

IBM Content Manager for iSeries



Understanding Advanced Workflow

Version 5 Release 1

IBM Content Manager for iSeries



Understanding Advanced Workflow

Version 5 Release 1

Note

Before using this information and the product it supports, read the information in "Appendix. Notices" on page 27.

First Edition (May 2001)

This edition applies to Version 5.1 of and to all subsequent releases and modifications until otherwise indicated in new editions. This edition replaces SC34-3139-00

© **Copyright International Business Machines Corporation 1998, 2001. All rights reserved.**

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About This Book	v
Who Should Use This Book	v
How This Book Is Organized	v
Prerequisite and related information	v
Information included in your product package	v
Support available on the Web	vi
Operations Navigator	vi
Information available on the World Wide Web	vi
How to send your comments	vii

Chapter 1. Introduction to Advanced Workflow	1
Understanding Workflow	1
Understanding Work Processes	1
Planning a Work Process	1
Information to be Processed	2
How Information is Handled	2
Actions to be Taken	2
How Information Flows Through the Process	3
How Everything Fits Together	3
Content Manager for iSeries Workflow Components	4
Workflow Builder	4
Content Manager for iSeries Client for Windows	5
Content Manager for iSeries Server	5

Chapter 2. Workbaskets and Action Lists	7
Defining Workbaskets	7
Access List	7
Default Action List	7
Reassignment of Work	7
Listing and Presenting Work Packages	8
Maximum Number of Items in a Workbasket	8
Defining Action Lists	8
Workbasket Actions	8
Defined Actions	9
Routing Along Predefined Paths Using the Continue Function	10

Routing Along Undefined Paths	10
---	----

Chapter 3. Understanding Collection Points	13
Collection Points in Workflow	13
A Collection Point	14
Terminology	14
Events and Event Types	14
Route	14
Item	15
Customer-defined (50-99)	16
Days	16
Routes	16
Work Package Instances	17
Event Lists	18

Chapter 4. Understanding Decision Points	21
Decision Points in Workflow	21
A Decision Point	21
Terminology	21
Variables	22
Operators	23
Alternatives	23
Routes	24
Advanced Concepts	25
Work Package Instances	25
DECIDE Command	26
COPY Command	26

Appendix. Notices	27
Trademarks	29

Glossary	31
---------------------------	-----------

Index	39
------------------------	-----------

About This Book

This book shows you how to analyze your enterprise's work processes so you can create your own automated work management process, using Workflow Builder.

Who Should Use This Book

This book can help application programmers who are writing line-of-business applications for IBM Content Manager for iSeries.

For detailed information on how to create processes using the Workflow Builder, refer to the Workflow Builder's online help.

How This Book Is Organized

This book contains the following information.

- "Chapter 1. Introduction to Advanced Workflow" on page 1 explains work management concepts, including overall workflow and types of routing.
- "Chapter 2. Workbaskets and Action Lists" on page 7 explains how to use workbaskets, the various types you can choose, rules that apply, how to associate action lists, how to assign actions, and how to link action lists with decision points.
- "Chapter 3. Understanding Collection Points" on page 13 covers the concepts behind collection points and how they relate to the work management process, how and why you would use them, events, event lists, event types, and advanced concepts for handling routing and work package instances.
- "Chapter 4. Understanding Decision Points" on page 21 covers decision points and how they fit into the work management process, how and why you would use them, variables, alternatives, and advanced concepts for handling work package instances.

Prerequisite and related information

Use the iSeries Information Center as your starting point for looking up iSeries and AS/400e technical information. You can access the Information Center in one of two ways:

- From the following Web site: <http://www.ibm.com/eserver/iseries/infocenter>
- From CD-ROMs that ship with your Content Manager for iSeries order.

Information included in your product package

The *IBM Content Manager for iSeries* CD-ROM contains each publication in portable document format (.PDF).

Table 1 on page vi shows the publications included with IBM Content Manager for iSeries. When you order IBM Content Manager for iSeries, you also receive IBM Content Manager for iSeries Client for Windows. You can also separately request Client for Windows.

Table 1. IBM Content Manager for iSeries publications

File name	Title	Publication number
c2711330.pdf	<i>Planning and Installing</i>	SC27-1133-00
c2711350.pdf	<i>Getting Started with Client for Windows</i>	GC27-1135-00
c2711360.pdf	<i>System Administration Guide</i>	SC27-1136-00
c2711370.pdf	<i>Messages and Codes</i>	SC27-1137-00
c2711380.pdf	<i>Understanding Advanced Workflow</i>	SC27-1138-00
c2711390.pdf	<i>Application Programming Guide and Reference</i>	SC27-1139-00

Copying the PDF files: To copy the PDF files from the CD-ROM to your hard drive:

1. Change to the directory for the language that you are using (for example, ENU for English).
2. Copy *.PDF files to your designated hard drive directory

Installing the PDF reader: The Adobe Acrobat Reader is available from <http://www.adobe.com>. To install Acrobat Reader, follow the instructions in the Acrobat installation program or the installation text file.

Support available on the Web

Product support is available on the Web. Click **Support** from the product Web site at:

<http://www.ibm.com/software/data/cm/>

The documentation is included in softcopy on the product CD-ROM. To access product documentation on the Web, click **Library** on the product Web site.

Operations Navigator

IBM iSeries Operations Navigator is a powerful graphical interface for managing your iSeries and AS/400e servers. Operations Navigator functionality includes system navigation, configuration, planning capabilities and online help to guide you through your tasks. Operations Navigator operation and administration of the server easier and more productive and is the only user interface to the new advanced features of the OS/400 operating system. It also includes Management Central for managing multiple servers from a central server.

For more information on Operations Navigator, see the Information Center.

Information available on the World Wide Web

More iSeries information is available on the World Wide Web. You can access general information from the iSeries home page, which is at the following Web site:

<http://www-1.ibm.com/servers/eserver/series/>

To access workshops on advanced iSeries functions, use the Technical Studio, located at:

<http://www.iseries.ibm.com/tstudio>

How to send your comments

Your feedback helps IBM to provide quality information. Please send any comments that you have about this publication or other IBM Content Manager for iSeries documentation. You can use either of the following methods to provide comments:

- Send your comments from the Web. Visit the IBM Data Management Online Reader's Comment Form (RCF) page at:
<http://www.ibm.com/software/data/rcf>
You can use the page to enter and send comments.
- Send your comments by e-mail to comments@vnet.ibm.com. Be sure to include the name of the product, the version number of the product, and the name and part number of the book (if applicable). If you are commenting on specific text, please include the location of the text (for example, a chapter and section title, a table number, a page number, or a help topic title).

Chapter 1. Introduction to Advanced Workflow

The workflow functions of Content Manager for iSeries let you control the flow and performance of work in your business. When users work with documents and folders, they must make decisions on what actions to perform. Workflow aids in this process by letting you determine in advance how you want the work to be performed.

Understanding Workflow

Most business operations can be characterized as a set of interrelated processes. Work flows from one employee to another, and from one department to another. Some simple processes might require only a few steps, while more complex processes involve a number of employees in different departments.

Workflow lets you move work through a process, gather documents at collection points, and monitor work throughout the process. For example, a mortgage loan company receives large volumes of loan applications in the mail. During the approval process, the company receives follow-up documents such as credit reports and appraisals. Employees spend several hours each day opening, sorting, filing, and monitoring applications, as well as collecting pertinent documents for final approval.

Documents are moved from one employee to another as they are received and checked. As the loan package is completed, it might be handled by employees in more than one department.

Understanding Work Processes

As in the above mortgage loan company example, most enterprises that handle documents perform some or all of the following tasks:

- File documents for later retrieval
- Collect documents, forms, and information from different sources, then deliver these documents to somewhere to be processed
- Match incoming mail with documents currently being processed
- Hold some work until another department or group is finished with its part of the work.

A *work process* represents the flow of work in Content Manager for iSeries. It describes the actions that can be performed on a group of one or more documents and the path this group takes through the workflow process.

A work process reflects work the way that it is performed, with a clearly defined scope and boundaries. It defines the sequence of activities and tasks, and the connections and relationships among those activities and tasks. A work process determines the criteria that are used to make decisions on the flow of work and the points at which those decisions are made.

Planning a Work Process

Before you begin to define a work process, you must determine the work that your business performs, in addition to where and how it is performed.

What is the final product? The final product might be the result of all the work accomplished by your business, by one department in your business, or by certain employees from different departments. For example, the final product of the mortgage loan company is the letter sent to the applicant approving or rejecting the loan application.

Analyze the information that must be processed to produce the final product, determine the actions that must be performed and where they are performed, and decide how you want the information to flow through the work process.

Information to be Processed

Consider the information that must be handled. What types of input support the final product? What are the specific documents that must be processed?

A *work package* consists of zero or more items that can be moved together from one place to another in a work process. An *item* can be either a document or a folder. For example, a mortgage loan company initially receives loan applications and later receives follow-up documents, such as credit reports and appraisals. These documents, within a folder, form the work package.

How Information is Handled

Who can best handle each step of the process? For example, a loan officer might verify that a loan application is complete, then file the application until a certain document is received from the applicant. When the document arrives, the loan officer might be responsible for matching the document with the application, as well as for processing that document.

The loan applications could be grouped in a *workbasket* that is accessible to a number of loan officers. A *workbasket* is a point in the process where work packages are handled. Workbaskets could be defined within the work process to handle each part of the loan process, such as credit reports and home appraisals. For more information on workbaskets, see “Chapter 2. Workbaskets and Action Lists” on page 7.

Actions to be Taken

Consider what actions must be taken on the contents of a workbasket. For example, a loan officer can accept an application or reject it as incomplete. An *action list* defines the actions that a user can perform on the work package. An action list can describe the following:

- Selectable options that are available to the user
- Customer-defined options.

Depending on whether an application meets the initial credit requirements, a loan officer could select one option to continue the application through the work process, or another option to reject the application.

When a user works with a workbasket in Content Manager for iSeries, only those actions that you have previously defined for the work package at the workbasket are available.

For more information on workbaskets, see “Chapter 2. Workbaskets and Action Lists” on page 7.

How Information Flows Through the Process

Consider how you want information to flow through your process. For example, when is the initial loan application reviewed? What supporting documents are needed to move on to the next step in the process? What criteria determine whether an application is accepted or rejected? This flow of information is the basis of your work process, which defines the way users perform their work. A work process consists of decision points, collection points, and the paths that guide the work throughout processing. Where does the input originate? Your work process must begin at some point. In the mortgage loan company, the loan application submitted by the applicant is the document that starts the work process.

Decision points determine the path of the work based on a set of values you specify. For example, a loan application can be rejected if the applicant's salary is insufficient. *Collection points* manage work packages that need to wait for additional documents, the completion of another action, or the occurrence of business events. For example, the loan application cannot be approved or rejected until the credit report is received and processed.

One decision point in the mortgage loan example might move an application to a *collection point* to wait for credit reports and appraisals. You can assign time guidelines to the collection points to wait for documents and continue them along the path when they arrive, or signal that a document has not been received. For example, if a credit report is valid for a certain length of time, the collection point might wait that length of time for other supporting documents, such as a home appraisal, after it receives a credit report.

Workpackages within a process can continue along parallel paths to several collection points at one time. In this way, documents that arrive at different times do not affect other parts of the path.

When all of the documents are received, the work package can continue along the path to a final point—for example, loan approval.

How Everything Fits Together

After you analyze the information that you want to process, determine the actions that you want to perform, and decide how you want the information to flow through the work process, you are ready to create a work process diagram, which is the graphical representation of your work process.

A work process diagram shows how work moves through the various activities in the process, noting what tasks the activity involves. It describes the flow, the main elements, and the key measuring points of a work process.

The work process diagram in Figure 1 on page 4 shows how information might flow through a mortgage loan company.

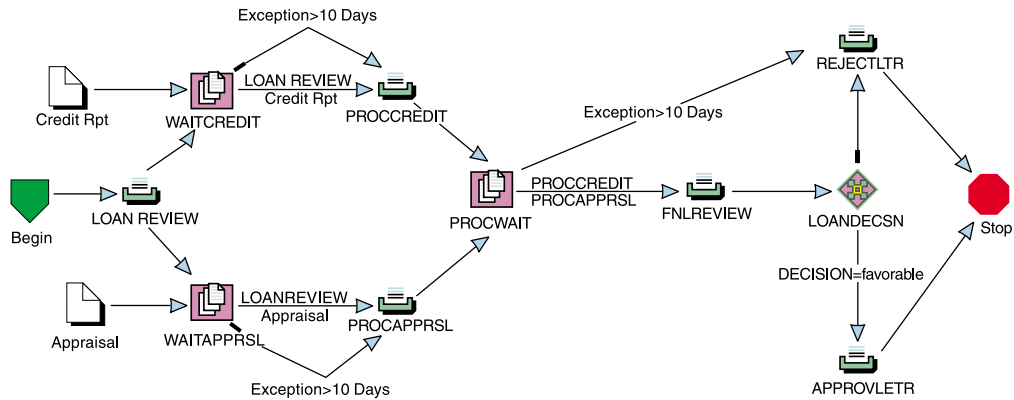


Figure 1. Sample Work Process Diagram

Each symbol in the work process diagram represents a point at which work is done, documents are collected, or decisions are made. A mortgage loan application must be reviewed, supporting documentation must be collected, and the application must be approved or rejected, depending on certain criteria.

Each symbol also has information associated with it, which can describe the work that is done, the information that is collected, or the decisions that are made.

Content Manager for iSeries Workflow Components

Workflow Builder

The Workflow Builder of Content Manager for iSeries is a versatile, easy-to-use graphical tool with which you can define and optimize the work process of a workgroup, department, or enterprise. Using icons, a user can construct a comprehensive, easy-to-understand graphical representation of a work process.

Before you use Workflow Builder to create a process, you need to define your privilege sets, access lists, users, index classes, and workbaskets on the Content Manager for iSeries server.

You can use Workflow Builder to create action lists, and work processes. You must log on to Content Manager for iSeries and check in any new or updated action lists and work processes before Content Manager for iSeries can use them. The new or updated data then becomes part of the Content Manager for iSeries database.

When you check in an action list or work process, you permanently store the new or updated data associated with it on the Content Manager for iSeries server. When you check out an action list or work process, Workflow Builder queries the Content Manager for iSeries server, stores a copy of the data on your workstation, and marks it as being checked out to you, preventing anyone else from updating it until you are finished.

Although you can use Workflow Builder to build action lists and work processes, you cannot use Workflow Builder to run the work process. You must run the work process you create with Workflow Builder using the Content Manager for iSeries Client for Windows®.

Note: If you modify an existing process, the changes made will only affect new work packages started on the process. The changes will not affect work

packages already on that process. Similarly, any changes made to an action list only affect new work packages or new items in the associated workbaskets.

Content Manager for iSeries Client for Windows

The Content Manager for iSeries Client for Windows application provides an interface that enables you to bring documents into the system, view and work with them, and store and retrieve them.

The client provides access to documents and folders that are in workflow through workbaskets. A workbasket, which is defined using Content Manager for iSeries system administration, is a place where you perform work on items. For example, it can be your in-basket, a shared stack of sorted invoices, or a table of correspondence. When a folder or document is at a workbasket, it is available for processing.

When a user works with the contents of a workbasket, they do not necessarily need to know where, or even whether, the workbasket exists within a process. From his perspective, the contents of the workbasket just represent specific work to be done. When a user completes work on an item, he simply continues the item on to the next step of the process.

The Content Manager for iSeries Client for Windows is not aware of the details regarding a process definition. The process definition, created by Workflow Builder, is stored in the database on the Content Manager for iSeries server. When a user continues work on a process, the request to route the item is sent to the server. The server handles routing the item to the next location in the process.

The actions a user may perform at a workbasket are controlled by an action list. When items are worked at a workbasket, the client menus and toolbar will be modified to reflect available actions based on the action list definition.

Note: When changes are made to an action list or process using Workflow Builder and checked into the host, the user needs to logoff and then logon again to the Content Manager for iSeries Client for Windows in order to pick up the changes.

Content Manager for iSeries Server

The Content Manager for iSeries server provides library, object and workflow services. The server maintains library and workflow information and manages the storage of Content Manager for iSeries objects.

Work process and action list definitions created using the Workflow Builder are maintained in the Content Manager for iSeries database on the server.

The Content Manager for iSeries server also provides the *run-time* workflow services for the Content Manager for iSeries Client for Windows. Services are required to route and keep track of items within a work process.

Chapter 2. Workbaskets and Action Lists

A *workbasket* can be thought of as a queue of work you create for one or more employees to use or as the in-basket from which employees take work. Work assigned to a single workbasket can be worked on by as many employees as you have assigned to that workbasket.

Defining Workbaskets

A workbasket definition includes the rules that govern the presentation, status, and security of its work packages. You specify the rules for each workbasket at the same time you create the workbasket. Workbaskets are defined on the Content Manager for iSeries server using Profile Maintenance. See the *IBM Content Manager for iSeries: System Administration Guide* for a complete description of defining workbaskets. The workbasket definition includes the following:

- User workbasket
- Access List
- Name of Default Action List
- Work Reassignment
- Listing and Presenting to Users
- Number of Items

Access List

An access list consists of one or more individual user identifiers or user groups and the privilege set associated with each user identifier or user group. The privilege set is used to define a user's authorization to access or perform certain tasks on objects stored in Content Manager for iSeries. The access lists are used to control user access to items in a workbasket.

Default Action List

The name of the action list to be used as a default with this workbasket is the default action list. This action list is used if the user does not select a different action list to associate with the workbasket and the work node.

Reassignment of Work

When an employee's day-to-day decisions are the basis for special handling or exception processing, you might want to let the employee reassign a work item to another workbasket for special handling. You can do this by specifying that the receiving workbasket can have work reassigned to it. For normal, day-to-day routing decisions along predictable paths, use the Continue function to move information to the next workbasket.

You can also prevent work from being reassigned to a specific workbasket. For example, it might not make sense to let employees reassign work to a workbasket where indexing is performed.

Listing and Presenting Work Packages

Work at a workbasket can be user-selected or system-assigned. For system-assigned workbaskets, work can be presented in first in, first out (FIFO); last in, first out (LIFO); or priority order.

Another option for controlling the presentation of work at a workbasket, is to define a workbasket as a user workbasket. (For more information about defining workbaskets, see the System Administration Guide.) When a user is working with this type of workbasket, access is granted to that user:

- to work packages owned by that user.
- to work packages which are unassigned.

This feature is overridden if a user has the administrative privilege for workbaskets.

Maximum Number of Items in a Workbasket

You can specify the maximum number of items that you want a workbasket to contain before the system performs overload processing. When an overload occurs, the user will be notified, and optionally, the overload user exit will be invoked.

Defining Action Lists

An *action list* is an approved list of the actions that a user can perform while working with items in a workbasket . When you create a work process, you assign a list of actions to each workbasket. The action list must be a comprehensive list of all actions performed on a work package or its contents. If an action is not listed on the action list for a workbasket, the employee cannot perform that action.

Note: If the user's privilege set does not allow an action, the action is disabled for that user on the Content Manager for iSeries Client for Windows menu.

Use the Content Manager for iSeries Workflow Builder to define your action lists.

Workbasket Actions

Each workbasket that you define can have identical actions, or each workbasket can have a different set of actions. You specify which actions you want your employees to perform on the work package or its contents by creating action lists. If you do not specify a specific action in an action list, the user of the workbasket associated with that action list cannot perform that action. For example, if you do not specify a CONTINUE action for an action list, Content Manager for iSeries users cannot send a work package along to the next step in a predefined process.

You can customize action lists. Specifically, you can tailor the menu prompt, associate a toolbar icon with each action, and define a shortcut key for the action. Actions are applicable to both documents and folders, and you can define them accordingly. For example, you can use an action list to restrict what a user can do. You can define the delete action for only documents, so that users are not allowed to delete folders.

When you open a workbasket and select an item in the IBM Content Manager for iSeries Client for Windows , defined actions in the action list are available on the client menus and toolbar.

Defined Actions

Following are the definitions of the actions, in alphabetical order, that you can use in action lists:

Activate	Releases a suspended work package according to user-specified reasons.
Add to folder	Adds an item to a folder.
Add to new folder	Adds an item to a new folder.
Add to workbasket	Adds a copy of an item to a workbasket.
Change process	Removes the work package from the current process, regardless of the location of the work package within the process and creates a new work package (which contains the same information as the current work package), and assigns the new work package to a new user-specified process.
Continue	Removes the work package from the current workbasket and releases the work package to continue on to the next predefined step of the process.
Delete	Removes an item from the Content Manager for iSeries system.
Delete and next	Deletes the current item and displays the next item. The item is deleted from the Content Manager for iSeries system. This action only applies to work packages and system-assigned workbaskets.
Export	Exports documents stored in Content Manager for iSeries, for use by another application.
Next	Displays the next highest priority item in a system-assigned workbasket.
Notes	Allows you to create new or edit notes associated with an item.
Priority	Allows you to view and change the priority associated with an item at a workbasket.
Reassign	Adds the work package to a specified workbasket and removes the work package from the current workbasket.
Reindex	Changes the index class and key fields associated with an item.
Remove from folder	Removes the item from the folder.
Remove from process	Removes the work package from the current process, regardless of the location of the work package within the process.
Remove from workbasket	Removes the item from the workbasket and the instance of the item from the process.
Save and next	Saves any indexing changes, notes log changes, or annotations that you have added to the current

	item, and displays the next item. This action only applies to work packages and system-assigned workbaskets.
Set variable	Allows you to set the value of a new or existing work package variable.
Start process	Creates a new work package, which contains the same information as the current work package, and assigns the new work package to a user-specified process.
Suspend	Holds a work package at a workbasket until stated criteria have been satisfied.
User-defined option	Sends information about the current work package to a user exit, from which custom code can be executed.
Workflow information	Displays all instances of an item with workflow.

The following actions permit work to continue. Your action list probably needs at least one of these:

- Continue
- Add to workbasket
- Reassign
- Change process
- Remove from process
- Remove from workbasket
- Next
- Save and next
- Delete and next
- User-defined option (user exit).

Routing Along Predefined Paths Using the Continue Function

One of the most important features of action lists that you will want to exploit is the feature supported by the continue function. This function lets a user select an option that indicates the work is done, releasing the work package so it can continue along the predefined path. Employees can use the continue function extensively in workbaskets where they have to make decisions about routing. However, you no longer need to supply users with lists of people or routes to send the information on to. You can set up an action list with multiple continues, each with its own prompt label, which will send the work package to the correct group of users, using the continue function in conjunction with a decision point.

For example, you could define two continue actions, one with an **Accept** prompt and another with a **Reject** prompt. Based on the option the user selects, you can cause the work package to be sent along different routes within the process.

Routing Along Undefined Paths

Even in the best defined process, a work package might need to be sent along a path that has not been previously defined in your current work process. You can use the reassign action to assign a work package that is currently on a predefined process to another workbasket that is not defined as the next step in your process. Once continued from its reassigned location, a work package continues along its

original process definition path. When reassigned, a work package is not removed from its original process. You can use this function to correct human mistakes in a process.

For example, consider a process that is defined to send work packages from workbasket A to B to C. A user who selects a work package from workbasket B notices that the work needs management approval. The user reassigns the work package from workbasket B to workbasket manager. When the manager completes that work and uses the continue function, the work package continues to the next step of the process that follows workbasket B, in this case, workbasket C. If the work is returned to workbasket B, the manager could reassign the work package to workbasket B. A work package can be reassigned as many times as necessary to workbaskets inside and outside of its current process. The work package always remains associated with its current process. Following a reassign function, the first time a user selects a continue function, work continues to the next step in the process that would normally follow the step from which the work package was first reassigned.

Chapter 3. Understanding Collection Points

You can use a collection point to collect information from different sources before passing the work package on in the process. A collection point holds and manages work packages that cannot be processed completely until additional information has been received. A collection point performs no action. It merely represents the fact that a work package is waiting for one or more events to occur. The work package could be waiting for a single event, two, three, or even more events of different types. A collection point waits for events to occur. (For more information on the types of events that can be specified in a collection point, see “Event Lists” on page 18.) The three most common types of events are based on the following:

- | | |
|---------------|--|
| Items | The collection point waits for the arrival of documents or folders of a specific index class or classes to be added to a particular folder work package. |
| Routes | The collection point waits for processing to occur in a previous step (workbasket). |
| Days | The collection point waits for a specific number of days. In some work processes, several documents of different types might all have to arrive before any of them can be worked. The documents that the collection point is waiting for comprise an event list. For example, in a loan application, you might need to have a credit appraisal along with the loan application before any formal work can be done on the loan application. The method to represent the fact that several documents must be together before either of them can be worked on is called the collection point. |

Then, after the event that you specified has occurred, the work package is sent along the processing path that you set up. If the specific event has not occurred in a specified amount of time, the work package is processed along another path that you also set up. This alternative path is referred to as the *exception path*. The criteria for sending the work package along the exception path is based on the number of days specified for the work package to wait before it continues through the process.

Collection Points in Workflow

Collection points exist within processes. A process, within workflow, is a set of routes, workbaskets, collection points, and decision points that are organized in a specific way and through which work packages move from start to finish. Figure 2 shows an example of a simple process diagram.

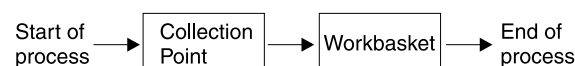


Figure 2. A Very Simple Process with a Collection Point

The example is simple to keep the meaning of the diagram fairly clear. A work package which enters this process first goes to the collection point. When it leaves the collection point, it continues to the workbasket. Finally, when the work package leaves the workbasket, it has completed the process.

A Collection Point

A collection point is a part of a process where work packages can be sent. But, what is a collection point, why would you want to send work packages to it? And when does a work package leave a collection point once it is there? A collection point is a location in a process where work packages wait until one or more events have occurred.

By sending a work package to a collection point, you stop it from continuing through the process until some set of necessary events have occurred. When all events in a set have occurred, the work package automatically leaves the collection point and continues on to the next step in the process. A set of events is referred to as an *event list*.

Terminology

Following are a few terms that are used when discussing collection points. These terms can be quite confusing. It is important that you notice how they relate to each other.

Matching an event Finding an outstanding event at a collection point that has the same work package, event type, and match criterion as the event being used for comparison.

Satisfying an event Removing a matched event from the outstanding events at a collection point. This is always done when an event is matched.

Releasing a work package from a collection point Removing a work package from a collection point when all of the events in an event list have been satisfied. The work package goes to the route specified as the completion route for that event list.

Activating a work package at a collection point Creating the outstanding events for a work package at a collection point when an attempt to match an event has failed.

Note: This is not the same as the Activate action discussed in “Defined Actions” on page 9.

Events and Event Types

Events have been mentioned throughout this introduction to collection points, so it might be a good idea to explain just what they are. An event consists of an event type and a match criterion. Event types are actually two-digit numbers, but they are usually referred to by words which describe what events of that type are supposed to wait for. Valid event types are route, item, customer-defined, and days.

Route

A route event indicates that the collection point is waiting for an *instance* of a work package to enter the collection point from a certain workbasket. There is no route listed in the collection point event list in Workflow Builder. Instead, the workbasket name is shown as a choice in the event list to represent the route.

Item

An item event indicates that a folder work package is waiting at the collection point for items of a specific index class to be added to it. You may alternatively specify a work package variable in place of an index class value. The variable's actual value is evaluated at the time the collection point is activated, allowing run-time flexibility to the collection point definition. Along with the item specification, you can specify the number of times that the item event should occur. You can specify the number of items with a numeric value or with a work package variable. The variable's actual value is evaluated at the time the collection point is activated, allowing run-time flexibility to the collection point definition.

If an item node is assigned the *ANY index class, an item of any index class will match the associated collection point event. Be careful when using the *ANY index class for your item event. Consider the following examples.

Example 1

An *ANY collection point is shown as Figure 3.

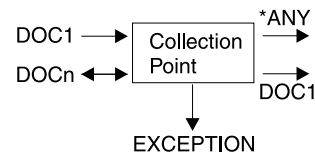


Figure 3. An *ANY Collection Point

In Figure 3, which route do you think will be taken when DOC1 arrives at the collection point: *ANY, DOC1, or both? The answer is the route that is defined first. What is defined first? When Content Manager for iSeries matches events, it simply attempts to match the incoming event to the unresolved events in the order they were drawn. If DOC1 and *ANY are defined as collection point events, and DOC1 arrives at the collection point, either the DOC1 or *ANY event is matched depending on the order in which the connections were drawn using Workflow Builder.

Note: Each outbound route from a collection point has an event list associated with it. There is never more than one event list matched for a collection point. Each outbound route associated with the satisfied event list is started.

Example 2

You can define an event list that includes more than one item event. If you assign one item event of *ANY index class and another that requires a specific index class, it is ambiguous as to which collection point event will be satisfied when an item of that index class is added to a folder work package.

For example, define a process that includes a collection point node. Assign both *ANY and DOC1 item events to an outbound connector from this collection point node. If a DOC1 item is added to the folder at the collection point, it waits for a second item to arrive. If a second item is added to the folder at the collection point that is NOT a DOC1 item, the two items now at the collection point may or may not take the *ANY + DOC1 route. If the first DOC1 item satisfied the *ANY event, then the route is not taken until a second DOC1 item event arrives. If, on the other hand, the first DOC1 item satisfied the DOC1 event, then the route is taken when an item of any index class arrives at the collection point. Either case is possible.

Customer-defined (50-99)

A customer-defined event can be whatever the system administrator wants it to be. Processing for the customer-defined event types (50-99) is very similar to that of the *item event type*. The logic flow is identical. The customer-defined event types use the **SimWmMatchEvent** application programming interface.

In Workflow Builder, you can add a customer-defined event to your process using the user-defined event node.

Days

A days event indicates that the collection point is waiting until a work package has been in the collection point for a certain number of days. For the days event type, the code checks to see if it can satisfy the expired time check. If so, the work package instance is released and the exception route started.

Routes

Take another look at the process diagram that you saw earlier. See Figure 4.

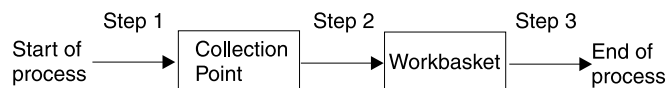


Figure 4. A Very Simple Process with a Collection Point Showing Steps

A work package that enters the process goes to the collection point. That is the first step in the process. In the second step of the process, the work package goes to the workbasket. The third and final step in the process is for the work package to leave the process. As you can see, each arrow in the diagram represents one step. In work management, steps are always recorded sequentially in groups.

A route consists of one or more steps. When a work package enters a process, it is actually entering the first route of that process. As the work package proceeds through the route, each step in the route is performed using that work package, until the end of the route is reached.

One way that a route can end is with a step which sends the work package to a collection point. This type of step, called a COLLECT command, must always be the last step in a route, since a work package leaving a collection point always goes to the new route specified in the satisfied event list of the collection point.

The process diagram in Figure 4 represents the true structure of a process. In this example, two routes are actually needed, since a collection point always terminates a route. The example shown in Figure 5 on page 17 shows the process diagram again, this time followed by the actual process objects (routes, a collection point, and a workbasket, in this case) which make up the process.

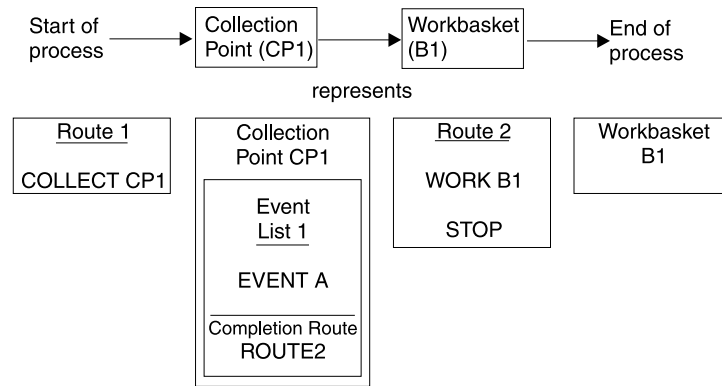


Figure 5. Simple Process with Process Objects for a Collection Point

A work package which enters the process goes to the first step of ROUTE 1, which is a command COLLECT to send the work package to the collection point called CP1. When event A occurs, the work package leaves the collection point and continues to ROUTE 2. The first step of ROUTE 2 sends the work package to workbasket B1. When the work package exits the workbasket, then the next step in ROUTE 2, the STOP command, is executed. STOP stops the movement of the work package through the process, ending the process.

In Workflow Builder, you can define a collection point node and collection point connector to specify this information. Workflow Builder generates the appropriate COLLECT command for you.

Work Package Instances

The simple process diagram that has been used until now involves a work package moving through the route from start to finish in a linear fashion. Figure 6 shows a process that contains a parallel route with collection point coordination.

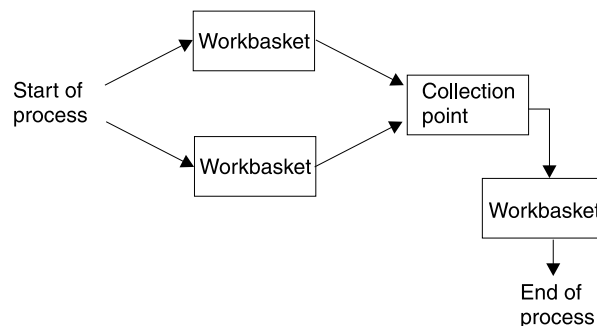


Figure 6. A Process with a Parallel Route with Collection Point Coordination

A work package entering this process immediately splits, continuing to two workbaskets at the same time. A pointer to the work package moves through the system. Such pointers contain the names of the work packages they represent. If information about a work package is needed, the system uses the pointer to determine which work package information to examine. These pointers are called instances of a work package. A work package can have up to 99,999 instances, which means that a work package can be in up to 99,999 different places at once during a process.

One of the event types of collection point events is the route event type, which indicates that a collection point is waiting for an instance of a work package to enter the collection point from that route. The concept becomes clear when the above process is examined. The work package splits into two instances, but is reassembled at the collection point so that only one instance leaves the collection point. To do this, the collection must be able to wait for individual instances of a work package. When all instances arrive at the collection point, the work package continues to the next step in the process.

See Figure 7 for a process diagram.

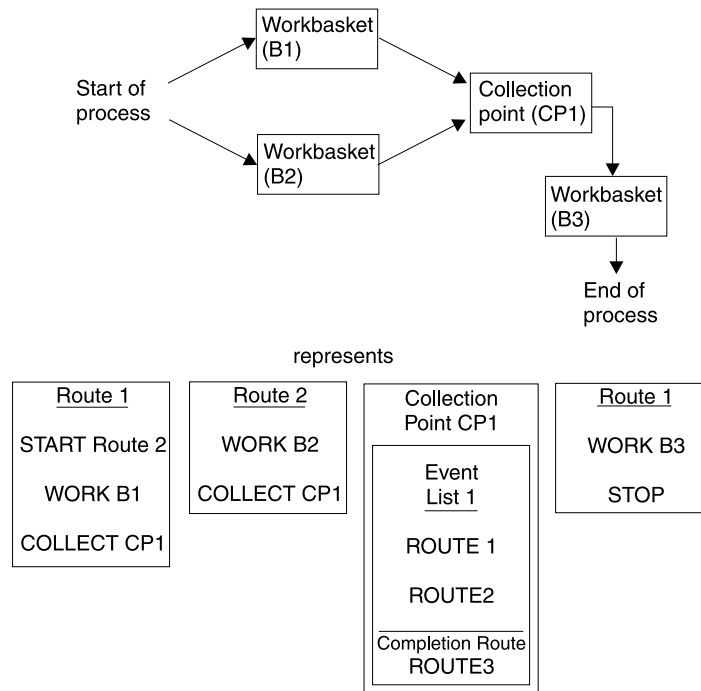


Figure 7. A Process with a Parallel Route and Objects with Collection Point Coordination

As illustrated, the second instance of a work package is created by ROUTE 1 when the START ROUTE 2 step is performed. Using START is the only way to create additional instances of a work package. Because the new instance must start on a different route, a collection point must be able to identify different instances by looking at the route from which each originates. Workflow Builder generates the appropriate START ROUTE command for you.

The collection point waits for a work package instance to arrive from ROUTE 1 and ROUTE 2. When both events are satisfied, the collection point releases the work package (actually, a single instance of the work package) to ROUTE 3, which ends the process after sending the work package to workbasket B3 for some final work.

Event Lists

A set of events in a collection point is called an event list. An event list contains one or more events, all of which must be satisfied for a work package to be released from a collection point.

A collection point can have from 0 to 999 event lists. It is possible to have no event lists in a collection point, because there is a special exception condition which always let a work package exit a collection point. The exception condition is the number of days a work package waits at the collection point. If a work package has not already left a collection point when that number of days has passed, then the work package leaves the collection point and goes to the route specified as the exception route for that collection point.

Logically, think of collection point evaluation this way. Within an event list, all events must be satisfied (logical AND). When deciding which event list is satisfied, each is evaluated in order until one is found to be satisfied. The order depends on the order the connections were drawn using Workflow Builder. The last connection drawn is evaluated first.

A work package is released from a collection point as soon as all of the events in any one event list have occurred. Event lists within a collection point are not related to each other. Figure 8 shows a collection point and its associated event lists.

Collection Point CP1

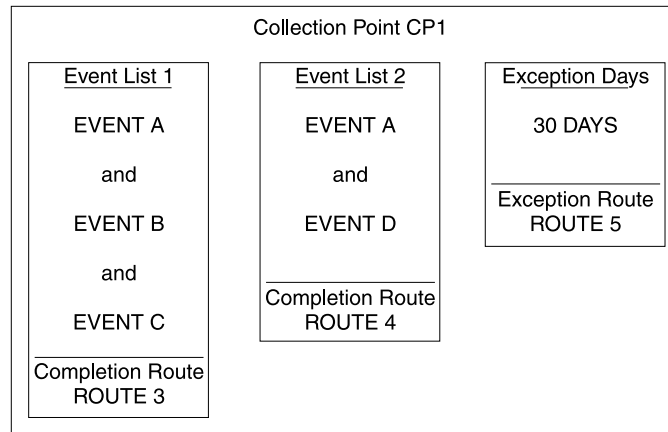


Figure 8. A Collection Point

Suppose a work package is waiting at the collection point shown in Figure 3 on page 15. If events A, B, and C are all satisfied for the work package, then the work package leaves the collection point and goes to ROUTE 3. If events A and D are satisfied, then the work package continues to ROUTE 4. If A, B, and C are not satisfied, and if A and D are not satisfied, then 30 days after its arrival at the collection point, the work package continues to the exception route, ROUTE 5.

Note: Event A occurs in two event lists. This is perfectly within the rules of collection points.

Chapter 4. Understanding Decision Points

Decision points let you make routing choices for the work performed by users. You can make these choices based on line-of-business information, workflow information, the action selected by the user, and so on. For each decision point, you define at least one alternative and an exception route.

When a work package enters a decision point, the alternatives are checked for a match. The alternatives that are defined in a decision point are checked sequentially according to the order the connections were drawn using Workflow Builder. The last connection drawn is evaluated first. The first match found is processed. If there is a match, the work package is sent to the route specified in the first matching alternative. If there are multiple matches, and the decision point was defined with an indication to start multiple routes, the work package is sent to the route specified in each matching alternative. If there are no matches, the work package is sent to the exception route specified.

Decision Points in Workflow

In workflow, a process is a set of routes, workbaskets, collection points, and decision points which is organized in a particular way, and through which work packages move from beginning to end. The easiest way to describe a process is by drawing a process as a diagram. Figure 9 shows an example of a simple process diagram with a decision point.

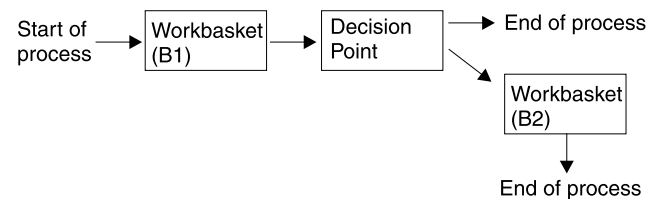


Figure 9. A Very Simple Process with a Decision Point

The example is simple to keep the meaning of the diagram fairly clear. A work package that enters this process first goes to the workbasket B1. When the work package leaves the workbasket, it is sent to the decision point. At the decision point, the work package can be sent to another workbasket or the process ends.

A Decision Point

A decision point is a location in a process where work packages are sent to different routes depending on criteria specified by the user. By sending a work package to a decision point, you are asking the system to start a route or routes after analyzing the values of variables associated with that work package. Unlike collection points, there is no waiting that takes place at a decision point.

Terminology

The following are a few terms used when discussing decision points.

- *Alternatives* are the variable, value, and route combination that determines where the work package goes when it leaves the decision point and what condition caused it to take a particular route.

- *Variable* is a 1- to 10-character alphanumeric name. There are four categories of variables:
 - system
 - customer-defined
 - library
 - external application
- *Value* is a 1- to 40-character alphanumeric string associated with a variable name. For example, variable *PRIORITY could have a value of 981231.

Variables

A decision point alternative match occurs when a work package entering a decision point meets the conditions (the variable and its value) that you define for an alternative. The variable can be one of four categories:

- system
- customer-defined
- library
- external application

System variables

Read/write system variables

- *ACTION** Lets you determine what happens when a user selects a given action at a workbasket before sending the work package to the decision point. The subsequent route for the work package is based on the action the user previously performed on the work package.
- *PRIORITY** Lets you assign a higher or lower priority to the alternative you are working on. For example, 2 has a higher priority than 1. The subsequent route for the work package is based on the priority value of the work package.

Read-only system variables

- *USER** Lets you determine the last user to work (get) a work package.
- *DATE** Lets you determine the date on which the work package was last worked.
- *TIME** Lets you determine the time at which the work package was last worked.
- *PROCESS** Lets you determine the process to which the work package is assigned.
- *OWNER** Lets you determine the owner of the work package.
- *LOCATION** Lets you determine the last location (workbasket or collection point) of the work package.

Customer-defined variables

Let you define your own variables and values for work packages. The

subsequent route for the work package is based on the value of the customer-defined variable. The customer-defined variable name is 1 to 10 characters, and the variable name cannot start with an asterisk (*). The value for this variable can be up to 40 characters.

Library variables

Let you utilize the indexing information associated with a work package. The variables are *INDEXCLS and the short names for each key field name within the index class. The library variables are read-only variables.

External Application variables

Let you use information outside the Content Manager for iSeries application, on the host system. A user exit is called when one of these variables is recognized in processing a decision point. An external application variable is recognized when the processing detects a variable name which is neither in the work package variable file, nor in a process, index class, or key field variable.

Operators

Decision points will support the specification of numerous operators. You can specify any of the following when defining an alternative:

- equal to
- greater than
- less than
- greater than or equal to
- less than or equal to
- not equal to

Alternatives

The diagram in Figure 10 illustrates what a decision point looks like.

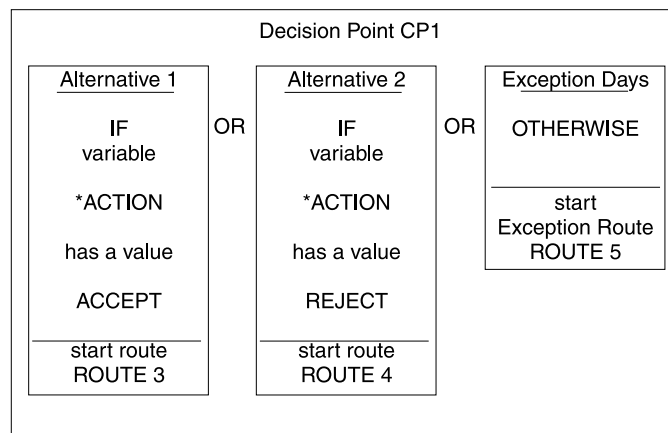


Figure 10. Inside a Decision Point

When the work package arrives at the decision point shown in Figure 10, the work package continues to ROUTE 3 if the variable *ACTION has a value of ACCEPT. If the variable *ACTION has a value of REJECT, then the work package continues to ROUTE 4. If *ACTION is neither ACCEPT nor REJECT, then the work package continues to ROUTE 5.

Routes

Take another look at the process diagram you saw earlier in Figure 9 on page 21. A work package which enters the process goes to the workbasket B1. That is the first step in the process. In the second step of the process, the work package goes to the decision point. In the third step in the process, either the work package ends the process or the work package continues to workbasket B2. The decision point decides the path flow of the work package. Each arrow in the diagram represents one step. In workflow, steps are always recorded sequentially in groups. A group of steps is called a route.

A route consists of one or more steps. When a work package enters a process, it is actually entering the first route of that process. As the work package proceeds through the route, each step in the route is performed using that work package, until the end of the route is reached.

One way that a route can end is with a step which sends the work package to a decision point. This type of step, called a DECIDE command, must always be the last step in a route, because a work package leaving a decision point always goes to the new route specified in the matched variable/value alternative of the decision point.

There are two route commands that are related to decision points. The first of these is the DECIDE command. When a route processes a DECIDE command step, it sends the work package to the decision point. The other route command is the COPY command. This allows you to assign a value to a variable in the route definition. You can assign a value to the system defined variables *ACTION and *PRIORITY as well as any customer-defined variables. If you use the COPY command to assign a value to an undefined customer-defined variable, the system creates that variable and assigns the specified value.

In Workflow Builder, you can use the assign value node to assign a value to a variable. Workflow Builder generates the appropriate COPY command for you.

The process diagram in Figure 2 on page 13 represents the true structure of a process. The example in Figure 11 on page 25 shows the process diagram again, this time followed by the actual process objects (routes, a decision point, and two workbaskets, in this case) which make up the process.

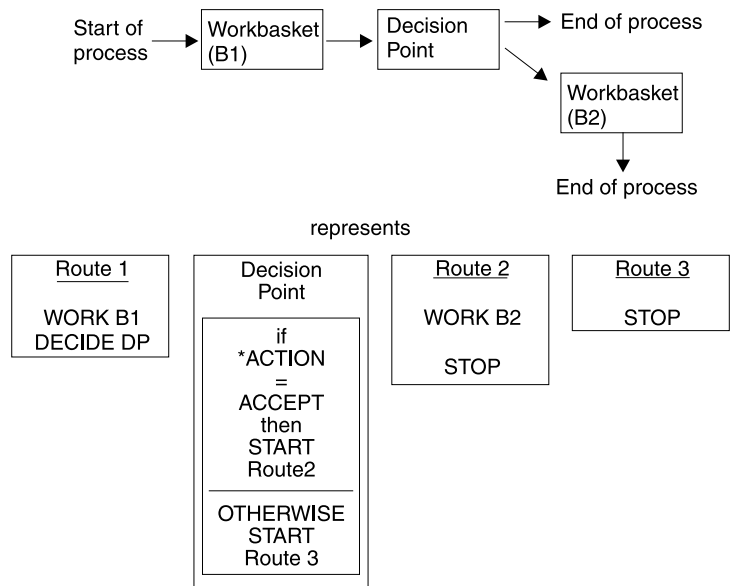


Figure 11. A Very Simple Process with a Decision Point and its Objects

Advanced Concepts

The following are a few more work management concepts that are crucial to comprehending all the nuances of decision points.

Work Package Instances

The simple process diagram we've been using until now involves a work package moving through the route from start to finish in a linear fashion. But what about the process shown as Figure 12?

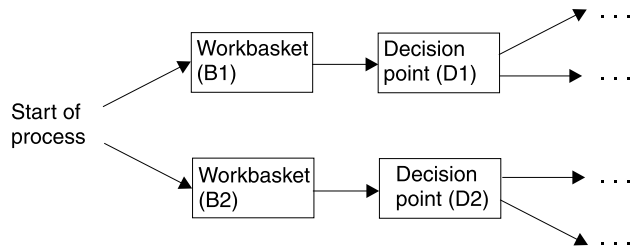


Figure 12. A Process with a Parallel Route and Multiple Decision Points

A work package entering this process immediately splits, going to two workbaskets at the same time. Actually, pointers to work packages, not work packages, move through processes. Such pointers contain the names of the work packages they represent. If information about a work package is needed, the system looks at the pointer to find out which work package to examine. These pointers are called instances of a work package. A work package can have up to 99,999 instances, which means that a work package can be in up to 99,999 different places at once during a process.

It is important to note that variables are associated with a specific work package ID and instance. In a completely serial process, there is only one set of variables

for a work package. In the above process, each parallel route has its own set of variables and values associated with it, so there are two sets of variables for the work package. Refer to the diagram show as Figure 13.

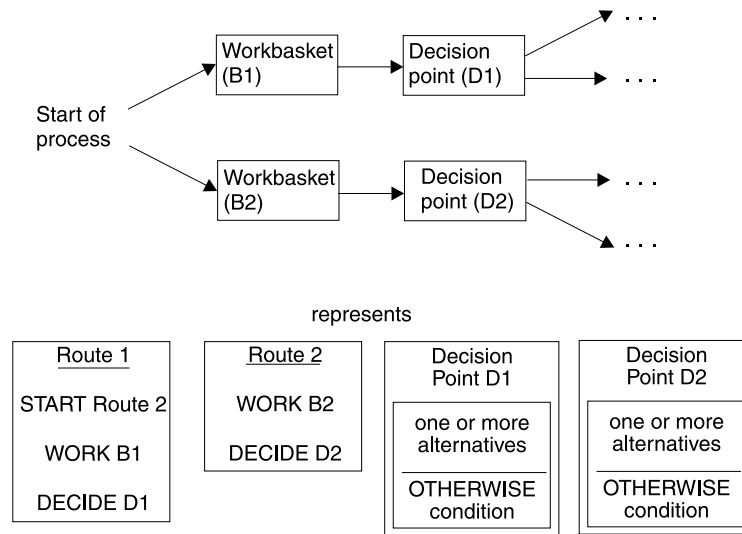


Figure 13. A Process with a Parallel Route with Multiple Decision Points and Objects

Note that the workbasket objects were not included, because they don't really add to the example by their presence. As you can see, the second instance of a work package is created by ROUTE 1 when the START ROUTE 2 step is performed. START is the only way to create additional instances of a work package. Assuming there are no customer-defined variables, there are four variables associated with this process. There is an *ACTION and *PRIORITY associated with ROUTE 1 and an *ACTION and *PRIORITY associated with ROUTE 2. In this example, we have kept the values of the variables the same, but we could have made the values different in each route.

DECIDE Command

The DECIDE command sends the work package instance to the decision point. When the DECIDE command is processed, the program looks at the alternatives defined for the decision point in this process. The system starts the route specified in the first matched alternative. If the decision point was defined with an indication to start multiple routes, the system starts the route specified for each matched alternative. If all the alternatives are checked and there is no match, the system starts the route specified as the exception route for the decision point.

In Workflow Builder, use the decision point node to define a decision point. Workflow Builder generates the appropriate DECIDE command for you.

COPY Command

The COPY route command assigns a value to a work package instance variable. When the system determines that a COPY command should be processed, the variable name and value specified in the COPY command are sent as parameters to the system. If the COPY command assigns a value to an undefined variable, the COPY command creates that variable with the desired value.

In Workflow Builder, use the assign value node to specify the value to be assigned to a variable. Workflow Builder generates the appropriate COPY command for you.

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document 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 document. The furnishing of this document 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 publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication 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 Corporation
J74/G4
555 Bailey Avenue
P.O. Box 49023
San Jose, CA 95161-9023
U.S.A.

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 Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, 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 measurements 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 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.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	iSeries
Advanced Peer-to-Peer Networking	Operating System/2
Application System/400	Operating System/400
APPN	OS/2
AS/400	OS/400
CICS	System/36

Microsoft, Windows, Windows NT and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Glossary

This glossary defines terms and abbreviations used in this book and the product document library. Refer to the *IBM Dictionary of Computing*, ZC20-1699-09, for terms or abbreviations that do not appear here.

The following cross-references are used in this glossary:

- **Contrast with.** This refers to a term that has an opposed or substantively different meaning.
- **See.** This refers the reader to multiple-word terms in which this term appears.
- **See also.** This refers the reader to terms that have a related, but not synonymous, meaning.
- **Synonym for.** This indicates that the term has the same meaning as a preferred term, which is defined in the glossary.

A

access list. A list consisting of one or more individual user IDs or user groups and the *privilege set* associated with each user ID or user group. You use access lists to control user access to items in Content Manager for iSeries. The items that can be associated with access lists are *index classes*, *workbaskets*, and *processes*.

action list. An approved list of the actions, defined by a supervisor, that a user can perform while working with items in a workbasket.

ad hoc route. A route that is not part of a defined process. An *ad hoc route* is started when a user assigns an item directly to a workbasket. The user manually routes the item from one workbasket to another by reassigning it.

administrator. The person responsible for system management, controls, and security, as well as case statistics. Synonymous with system administrator.

advanced peer-to-peer networking (APPN). Data communications support that routes data in a network between two or more APPC systems that are not directly attached.

advanced program-to-program communications (APPC). Data communications support that allows programs on an iSeries server to communicate with programs on other systems having compatible communications support. This communications support is the iSeries method of using the SNA LU session type 6.2 protocol.

annotation. An added descriptive comment or explanatory note.

APAR. Authorized Program Analysis Report.

API. Application programming interface.

application programmer. A programmer who designs programming systems and other applications for a user's system.

application program interface (API). The formally-defined programming language interface which is between an IBM system control program or a licensed program and the user of the program.

APPC. Advanced program-to-program communications.

APPN®. Advanced Peer-to-Peer Networking®.

archiving. The storage of backup files and any associated journals, usually for a given period of time.

AS/400®. Application System/400®.

attribute. Used in Content Manager for iSeries APIs, a single value associated with an item (document or folder). Each index class can have up to eight attributes.

B

binary large object (BLOB). A large stream of binary data treated as a single object.

C

cartridge. (1) A storage device that consists of magnetic tape, on supply and takeup reels, in a protective housing. (2) For optical storage, a plastic case that contains and protects optical disks, permitting insertion into an optical drive. See also *optical disk* and *cartridge storage slots*.

cartridge storage slots. An area in an optical library where cartridges are stored.

collection. The definition of storage management controls associated with a group of objects that typically have similar performance, availability, backup, and retention characteristics.

collection point. (1) The point where work packages wait for specific events to either occur or become synchronized before processing can continue. (2) A collection point is part of a work process. For example,

a collection point is where work packages that are part of the “open a new account” work process must wait until credit information is verified. See also *decision point*.

content class. A number that indicates the data format of an object, such as MO:DCA, TIFF, or ASCII.

control files. Files that govern the categories of work performed by an operator and the types of documents the system recognizes.

convenience workstation. A display workstation equipped with a printer and a scanner.

current document. A document that is being processed.

customization. The process of designing a data processing installation or network to meet the requirements of particular users.

D

DASD. Direct access storage device.

DDM. Distributed data management.

DBCS. Double-byte character set.

decision point. (1) The point where work packages continue on their current route or switch to an alternate route, depending on the specific information in each work package. Decision points are tables consisting of variable names, values, and routes. (2) A decision point is part of a work process. For example, a decision point is where work packages that are part of the “open a new account” work process receive approval or not based on credit information.

See also *collection point*.

direct access storage device (DASD). A device in which access time is effectively independent of the location of the data.

distributed data management (DDM). A feature of the System Support Program that lets an application program work on files that reside in a remote system.

display workstation. An image processing workstation used primarily for displaying documents that have been previously scanned or imported into the iSeries server.

document. (1) An item containing one or more base parts. (2) A named, structural unit of text that can be stored, retrieved, and exchanged among systems and users as a separate unit. Also referred to as an *object*. A single document can contain many different types of base parts, including text, images, and objects such as spreadsheet files.

document content architecture (DCA). An architecture that guarantees information integrity for a document being interchanged in an office system network. DCA provides the rule for specifying form and meaning of a document. It defines revisable form text (changeable) and final form text (unchangeable).

double-byte character set (DBCS). A set of characters in which each character occupies two bytes. Languages, such as Japanese, Chinese, and Korean, that contain more symbols than can be represented by 256 code points, require double-byte character sets. Entering, displaying, and printing DBCS characters requires special hardware and software support.

E

export. A process used to write data from a document in a system folder to a file. Export and import processes can be used to transfer documents among systems.

F

first in first out (FIFO). A queueing technique in which the next item to be retrieved is the item that has been in the queue for the longest time.

folder. In Content Manager for iSeries, an object that can contain other folders or documents.

folder balancing. In the iSeries, the process by which documents are distributed evenly among the available folders in the system.

folder manager. In IBM Content Manager for iSeries systems other than Content Manager for iSeries, the term used to describe the data model and a subset of the APIs. In Content Manager for iSeries, this term refers to the entire set of Content Manager for iSeries APIs.

G

Group III. A compression algorithm that conforms to a standard promulgated by the International Telegraph and Telephone Consultative Committee (CCITT).

H

HTML. Hypertext markup language.

I

image. (1) A single page of information; the result of scanning, or digitizing, a single sheet of paper. (2) An electronic representation of a picture produced by means of sensing light, sound, electron radiation, or other emanations from the picture or reflected by the

picture. An image can also be generated directly by software without reference to an existing picture. See also *page image*.

image data. Rectangular arrays of raster information that define an image. Image data is often created originally by a scanning process.

image host. The system where scanned and imported documents are permanently stored. See also *optical library subsystem*.

Image Object Content Architecture (IOCA). A structured collection of constructs used to interchange and present images.

image workstation. A programmable workstation that can perform image functions.

importing. A process by which documents are input into iSeries using files rather than the scanning process. Imported documents can be stored in Content Manager for iSeries on DASD and optical, and displayed and printed, in the same manner as scanned documents.

inbound. Pertaining to communication flowing in a direction towards the application program from external sources, such as a transmission from a terminal to the application program. Contrast with *outbound*.

index. To associate a document or folder with an index class and provide the key field values required by that class.

index class. A category for storing and retrieving objects, consisting of a named set of attributes known as *key fields*. When you create an item in Content Manager for iSeries, your application must assign an index class and supply the key field values required by that class. An index class identifies the automatic processing requirements and storage requirements for an object.

instance. An occurrence of a work package within a process. If the process consists of parallel routes, multiple instances of a work package exist.

iSeries object directory profile. A control file used in Content Manager for iSeries to identify iSeries object directories used for image document storage.

item. (1) Set of attributes and objects—one or more files containing image data, annotations, notes, or other content—that together represent a physical document, such as an insurance claim or a folder.

See also *document*. (2) The smallest unit of information that the library server administers. An item can be a folder, document, workbasket, or process. Referred to as an *object* outside of library server functions.

K

key field. An attribute of an item that represents a type of information about that item. For example, a customer data item might have key fields for the customer's name and social security number.

keyword. A name or symbol that identifies a parameter.

L

LAN. Local area network.

language profile. A control file used in Content Manager for iSeries to define country-specific parameters, such as time and date formats.

last in, first out (LIFO). A queueing technique in which the next item to be retrieved is the item most recently placed in the queue.

library server. The component of Content Manager for iSeries that contains index information for the items stored on one or more *object servers*.

LIFO (last in, first out). A queueing technique in which the next item to be retrieved is the item most recently placed in the queue.

local area network (LAN). A computer network located on a user's premises within a limited geographical area.

LU 6.2. In Systems Network Architecture (SNA), a type of session between two application programs in a distributed processing environment, using the SNA character string or a structured-field data stream; for example, an application program using CICS® communication with an iSeries application.

M

Machine-Generated Data Structure (MGDS). Data extracted from an image and put into generalized data stream (GDS) format.

magnetic storage. A storage device that uses the magnetic properties of certain materials.

magnetic tape. A tape with a magnetizable layer on which data can be stored.

magnetic tape device. A device for reading or writing data from or to magnetic tape.

MGDS. Machine-Generated Data Structure.

Mixed Object: Document Content Architecture (MO:DCA). An IBM architecture developed to allow the interchange of object data among applications within the interchange environment and among environments.

Mixed Object: Document Content Architecture-Presentation (MO:DCA-P). A subset architecture of MO:DCA that is used as an envelope to contain documents that are sent to the Content Manager for iSeries workstation for displaying or printing.

MO:DCA. Mixed Object: Document Content Architecture.

MO:DCA-P. Mixed Object: Document Content Architecture-Presentation.

MRI. Machine-readable information.

N

national language support (NLS). The modification or conversion of a United States English product to conform to the requirements of another language or country. This can include enabling or retrofitting of a product and the translation of nomenclature, MRI, or product documents.

network. An arrangement of programs and devices connected for sending and receiving information.

network table file. A text file created during installation that contains the system-specific configuration information for each node for each Content Manager for iSeries server. Each server must have a network table file that identifies it. The name of the network table is always FRNOLNT.TBL.

NLS. National language support.

O

object. (1) An item upon which actions are performed. A collection of data referred to by a single name.

The smallest unit within the system. For Content Manager for iSeries systems, this is typically a single-image document. (2) Any binary data entity stored on an object server. In the Content Manager for iSeries data model, *object* specifically refers to a document's contents or parts.

object authority. The right to use or control an object.

object directory. A control file used in Content Manager for iSeries to identify iSeries object directories used for image document storage.

object server. The component of IBM Content Manager for iSeries that physically stores the objects or information that client applications store and access.

operator. The person who handles daily system administrative tasks.

optical. Pertaining to optical storage.

optical cartridge. A storage device that consists of an optical disk in a protective housing. See also *cartridge*.

optical disk. A disk that contains digital data readable by optical techniques. Synonymous with digital optical disk.

optical drive. The mechanism used to seek, read, or write data on an optical disk. An optical drive may reside in an optical library or as a stand-alone unit.

optical libraries. Software used to store image data on optical platters. Only direct-attached optical systems contain optical libraries.

optical library subsystem. The hardware and software that provides the long-term storage of the image data. See also *image host*.

Optical Storage Support. Software that supports communication between stand-alone optical disk drives, the optical library, and Content Manager for iSeries. The software runs on the System/36™ 5363 unit serving as the optical controller.

optical system profile. A file used to define the optical controller used for the optical storage of documents.

optical systems. Hardware used to store image data on optical platters. Only direct-attach optical systems contain optical libraries.

optical volume. One side of a double-sided optical disk containing optically stored data.

OS/2®. Operating System/2®.

OS/400®. Operating System/400®.

outbound. Pertaining to a transmission from the application program to a device. Contrast with *inbound*.

override. A parameter or value that replaces a previous parameter or value.

P

page. A single physical medium; for example, an 8.5-inch by 11-inch piece of paper.

page image. The electronic representation of a single physical page. The bounds of a page image are determined by the electromechanical characteristics of

the scanning equipment, along with the image capture application specifications in the receiving data processing system.

page scan. The electromechanical process of scanning a physical page (paper) to create a bit image of the page.

pan. Progressively translating an entire display image to give the visual impression of lateral movement of the image.

PDF. Portable document format.

platter. See *optical disk*.

Presentation Text Object Content Architecture (PTOCA). An architecture developed to allow the interchange of presentation text data.

primary processor. In a group of processing units, the main processing unit and its internal storage through which all other units communicate.

printer workstation. A display workstation equipped with a printer.

priority. (1) A rank assigned to a task that determines its precedence in receiving system resources. (2) In Content Manager for iSeries workflow, the priority of the work to be performed. The priority affects the work sequencing of the work package. A larger number is a higher priority.

privilege. An authorization for a user to either access or perform certain tasks on objects stored in Content Manager for iSeries. The system administrator assigns privileges.

privilege set. In Content Manager for iSeries, collection of *privileges* for working with system components and functions. The system administrator assigns privilege sets to users (user IDs) and user groups.

process. The series of steps, events, and rules through which a work package flows. A process is a combination of the route, collection point, and decision point through which a predefined type or work package must progress.

For example, a process called "open new account" would describe the following:

- The steps that work packages related to opening a new account must follow
- The events (such as verifying credit information) that must occur before work packages for new accounts can be routed to another point in the system
- The decisions that determine whether to open a new account based on the information for that particular account (for example, a good credit rating versus a poor one).

process item. Item used as a building block in a work process.

profile. A file that governs the categories of work performed and the types of users recognized by the system.

program temporary fix (PTF). A temporary solution or bypass of a problem diagnosed by IBM as resulting from a defect in a current unaltered release of the program.

PTF. Program temporary fix.

PTOCA. Presentation Text Object Content Architecture.

R

release. To remove suspend criteria from a work package so that it is available to be worked. A suspended work package is released when the criteria have been met, or when a user with proper authority overrides the criteria and manually releases pending requests.

render. To take data that is not typically image-oriented and depict or display it as an image. In Content Manager for iSeries, you can render word-processing documents as images for display purposes.

resolution. In computer graphics, a measure of the sharpness of the image, expressed as the number of lines and columns on the display screen or the number of pels per unit of area.

rotate. A function of the document display window and the scan document display window. The orientation depends on the option selected.

route. A set of steps that move work between workbaskets, collection points, and decision points.

S

SBCS. Single-byte character set.

scanner. A device that examines a spatial pattern one part after another and generates analog or digital signals corresponding to the pattern.

scanner workstation. A display workstation equipped with a scanner.

scanning. A physical process that enters documents into an Content Manager for iSeries workstation. After a document has been scanned, it can be stored permanently.

search criteria. In Content Manager for iSeries, the text string used to represent the logical search to be performed on the library server.

secondary processor. In a group of processing units, any processing unit other than the primary unit.

server. On a local area network, a data station that provides facilities to other data stations; for example, a file server, a print server, a mail server.

side by side. A function on the document display window that displays two pages of a multipage document next to each other.

single-byte character set (SBCS). A set of characters in which each character occupies one byte.

slot. (1) A position in a device used for removable storage media. (2) A space in an optical library where an optical cartridge is stored. See *optical cartridge*.

SMS. System-managed storage.

spool file. A file that holds output data waiting to be printed or input data waiting to be processed by a program.

staging. The process of moving a stored object from an off-line or low-priority device back to an on-line or higher priority device, usually on demand of the system or on request of a user. When a user requests an object stored in permanent storage, a working copy is written to the *staging area*.

stand-alone. Pertaining to an operation that is independent of any other device, program, or system.

storage. The action of placing data into a storage device.

storage class. A storage class, in combination with an optical system identifier, defines the set of optical volumes upon which documents can be stored. Documents with the same storage class and optical system ID are stored on the same optical volume.

storage method. A means of grouping documents together for storage to an optical disk.

storage system. A generic term for storage in Content Manager for iSeries.

subsystem. A secondary or subordinate system, or the programming support part of a system that is usually capable of operating independently of or asynchronously with a controlling system.

suspend. To hold a work package at a workbasket until stated criteria have been satisfied. Work packages can be suspended for multiple criteria, therefore multiple suspend requests can exist for a work package. A document work package can be suspended

for a specific date. A folder work package can be suspended for a specific date or index class.

system administrator. The person who manages the Optical Library Subsystem and the departmental processor. The system administrator helps with problem determination and resolution. Synonymous with *administrator*.

system-managed storage (SMS). The Content Manager for iSeries approach to storage management. The system determines object placement, and automatically manages object backup, movement, space, and security.

System Support Program (SSP). A group of IBM-licensed programs that manage the running of other programs and the operation of associated devices, such as the display station and printer. The SSP also contains utility programs that perform common tasks, such as copying information from diskette to disk.

T

tape. See *magnetic tape*.

tape cartridge. See *cartridge*.

U

user. Anyone requiring the services of Content Manager for iSeries. This term generally refers to users of client applications rather than the developers of applications, who use the Content Manager for iSeries APIs.

user exit. (1) A point in an IBM-supplied program at which a user exit routine may be given control. (2) A programming service provided by an IBM software product that may be requested during the processing of an application program for the service of transferring control back to the application program upon the later occurrence of a user-specified event.

user exit routine. A routine written by a user to take control at a user exit of a program supplied by IBM.

user ID profile. A file that contains one entry for each user. The entries contain information such as processing eligibility.

V

volume. A certain portion of data, together with its data carrier, that can be handled conveniently as a unit.

W

workbasket. A container that holds work packages. Workbaskets can be used as parts of process definitions

or ad-hoc routes. In Content Manager for iSeries, a logical location within the Content Manager for iSeries system to which work packages can be assigned to wait for further processing.

A workbasket definition includes the rules that govern the presentation, status, and security of its contents.

workflow. A system that lets an enterprise define a work process and environment to automate workflow and control business processes.

work order. The sequence of work packages in a workbasket.

work package. The work that is routed from one location to another. Users access and work with work packages through workbaskets.

work process. In work management, the series of steps, events, and rules through which a work package flows. A work process is a combination of the route, collection point, and decision point through which a work package must progress.

workstation. A computer processor unit, image display unit, scanners, and printers with which the user performs input, indexing, and printing.

Index

A

- access list 7
- action
 - user-defined option 10
 - workflow information 10
- action list 2, 7
- action lists 7, 8
- actions
 - defined 9
 - for a workbasket 8
- activate 9
- add to a new folder 9
- add to folder 9
- add to workbasket 9
- advanced concepts 25

C

- change process 9
- client application 5
- client for windows 5
- collection point 3
 - terminology 14
- collection points 13
- collection points in workflow 13
- command
 - COPY 26
 - DECIDE 26
- component
 - client 5
 - server 5
- components
 - workflow 4
- continue 9
- continue function 10
- COPY command 26

D

- DECIDE command 26
- decision point 3, 21
 - alternatives 23
 - operators 23
 - routes 24
- decision point variables 22
- default action list 7
- defining workbaskets 7
- delete 9
- delete and next 9

E

- event 14
 - Customer-defined 16
 - days 16
 - item 15
 - route 14
- event list 18
- events
 - days 13

events (*continued*)

- items 13
- routes 13

export 9

H

how everything fits together 3

I

- information flow 3
- information to be processed 2
- instances 17
- item 2
- item event 15
- items
 - maximum number 8

L

listing work packages 8

M

multiple decision points 25

N

- next 9
- notes 9
- Notices 27

P

- parallel routes 25
- path 3
- planning a work process 1
- predefined paths 10
- priority 9
- process 1
 - information flow through 3
- process diagram 3

R

- reassign 9
- reassignment of work 7
- reindex 9
- remove from folder 9
- remove from process 9
- remove from workbasket 9
- route event 14
- routes 16
- routing
 - along undefined paths 10
 - with continue function 10
- routing along predefined paths 10

S

- save and next 9
- server 5

- set variable 10
- start process 10
- suspend 10

U

- understanding collection points 13
- using the continue function 10

W

- work package 2
- work package instances 17
- work packages
 - listing 8
 - presenting 8
- work process 1
 - planning 1
- work process diagram 3
- work reassignment 7
- workbasket 2, 7
 - definition 7
- workbasket actions 8
- workflow
 - understanding 1
- workflow builder 4
- workflow components 4



Program Number: 5722-VI1



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC27-1138-00

