphically.
<b>domain</b>	Set of values for which a data item is valid.
<b>external entity</b>	Person, place, or thing that can send or receive data items in the model.
<b>global object</b>	Object which exists for the entire model and can be reused at all levels of decomposition. Data stores, external entities and data items are global objects.
<b>local object</b>	Object which only exists at a certain level of the model and cannot be reused at another level. Processes and data flows are local objects.

<b>logical data flow</b>	Data flow that transports data between the various objects in a model. This type of data flow can connect all types of objects.
<b>object creation flow</b>	Data flow that creates one or more objects in a data store of the model. This type of data flow can only connect a process to a reservoir.
<b>off-page connector</b>	Object which is automatically created during process decomposition. It represents the flow of data between an object in the parent process and an object in the subprocess. The off-page connector displays a symbol indicating the type of object in the parent process that is the source or target of the data flow.
<b>Physical Data Model (PDM)</b>	Table-reference diagram that models the information system including the details of physical implementation.
<b>process</b>	ProcessAnalyst object which transforms the data items it receives before sending them on.
<b>process tree</b>	Presentation of all processes in a PAM in the form of a hierarchical tree structure.
<b>ProcessAnalyst Model (PAM)</b>	Functional analysis model of a system that presents the flow of data between the various objects in the system.
<b>property sheet</b>	Window that displays the properties of an object.
<b>root process</b>	Process at the top of the decomposition hierarchy.
<b>subprocess</b>	Result of decomposition of a process. A subprocess depends on another process which is higher up in the logical decomposition hierarchy of the PAM.

# Index

## A

- add
  - title box 58
- align
  - horizontal 62
  - symbol 62
  - vertical 62
- analyze
  - domain 2
  - function 1
- apply
  - background color 58
  - domain to data item 34
- arrange
  - flow 64
- attach
  - check parameter 32
  - data item to data flow 36

## B

- business rule 73

## C

- CDM 73
- center
  - on page 68
- check
  - PAM 55
- check parameter 25, 73
  - attach 32
- close
  - PAM 71
- color
  - background 58
  - default 60
  - symbol 60
  - window 61

- control flow 23, 73
- create
  - control flow 23
  - data flow 12
  - data item 25, 26
  - data store 44
  - domain 25, 29
  - external entity 10
  - object flow 46
  - process 8, 17
  - root process 5

## D

- data flow 5, 23, 74
  - attach data item 36
  - create 12
  - property 12
- data item 73
  - apply domain 34
  - attach to data flow 36
  - create 25, 26
  - define 25
  - select 36, 48
  - select data flow 48
  - store 43
- data store 43, 48, 73
  - auto fill 48
  - connect 52
  - create 44
  - data item 48
  - number 44
  - property 44
- data type
  - default 29
  - list 26
- decompose
  - process 16, 73
  - root process 15, 17
- define
  - data item 25
  - domain 25, 29

define (*continued*)  
external entity 10  
process 8, 17  
root process 5

display  
add title box 57  
align symbol 57  
color 57  
data item 66  
flow 57  
flow name 66  
manage 57  
message 55  
preference 66  
process tree 22  
domain 73  
analyze 2  
apply to data item 34  
create 25, 29  
define 25, 29

## E

external entity 5, 73  
create 10  
define 10  
highlight 58  
property 10  
shadow 58

## F

flow  
arrange 64  
control 23  
create 46  
data 5  
diagram methodology 2  
display data item 66  
display name 66  
hide name 66  
migrate 16, 21  
modify 64  
move 21  
name 12  
object creation 46, 48  
position 64  
property 23

flow (*continued*)  
text 66  
function  
analyze 1  
functional model  
methodology 2  
OMT 2

## G

Gane & Sarson 2  
global object 36

## H

hide  
flow data item 66  
flow name 66  
highlight  
external entity 58  
object 58  
title box 58

## L

list  
data type 26  
object 36  
local object 36

## M

methodology  
data flow diagram 2  
functional model 2  
Gane & Sarson 2  
OMT 2  
SSADM 2  
support 2  
Yourdon/DeMarco 2  
migrate  
flow 16, 21  
migrated flow 17

## N

number  
  data store 44  
  process 8

## O

object  
  creation flow 46, 74  
  global 36, 73  
  highlight 58  
  local 36, 73  
  shadow 58  
off-page connector 74  
OMT 2  
open  
  PAM 6

## P

PAM 74  
  CDM 1  
  center on page 68  
  check 55  
  close 71  
  define 1  
  display 57  
  functional analysis 1  
  global view 7  
  move 68  
  open 6  
  print 70  
  save 6, 71  
PDM 74  
print  
  graphic 70  
  PAM 70  
process 74  
  connect 17  
  create 8, 17  
  decompose 16  
  define 8, 17  
  hierarchy 22  
  number 8  
  property 8  
  root 5, 15  
  tree 22  
process tree 74

property

  control flow 23  
  data flow 12  
  data store 44  
  external entity 10  
  process 8  
  sheet 74

## R

root process 15, 74  
  create 5  
  decompose 15  
  define 5

## S

save  
  PAM 6, 71  
select  
  data item 36, 48  
  multiple symbol 62  
  shadow 58  
shadow  
  external entity 58  
  object 58  
  select 58  
  title box 58  
SSADM 2  
store  
  data 48  
  data item 43, 48  
subprocess 15, 74  
support  
  methodology 2  
symbol  
  align 62  
  color 60  
  multiple select 62

## T

title box  
  add 58  
  highlight 58  
  shadow 58

tree  
process 22

## **W**

window color 61

## **Y**

Yourdon/DeMarco 2