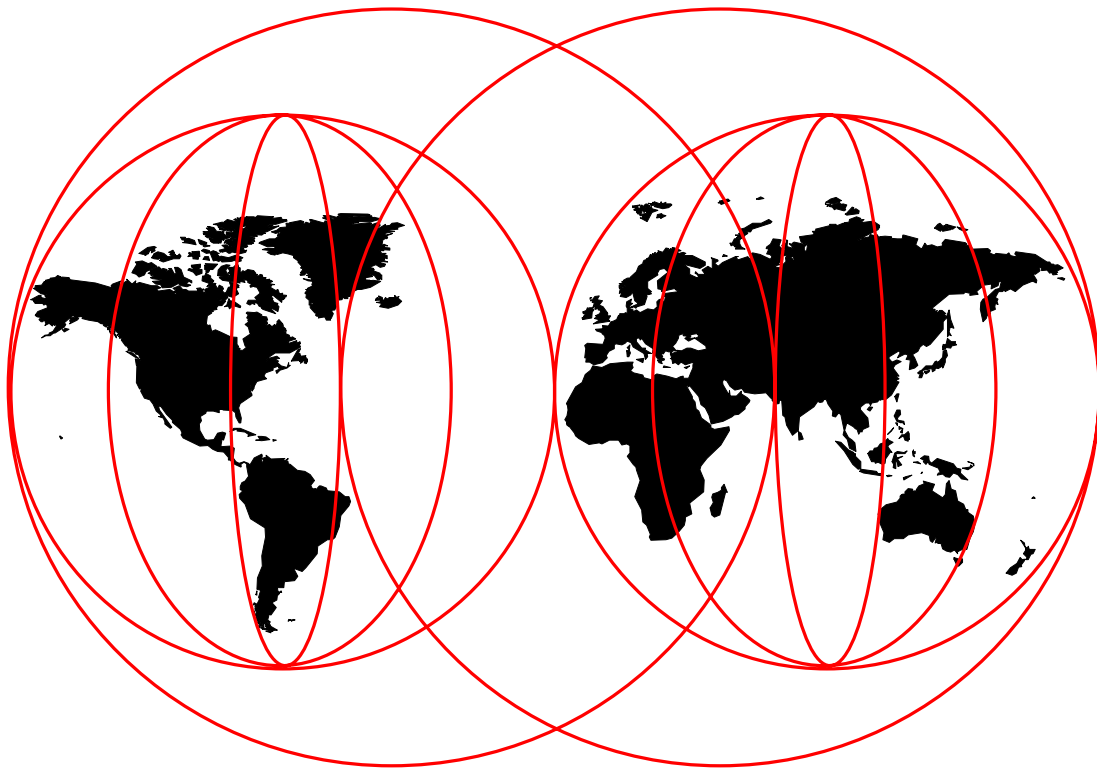


Image and Workflow Library: FlowMark V2.3 Design Guidelines

Bob Stegmaier Mike Ebbers Tomislav Begovac



International Technical Support Organization

<http://www.redbooks.ibm.com>



International Technical Support Organization

SG24-4613-02

**Image and Workflow Library:
FlowMark V2.3 Design Guidelines**

February 1998

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix C, "Special Notices" on page 29.

Third Edition (February 1998)

This edition applies to Version 2 Release 3 of IBM FlowMark, Program Number 5697-216 for use with the OS/2, Windows NT and AIX Operating Systems.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
522 South Road
Poughkeepsie, New York 12601-5400

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1996, 1998. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Preface	v
The Team That Wrote This Redbook	v
Comments Welcome	vi
Chapter 1. Introduction and Overview	1
1.1 Basic Concepts of FlowMark	1
1.2 FlowMark Buildtime: Defining Your Processes	1
Chapter 2. Importance of Process Design	3
2.1 Function, Performance, and Capacity	3
2.2 Understand the Basis: FlowMark V2.3	4
Chapter 3. Client/Server Can Mean Multiple Servers	5
3.1 Multiple Server Options with FlowMark V2.3	5
3.2 Dedicating Your FlowMark Servers	6
Chapter 4. How Big is a Process?	7
4.1 Performance and Capacity Considerations	7
Chapter 5. Starting and Deleting Process Instances	9
5.1 Starting Instances	9
5.2 Deleting Process Instances	9
5.3 Performance and Capacity Considerations	10
Chapter 6. How Big is an Activity?	11
6.1 Performance and Capacity Considerations	11
Chapter 7. How Many People Do I Assign to an Activity?	13
7.1 Performance and Capacity Considerations	14
Chapter 8. When Do I Use an Activity Block?	15
8.1 Performance and Capacity Considerations	15
Chapter 9. When Do I Use a Subprocess?	17
9.1 Performance and Capacity Considerations	17
Chapter 10. Data Container Usage	19
10.1 Performance and Capacity Considerations	19
Chapter 11. Using FlowMark Functions Wisely	21
11.1 Sign On and Sign Off	21
11.2 Shut Down	21
11.3 Filtering Work Lists and Process Lists	21
11.4 Refreshing Lists	22
11.5 Using the Monitor	22
11.6 FlowMark Bundle Activities	22
Appendix A. Factors Influencing the Size of a FlowMark Data Base	25
A.1 Activity Name and Description	25
A.2 Results of Staff Resolution	26

A.3 Data Containers	26
A.4 Other Things	26
Appendix B. The FlowMark Internet Site	27
Appendix C. Special Notices	29
Appendix D. Related Publications	31
D.1 International Technical Support Organization Publications	31
D.2 Redbooks on CD-ROMs	31
D.3 Other Publications	31
How to Get ITSO Redbooks	33
How IBM Employees Can Get ITSO Redbooks	33
How Customers Can Get ITSO Redbooks	34
IBM Redbook Order Form	35
Glossary	37
Index	49
ITSO Redbook Evaluation	51

Preface

This redbook tells you how to design your FlowMark processes to optimize performance, capacity and resource utilization. This version has been updated for IBM FlowMark V2.3 and runs on the OS/2, NT and AIX platforms.

It was written for technical professionals such as solutions architects, consultants, and application programmers who are implementing a FlowMark system. Some knowledge of FlowMark and client/server issues is assumed.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Bob Stegmaier is a Senior Marketing Support Representative at IBM's Dallas Systems Center in Roanoke, Texas, USA. Ever since FlowMark was announced, Bob has consulted with customers worldwide in all areas of FlowMark and written technical documents. He has been with IBM for 30 years, including a variety of programming assignments and 10 years with the Dallas Systems Center. He has a degree in economics from George Washington University in Washington D.C.

Tomislav Begovac is at the IBM FlowMark Competency Center in Boeblingen responsible for FlowMark technical support in EMEA (Europe, Middle East, and Africa). Before his current assignment, he was managing the development tools department in the IBM German Software Development Lab. Tomislav can be reached at beg@de.ibm.com.

This redbook was published by the International Technical Support Organization.

Mike Ebbers is a Senior International Support Representative at the International Technical Support Organization, Poughkeepsie Center. He has worked at IBM for 24 years. He produces redbooks on workflow and image products. Mike previously developed Education and Training courses on imaging and printing.

Thanks to the following people for their invaluable contributions to this project:

Joachim Schmitz
German Software Development Lab
Boeblingen, Germany

Mike Lang
IBM National Technical Support
Dallas Systems Center

For updating the redbook for FlowMark Version 2.3, thanks to:

Tomislav Begovac
German Software Development Lab
Boeblingen, Germany

Comments Welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in "ITSO Redbook Evaluation" on page 51 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Web sites:

For Internet users <http://www.redbooks.ibm.com/>

For IBM Intranet users <http://w3.itso.ibm.com/>

- Send us a note at the following address:

redbook@us.ibm.com

Chapter 1. Introduction and Overview

Workflow management helps you manage and control your business processes, pinpoint areas for improvement, and streamline your procedures for speedier cycles and shorter response times. By defining the flow of work, everyone is notified of outstanding work and presented with the required information and an appropriate application to perform the task.

1.1 Basic Concepts of FlowMark

FlowMark is a tool to help you automate and streamline your business processes. This is critical for success in today's business environment. FlowMark was written as a pure application-independent workflow engine which manages the flow of work throughout your organization. It does not have roots in forms, document or image workflow as some other products do. FlowMark helps you harness the power of other applications to handle individual tasks. You decide which applications are best suited to assist you in performing these individual tasks.

FlowMark is designed to handle the flow of work from user to user, following the business rules specified in your business process. Users have a work list (or perhaps more than one) where activities are placed automatically when the flow through a "process instance" determines there is something for them to do. The users select the next item they wish to work on from the list, and FlowMark calls the appropriate application program (defined in the process design) to perform that task. Often it is an application the users already understand. They can work in the application as long as needed to complete the task. When the program returns to FlowMark, the server is notified, and navigation continues through the process to the next activity or activities to be done. These then appear on the work lists of the people responsible for completing them.

While FlowMark has a user-oriented view, it is also able to include in the process flow both automatically started activities, that start immediately on a single user's desktop, and what are called "unattended" activities. These are batch jobs that expect no user interaction and are started automatically. Some process designs achieve the same function by assigning an activity to a specific user, specifying that it start automatically. The program is implemented so that it executes without any user interaction.

1.2 FlowMark Buildtime: Defining Your Processes

"Buildtime" is the definition facility of FlowMark. This is where you define your processes and their hierarchies, assign programs to activities, and allocate staff such as people or roles. The following definition facilities are provided:

- The Process Definition facility is where you specify and maintain process models as activity networks. These can involve multiple steps and many users. The creation and manipulation of activity networks is done with a graphical interface. An activity network can be considered to be a directed graph, where the nodes represent activities to be performed and the connectors between the nodes represent either control flow or data flow. The control flows can be unconditional or conditional. If conditional, then a Boolean expression is associated with the connector.

- The Staff Definition facility is used for the definition of staff personnel to which activities could be assigned. It maintains information about people, skill levels, roles, organizations, their relationships and their authorizations.
- The Program Registration facility is used to register programs that are invoked at process execution by an activity, and to specify input/output parameters for the programs. All programs that you wish to attach to an activity (including those used as “tools”) must be registered.
- The Data Structure Definition facility is used to define the structure of the information (in data containers) that is transported from a program to FlowMark and from FlowMark to a program. The data structures can also be used to transport data to/from activity blocks and subprocesses.
- The Server Definition facility is used to define the various FlowMark servers in an enterprise. All servers to be used for the execution of remote subprocesses must be registered.

These are the FlowMark Buildtime facilities that you will use to design your processes. In the use of these facilities, the guidelines in this redbook will come into play.

Chapter 2. Importance of Process Design

The design of your processes is critical to the project and to your business. It must be done well. You should expect to refine your processes on an ongoing basis. Since FlowMark provides data in the audit trail on process performance, you can more easily find the weak points in the processes. Since the “process flow” is now in FlowMark and not buried deep in a large application, it is easy to make the changes necessary to achieve the improvements.

2.1 Function, Performance, and Capacity

FlowMark will automate your processes, make them reliable, and see that they run quickly. If poorly designed processes caused you to move to workflow automation, keeping the same process design will merely automate the problem. It is important that you have quality processes defined before you start prototype testing with users. They need not be perfect, but quality should be evident.

On the other hand, you may already have good processes defined. But you can use FlowMark to speed the flow from user to user, call the correct application so the user does not need to remember how, pass data from user to user so it does not have to be repeatedly entered, and see that the business rules are followed exactly as defined even when the users do not remember the nuances of particular exception cases.

Process design touches everything: the way people work, the speed at which things get done, the results the customer sees (or never sees), the performance and capacity of the FlowMark server and database, the demands on the LAN, and the underlying application.

There are no hard-and-fast rules. There are always business reasons to do things in a different way. But, especially for new users, guidelines can be helpful. As you gain experience, you can modify these to fit your own business. Included here are some ideas about capacity and performance. It is never too early to begin thinking about performance. As you are designing, everything you create influences the performance of your business application.

Some of your processes may fit easily within these guidelines, while others may not. If those that fit are the most frequently used (the ones with the highest volume of instances or activities), your design should work just fine. However, if you do things that have a negative impact on performance or database capacity, even a little bit, and they occur many times per hour or per day, the cumulative impact on your system can be serious. Sometimes it is not so much what you do, but how many times you do it, that determines the real impact.

It is important to understand the workload volumes in your system. It helps to do some basic math: *the number of instances per day times the number of activities per instance times the number of users*. The result can be enlightening. Additional hardware can often overcome performance and capacity limitations. Sometimes that is the right answer.

It is impossible to teach everything there is to know about processes here, but there are some frequently asked questions to think about.

2.2 Understand the Basis: FlowMark V2.3

The suggestions in this redbook are based on FlowMark as it is implemented in Version 2 Release 3, the generally available (GA) level of code. As early users find new and intriguing ways to solve their business problems using workflow, they will at times desire functions that are not offered or find uses that exceed the capacity intended by the original design. While the following information reflects some of these limitations, enhancements are being developed in FlowMark to remove constraints and provide new functional capability. Please use this only as a guide for the use of FlowMark Version 2 Release 3.

While this document reflects the implementation in V2.3, you can expect enhancements. When new releases become available, understand what they provide and do not let this document deter your use of the added functions.

Chapter 3. Client/Server Can Mean Multiple Servers

While FlowMark is a client/server tool, do not limit your thinking and design to a single server. You should consider capacity and performance in your design, and there are times when you will need multiple servers to achieve your goals, particularly if you are designing something larger than a departmental system. You may need to ask this question: "How many servers, and of what kind, do I need to achieve my business requirements?" Do not inhibit your design by thinking: "How do I constrain my design so it works on a single machine?" In today's world of constantly improving price/performance, hardware may not be as significant an impact on the overall cost of a project as it once was. Investigate the cost implications and approach the topic with an open mind.

If you intend to start with a single server, but expect over time to add more users or more applications to your workflow platform, make sure you plan and design with the concept of multiple servers in mind. How will you divide the work between servers? Will you have a different set of users per server, with some people in each set capable of performing all activities for each process run on that server? Will you want to eventually move part of a process to a different server while keeping the "main part" on a current server, thus requiring use of a subprocess? This planning will make it much easier later when you add additional servers.

Consider the concept of a red team, a white team, and a blue team, each with its own server, each independently running the same process template, doing the same kind of work. Or perhaps you prefer different servers with different types of users doing different processes or subprocesses. Perhaps your application calls for regional servers serving local offices, running a process that under some circumstances needs to run a subprocess for headquarters approval. You could have two or more regional servers, each with its set of users doing the local activities, and a single headquarters server where that staff does the tasks in the approval subprocesses for all regions.

3.1 Multiple Server Options with FlowMark V2.3

FlowMark supports servers with OS/2, Windows NT, AIX and HP-Unix operating systems. The OS/2 and NT platforms give some range of scalability within the hardware where it runs. AIX gives you a step up in performance and capacity and a broad scale of available hardware in the RS/6000 family. You can start small and step up in platforms (even change operating systems) with little effort. This involves a change of servers, but there is no change required in your process models; they can be moved as is. Also, there is no change required on your client workstations. Both server environments support all client environments.

If a single server is not the solution to do your job, what are the options available with FlowMark V2.3?

- Simply divide the work. Have different server machines, with different databases and FlowMark servers for subsets of the users. Divide the users by teams, by organizations, by location. You will need to have a mix of users on each server machine so that all activities in the processes you want to run on that server can be completed by the users there. This could influence your process design.

The expectation in this concept is that each server and database combination are independent of any other. There is no communication between FlowMark servers. However, this solution could, over time, be combined with the following option.

- Have processes on one or more servers perform subprocesses on other servers. This function in V2.3, introduces the concept of domains. A domain is what most people consider as the “FlowMark server,” the FlowMark Runtime server and its database. In Buildtime, there is a new selection on the primary menu called Servers. From there, you define each “server” in the system as an object along with its network address and communications protocol. Then, when you create a subprocess icon in process design, you specify on which server this is to run, on the “Server” page in the notebook. That page also allows the server to be specified in a data structure member of the process activity's input (source) container. The subprocess template must be located in the database of the other server, and all staff resolution (who does which activities) is done on that other server based on the staff defined in its data base. This would follow the regional and headquarters example discussed above.

If you conclude that you need to have multiple OS/2 or NT servers, consider the other server platform option, AIX. A single fast AIX server can do the same work as several OS/2 or NT servers. It does involve a change in operating system, which may have training and skill implications, but it may be the better choice.

These capabilities may not provide you with the ideal solution that you would like to see immediately. But it is the goal of FlowMark developers to provide you with an enterprise solution. They are working to give you increasing levels of function to connect a growing number of users within the overall workflow framework. You will implement your solution today within the current capabilities of FlowMark, but you should design with the intent of eventually expanding the horizon.

3.2 Dedicating Your FlowMark Servers

It is best to have your FlowMark servers as dedicated machines. If you put other applications on the server machines, there is a good possibility that you will sometimes affect the service you give to your FlowMark users. Even “small” applications that use resources only occasionally can use significant resources during the few times they need them. These blips in usage can result in poor response times for users who are sharing that server.

If you have excess capacity on a server and need to use it for something other than FlowMark, then understand the risks and keep an eye on what these other things do. They may impact processor utilization and may also cause conflicts in memory (swapping) or disk I/O.

Chapter 4. How Big is a Process?

How big should a process be? How much of the business should it encompass? How many activities should it contain? The answer lies somewhere between bigger than the head of a pin and smaller than a galaxy. Again, there are no hard-and-fast rules, but some guidelines can help.

If the process has only one or two activities, it is probably too small. If one person does “this” then “that” and is done in one minute, then FlowMark may not add much to the work effort. But is this “process” really part of a bigger picture? Look at the front and back ends to see if these two activities are really part of a larger business process. Or perhaps there are exception conditions that result from special cases or errors that are discovered by the work done in these activities. These may show that you really have a larger process to consider.

If you expect to run the entire corporation with one massive looping process, then the process is probably too big. A process should represent the steps needed to satisfy a specific request from a customer or employee. Sometimes you may want to break a request into multiple processes.

While it is not possible to define the ideal number of activities for a process model, we can use a guideline of 5 to 20 activities. If you have more than 20, consider breaking it up by using starts of independent processes or by using subprocesses. If you have fewer than five, consider combining processes or folding in exception conditions as mentioned above.

You may begin your first implementation with fewer activities, with the plan to bring more outside activities into the process model as you gain experience in workflow. You might automate only a part of a business process, expecting to add more to it after gaining some experience. Here, a smaller process than the guidelines suggest would be acceptable.

What about exception processing? Should it be part of the process, or a separate process? Maybe. (Remember: no rules, just guidelines.) If exceptions are part of the business and do not last markedly longer than the average process, maybe they are just part of the process. But let's say you have a process to handle customer orders and shipping, and occasionally a damaged order arrives at the customer, which starts a six-month-long involvement of insurance companies, lawyers, and lawsuits. In this case it might be better to have an activity in the main process call the FlowMark API to start a separate damaged order claim process and pass appropriate information in the source container to that process. Allow the main order and shipping process to complete separately. All the needed information should be in your application database, available to the damaged order claim process.

4.1 Performance and Capacity Considerations

Creating a process instance influences both performance and database size. It involves copying parts of the process template, inserting the new instance into the database, and then doing some evaluation at the process level, staff resolution, and updating process lists for those users who have it displayed. As this is a non-trivial amount of work for the server, make sure there is payback. Very short one- or

two-activity processes (or subprocesses) need to be questioned as they will entail this overhead, which is much more than just “inline” activities. On the other hand, a very large process, with many activities and long paths that are infrequently used because of the transition conditions, can impact database size. The process instance size is influenced by the number of activities, and the instance takes this database space as long as it exists, although noticeably less in V2.3 than with earlier releases. See Appendix A, “Factors Influencing the Size of a FlowMark Data Base” on page 25, which describes some of the fields in a process template that influence the amount of data stored for each instance and thus the size of the database.

If you see the need for a one- or two-step process with both steps done by the same person, or you have a process where most of the instances are ended by a single person after a step or two, then consider the following idea. Often the business event that causes a process instance will record that information in a file or queue that is serviced by a program that creates the process instance. You could have a “constant” process, an instance for each person, which has the single step typically done. When the user selects this activity from the work list, the program does a “get next” from this queue of work and does the task. It then is designed to fail the exit condition, so that it goes back as ready on the user’s work list.

If a special condition or exception case is determined which requires further processing, then instead of setting condition data in the output container to have a larger process continue, it could use an API call to start another process that would do that additional processing. This concept can eliminate much of the overhead of creating very many short processes, yet provides the ability to handle more complex exception conditions, all using the FlowMark work list as a single place to find the things you need to do. This can also help in reducing the large number of people being shown a single item (see Chapter 7, “How Many People Do I Assign to an Activity?” on page 13).

Chapter 5. Starting and Deleting Process Instances

The most obvious way to start a FlowMark process is to open your process list, copy a process template and start the instance manually. But it is probably better to provide a simple FlowMark API program to do that. This program could be accessible to users via an icon on their desktop, or could be called by one of your application programs that determines the need for a process to start as the result of some business event.

5.1 Starting Instances

If you have an application that collects many of these requests, be careful when you decide to start many process instances. Starting 200 instances might be fine in the middle of the night when there are no users on the system. But a program that does this during normal working hours will likely take over the server, much to the detriment of user response time. If you do have the need to process accumulated queues of requests during working hours, consider programming a delay in the creation loop to give your users a chance to get their work done.

Investigate the `ExmcStartProcess` API call. While doing that, consider providing the instance name yourself. You will find that having instance-specific information in the process instance name makes using your process list, the monitor and the audit trail much easier. Customer name, invoice number, account number and other information might be made part of the instance name.

5.2 Deleting Process Instances

A process instance is called “finished” when the workflow has progressed through all defined activities to the end of the process. A process instance is called “terminated” if the process was forced to finish by an authorized user or by the API call `ExmTerminateProcess`.

Finished or terminated process instances are automatically removed from the work lists and deleted from the database if the user who creates the process instance has checked the checkbox “Delete finished items” in the personal data settings notebook. It lets you define the following for each user:

- Activities that have the status finished, disabled, or force-finished are automatically deleted from the user's work list.
- Process instances created by the user with the status finished or terminated are automatically deleted from the database when they are finished or terminated.

Whenever an instance is deleted, all activities for the process instance are deleted from the work lists, as are all subprocesses for the process instance.

If the “Delete finished items” checkbox is not checked, you have several other options for deleting process instances:

- Delete them manually from the process list.
- Have them automatically deleted by the server. after a specified interval during a nightly database processing run

- Write a program that calls the ExmDeleteProcess API. You can use this API to explicitly delete individual instances whenever you wish, based on such things as time and date or other criteria external to FlowMark processing.

The finishing and subsequent deletion of process instances is at least as resource intensive as creating an instance.

When finishing/terminating and then deleting a process instance, FlowMark internally has to:

- Search for the process
- Update all work lists that an instance has been terminated (even if no clients are running) to delete finished activities of the instance
- Update all process lists that an instance has been deleted
- Delete all data relating to the instance from the database
- Perform all these tasks for any existing subprocesses

You might therefore want to off-load much of this activity to hours when there is less system activity and you are not likely to impact the users.

5.2.1.1 Deleting Process Instances and Activities Offshift

By not checking the “Delete finished items” checkbox, you off-load the deletion to a delete interval defined by the “Days cleanup interval” field on the FlowMark Start Servers window. The cleanup interval is defined in days. Cleanup is done by the FlowMark notification servicer, which starts deleting instances and activities at midnight. Deletion of all finished or terminated instances and activities will then start when the cleanup interval is passed. Since this happens at midnight, you need to have the server running at night for this to occur.

5.3 Performance and Capacity Considerations

The basic idea to understand here is that instance creation, and to some extent instance deletion, are resource intensive. There is a fair amount of work involved for the server, and there are significant updates (inserting or deleting data) to the database. It is important to avoid doing much of either during normal working hours. It is the same load whether you do it by a program that loops or by selecting many from a process list and specifying delete. If you do significant numbers at once it will impact your users. If it is possible, defer these efforts until off hours.

On the other hand, leaving completed instances in your database will take up database space, and can have some impact on performance. So you need to make some trade-offs between daytime performance and database capacity.

To summarize this point, you should remove instances from the database as soon as practical after they have completed. Do it at times where the resource usage will not impact your users, especially on busy servers.

Chapter 6. How Big is an Activity?

How much work should an activity represent? How long should a user take to complete an activity? For activities that involve the user interacting with a program, thinking about the problem, and then responding correctly, consider these guidelines:

- An activity is done by one person (if you want another person involved, that is a reason to create the next activity).
- An activity is done at one sitting (usually there is no long time spent doing other things in the middle of the activity).
- An activity should take 5 to 10 minutes or more on average to complete (a user could be expected to do 5 to 20 an hour and keep that pace all day).

But there are some exceptions. You may choose to keep your process at a higher level (in application development, write the “class,” unit test the program, and other things that may take days). A manager’s sign-off only takes a minute, but must be done. Exceptions are not necessarily bad. But when you design one, ensure that you have a good reason.

6.1 Performance and Capacity Considerations

Typically, FlowMark is used to automate processes of “knowledge workers.” These are people who are paid good salaries to think. If your design expects a user to do two activities a minute and keep it up hour after hour, there is not much thinking going on; that is more of a mindless task. How long does it take to read a screen or two, analyze the information, make good decisions, and enter some data or a response? Probably more than 30 seconds.

Consider the potential overhead as you design. When a user selects an activity from the work list, there must be communication between client and server, the specified application program must be called, probably loaded from disk, be initialized, and then communicate with the user. There are applications that take 10 seconds (or much more) when you initiate them from an icon or the command line. This is a lot of overhead that will impact workers if you have designed what you believe is a 30-second activity.

Another possible impact of very small activities, particularly for cases where you intend users to do significant volumes of them, is the affect it has on the user’s work list (it can get very large), and on the network traffic, constantly adding, updating, and deleting things from the work lists. We will say more about the size of user work lists in subsequent sections.

There is some amount of work required from FlowMark when you go from activity to activity in your process. The completion of one activity is recorded at the server, which performs navigation evaluation to determine what is next, executes staff resolution (who is to do it), and sends the activity item to those who should receive it. While perhaps not a major factor, try not to let the overhead of doing something become a significant part of the work done. Use the tool only where it adds significant value.

Here is an example. A new customer comes to your retail business. You would like to keep the customer for a long time so, as part of the “new customer process,” you gather information into a database. The process design has the following activity steps:

1. Enter customer and spouse names
2. Enter social security numbers
3. Enter addresses
4. Enter home/work phone numbers, etc.

This is too detailed. It looks more like a program design than process design. There would more likely be a single activity to "create a customer record." The application program for the activity might do those things, with different windows, but there is no need for that level of detail in the process design.

If you have a process that is not heavily used, perhaps one for some unique circumstances, the size and number of activities has less impact. The real impact on performance comes when you multiply an aspect by 1000 or even 5000 times for the number of activities, or when the process template runs every day.

Exceptions to this are activities that are started automatically and do not require any user interaction (in FlowMark terms, autostarted and unattended). These will run at machine speed. But a high number of these automatically started activities can put a lot of stress on your FlowMark server. Be careful if you find that a significant number of the activities in your process have no real user interaction. Even significant amounts of work are done quickly at dedicated machine speeds. A significant number of such activities can cause workloads equivalent to a great many users working at the 5 to 10 minutes per activity guideline, and will impact system performance and user response time.

Perhaps another example would help make the point. If a typical user on your FlowMark system does 100 activities a day (for example, evaluate profitability of a business proposal, enter information for a new customer, etc.), that would be about one activity every 5 minutes. It might be possible for a “fast computer” to complete an activity every 3 seconds. In that case the computer playing the part of a “single user” could put the same resource demands on the FlowMark server as 100 of those “5 minutes per activity” users. As always, automatic activities are not inherently bad; they are useful, even necessary. But understand the implications as you design your processes, and plan for the necessary capacity in your FlowMark server(s).

Chapter 7. How Many People Do I Assign to an Activity?

There are many options in FlowMark to help assign activities to different people. These are quite helpful in getting the job done. But do not go overboard, giving everything to everyone. Think about the implications. Narrow the range of who is assigned activities as quickly as possible in the flow of your process.

One reason to assign an activity to multiple people is to enable them to do their own load balancing, all picking from a single list of work.

But there are negative aspects if you “resolve” activities to too many people. There is more work for FlowMark:

- It must find each of these people and update their work lists in the database.
- Then it must send messages over your network to update each person's work list (when the activity becomes “ready,” then when it begins “running,” and again when it is “finished”).

Besides, a constant stream of items to a work list can be an annoyance to the users, because:

- Their work list will “roll,” since they share it with many others. Activities will constantly be added, changed, and deleted, as their work list reflects the tasks of others.
- There is an increased probability of conflicts. If all users have their work list sorted in the same order with the same items, all will have the same activity at the top. When one selects that top activity, someone else may also be selecting it at the same time. Only one will get it; the others are told that someone else is doing it.

Of course, you want to design your workflow system to keep the users happy, so they can get more work done. Here are some ideas:

- Start with a guideline of three to five people assigned to an average activity (whether by role, skill level, or organization).
- Limit the number of persons who receive any single activity to 10 people, or 20 at most. Consider three to five people as the average to shoot for.
- Although early in a process flow there may be a need to put things on more people's work lists, try to narrow the number as the flow progresses. The number may ebb and flow at times, but try to keep it under control. There are many options on the staff pages in the activity notebook that can help you.
- Spend some time considering how best to use the different capabilities of role, organization, and skill level. How will you use them to divide and differentiate the total population of users? Intelligent use of these staff attributes is what will enable you to assign individual activities to reasonable numbers of just the right people to do a specific job.
- Consider assigning an activity to “whoever” did an earlier activity. In effect, when a person selects an activity from a shared role, and the same role has later activities in the process flow, then have that person own the rest of the activities for that role in that specific process. This may be desirable since this person is familiar with the details of this specific instance.

- If there is some front end program that uses FlowMark API calls to create and start process instances, have it divide the work in a round robin way: one for team A, then team B, then team C, then back to A. You can use the system fields in the data container to limit the “range” by department, or have separate roles whose members have the same job. For instance, instead of a role “Clerk” for billing, you may have roles CLERKA, CLERKB, and CLERKC. Again, you can use the data container (or use a separate but “the same” process template).

7.1 Performance and Capacity Considerations

For every user who is assigned an activity, there is a commitment of work efforts for FlowMark, including the actual assignment, updating the database, notifying the user of the activity and when it changes state, and database space to maintain the extra data. Each of these work efforts is small. However, when multiplied by many users for many activities in many instances, the impact is significant. Weigh that against the benefit you expect from assigning an activity to multiple users. Finally, consider that only one user will really process the work item. Each person assigned beyond that one adds overhead to the operation of your system.

Chapter 8. When Do I Use an Activity Block?

An activity block is a construct that allows you to group several activities together. Its major functions are:

- To reduce clutter at a higher level. This lets you have a cleaner big picture at upper levels of your process.
- To allow a loop. The activity block can have an exit condition, so the entire block is repeated until particular conditions are met. This gives you a “do until” construct for a group of activities.

Consider using a block when you have the above needs and when you have at least three activities to include in it. You can make a single activity loop by specifying an exit condition for the activity itself. If an activity exit condition fails, that activity is returned (in a ready condition) to the work list of the user who performed it.

8.1 Performance and Capacity Considerations

There is not much overhead in using an activity block. The cost incurred is for evaluation at the beginning and end of the block itself. It differs from a subprocess in potential database space usage. The block is always included in the instance at the time it is created. The next section explains how a subprocess differs in this aspect.

Chapter 9. When Do I Use a Subprocess?

A subprocess is really just a process, but it is called by another (parent) process. It has functions similar to an activity block, but it can do more for you. The characteristics of a subprocess include:

- Reusability: a subprocess can be invoked at multiple points in a single process; and it can be invoked from different processes. It is a reusable object.
- It can run as a stand-alone process.
- It can run on a separate server from its parent process, using a different FlowMark database and different users (staff definitions). This permits you to shift work to another server.
- It can be tested alone, allowing independent application development.

Like an activity block, a subprocess:

- Can have an exit condition, so it can repeat until specified conditions are met.
- Helps make high-level process diagrams easier to understand.

In addition to the many functions that subprocesses can serve, they can be used to save some amount of database space, in the case where they are infrequently used in your process flow. With such a list, there is the temptation to use them everywhere. But you need to be careful not to overuse them, as you can significantly alter the performance of your system.

9.1 Performance and Capacity Considerations

A subprocess affects performance and capacity. You are trading off between processor usage and database space.

A subprocess is “late bound,” that is, not invoked until needed. There is no database space consumed unless the subprocess is invoked. Thus, if you have a process that includes multiple special cases or exceptions or has several possible paths, the use of subprocesses will reduce the amount of database storage used for a process instance. Rather than having many different activity steps loaded when an instance is created, you can defer them so only the particular special case (subprocess) is loaded, and then only if you need one. Depending on your business process, this could save some amount of database space plus the overhead to create large instances.

On the other hand, when you invoke a subprocess, the effort involved is similar to creating a process instance. There are performance implications for creation and deletion of the instance. See Chapter 5, “Starting and Deleting Process Instances” on page 9 for a discussion of this.

It is worth considering the ability to run a subprocess on another FlowMark server even if your original plans are for a single server. If you later add other servers, you can simply update the server page in the subprocess activity notebook and change the server (domain) where the subprocess runs.

So you need to make a decision. As in most performance and capacity decisions, there are trade-offs. In this one, you are weighing processor usage on the server versus space in your database. Ask yourself how frequently the particular path is followed. If the probability is high that the function is used every time, it is better to

leave the activities inline. If there is a low probability of needing the function, a subprocess is preferable. Also, with subprocesses there are also the considerations of ease of use and having a reusable object. You must make the trade-offs.

Chapter 10. Data Container Usage

The FlowMark data container is used to pass information from activity to activity within the process. It also controls the flow within the process when data fields are used in transition conditions. The terminology used is “process-relevant data.” It is important to understand this concept. The data container is not intended to be a database. The data you specify for the data container should be specifically needed for:

- Navigation, referenced in transition or exit conditions.
- Pointers into your enterprise databases where the application information really exists (for example, customer number, account number, invoice number, image folder, and document name).
- Variable information you would like to have appear in activity descriptions. This helps users to quickly find particular activities, and can assist in limiting work list size by Include (filtering) and Find functions.

While the data container is very useful, you need to be careful not to abuse the concept.

All data container information is kept in the process instance in the database. It stays there for the life of that instance. Every input and output data container created along the way exists in the database until that instance is deleted. Consider a somewhat extreme example. If you specified 10,000 bytes of data container, then pass it to/from 20 activities, it appears 20 times in your FlowMark database. It takes many times its original space since it is copied over and over, remaining there from the time the instance is created until it is deleted. When you multiply by the number of instances you may have in your database at peak times, you could encounter database space problems.

Another aspect is the impact this might have on runtime client response. There is overhead in accessing data items using the container APIs. If you have many items, most of which are not really used in an activity but merely passed on downstream in the process, there may be unneeded delays at the client.

10.1 Performance and Capacity Considerations

Here are some data container guidelines:

- Keep only key data in the data container. Keep application data in the application databases and let the applications access it from there. Pass the application the key needed to find it.
- Do not take the approach of just passing all data to every activity in the process. Have different structures that represent the specific data needed at different points in the process.
- Map data elements to the specific point where they are needed in the process.
- Do not use the FlowMark data containers as an application database or a data file server.
- Use the newest container APIs for the client.
- While you should use descriptive names for data fields, do not make them extremely long.

Chapter 11. Using FlowMark Functions Wisely

There are many functions in FlowMark that can influence system performance but are not directly related to process design. They relate to end-user activities. The people who design a FlowMark system, those with the most knowledge of FlowMark, frequently have the opportunity to influence the training the users get. Therefore, here are some guidelines to help your users.

11.1 Sign On and Sign Off

It is best to sign on to FlowMark once a day. A signon can potentially result in the loading of a lot of information from the server to your workstation. If you repeatedly sign off and on, the information that is downloaded is often much the same as what your workstation had previously. Bringing that information down again and rebuilding your work lists and process lists creates additional system load, additional network traffic, and delays on your workstation. When protecting a user's worklist is a concern, consider use of a software "lockup" function to lock the workstation when not attended.

11.2 Shut Down

Train users to close some off their windows before closing FlowMark. When a user signs on, FlowMark tries to restore the environment to what it was when FlowMark was shut down. If you leave FlowMark by just closing the Runtime - Icons window, leaving not only your work list but possibly other work lists and your process list open, that is what FlowMark will open when you next sign on. Over the course of a day you may open many of these, but it is unlikely you would like to wait for the server to provide all that information and the client to build all those icons and lists first thing in the morning. It would be better to train the users to close the various windows, except perhaps their primary work list, before closing the FlowMark Runtime window. It will make for less system load and a quicker startup in the morning.

11.3 Filtering Work Lists and Process Lists

In FlowMark V2.2, a new concept, "filtering at the server," was introduced. You can specify to the server which items in your work list or process list you wish to see. Only those items are loaded from the server to your client workstation. Appropriate use of this technique can significantly reduce both server resources for signon and list refresh and client response time as well.

To use this filtering, it is important that you include it in the process design. Filtering implies that there are things the users can specify that will allow them to differentiate between what they want to see and what they would rather not. For processes, the process name, process description, and category are important. For work lists, the activity name, activity description, and process name can be used. If you understand how your users want to classify their work, you can provide a better design (naming conventions and inclusion of variables) for these fields, which will help the users make use of filtering.

To see the exact possibilities, or to help train your users, do the following, starting from the Runtime client icons:

- For work lists, select the work list icon, then the specific work list. Open settings, and go to the Activities page. You can do this for each work list if you have multiples.
- For process lists, select the User information icon, then select the database, then in the Personal data notebook, go to the Processes page.

Chapter 7, “How Many People Do I Assign to an Activity?” on page 13 talks about long work lists. A frequent cause of long work lists is assigning each activity to many users. We suggest that you limit how many things are assigned to each user, where practical.

Another idea to treat carefully is “shared work lists.” Some have created artificial users who play many or all roles in their system. This provides an overall view of the work in the system. But it can also result in very large work lists, and anyone accessing such a list will feel the impact of the load on the system.

Whatever the cause of long lists, you can help limit the impact. Implementing this is a combination of process design and user training. If you design it but the users do not use it, nothing is gained. If you do not design for it, the users will find nothing to use. But it can have a significant positive impact, especially when there are long work lists and process lists.

11.4 Refreshing Lists

“Refreshing” is a request to erase the copy of a list on the user's workstation and bring a fresh copy from the server, rebuilding the list for the user. There are times when this needs to be done, but many people doing it regularly can impact system performance. This applies to process lists and work lists.

11.5 Using the Monitor

The FlowMark monitor function is extremely valuable in determining what is happening in a specific process instance. But, again, a tool can be misused. Unless you have a very good reason, do not constantly open or refresh monitors because refreshing a process monitor (or sometimes having several displayed and refreshing all) can put unnecessary overhead on your server. If many users repeatedly do this, the effect is multiplied. Teach your users the great power it has, but teach them to use it wisely,

11.6 FlowMark Bundle Activities

An advanced FlowMark function called “bundles” is described in Appendix A of *IBM FlowMark: Modeling Workflow, SH19-8241*. This function allows you to dynamically create one to n parallel activities (or activity blocks or subprocess activities). If you use them, here are a few things to remember. The effect of the bundle planning tool is a real-time modification of the process instance. It will involve some amount of updating of the database. Since the resulting modification may result in 10 or 20 parallel activities, the effect is the same (for doing staff resolution and updating work lists) as if 10 or 20 users all completed activities at the same time. This can be a peak in demand on the server. Also, bundles imply

arrays of data container items. The effects of large data containers was discussed in Chapter 10, “Data Container Usage” on page 19.

Appendix A. Factors Influencing the Size of a FlowMark Data Base

When you design your process in FlowMark Buildtime, you create a process model. You then translate this model and create a process template. This is a bit like compiling a program. The template contains the rules and other information necessary to run an instance of the process. You will likely have many instances of each process template active in FlowMark at any point in time. While all instances that were started with a particular template will use that one single template (it is not duplicated), there is a certain amount of information in the database for each unique instance. Some is there because it can be altered by the user, and some is there based on things happening dynamically as each instance follows its particular path through the process template.

Since these things will influence the size of your database, it may help to understand what some of them are. You will see that many have already been mentioned throughout this redbook.

A.1 Activity Name and Description

While you may think of these two items as fixed, since you specify them when you define the process, they are actually dynamic (or can be), so copies are kept for each instance.

If you provide a description of the activity on the general page of the activity notebook, you can include in it the contents of variables from the input data container. This is often done because it helps convey information from previous activities directly to users' work lists, helping them understand more about the activity or instance, and allowing them to better choose which activity to work on.

Users can alter both the activity name and the description. They do this from their work list, selecting the activity and its settings and then going to the General page of the notebook. They might do this to make note of something that kept them from completing the activity or to pass information on to others who have that activity on their work list as well.

The point is that for each activity in each instance, copies of the activity name and description are kept in the database. The bigger you make these (and description can be up to 1024 characters), the more space will be taken in the data base. Remember the guideline that it is important to understand how many times something occurs. To take an extreme example, if you filled all 1024 characters in the description of every one of 50 activities in a process and then had 2000 instances in the database, how much space is that?

$$1024 * 50 * 2000 = 100,000 \text{ KB}$$

And that does not count what FlowMark and the database would require just to keep track of it. As always, use the function when it helps you, but do not misuse it. Constant information is better put on the Documentation page of the activity notebook. Only one copy is kept of that, in the translated template, no matter how many instances.

A.2 Results of Staff Resolution

When FlowMark determines that a particular activity should be run, it goes through a function called staff resolution. This function determines who should have this activity as "ready" on their work list. Based on what the process designer has specified on the two "staff" pages in the activity notebook, such as role, organization, and skill level, FlowMark determines all the people who should see this, includes that in the dynamic instance information, and adds it to the database copy of the users' work list. As you can see, the more people an activity is assigned to and the more items a person has on his work list, the more space is required for the database.

A.3 Data Containers

We discussed this in Chapter 10, "Data Container Usage" on page 19. As FlowMark navigates through the process, your application programs add data to output data containers. This data is then copied into one or more input containers, as defined in the data mapping of your process model. The number of fields and the size of the data provided for string fields, as well as the number of input and output containers, will help determine the necessary space in the database for each instance. Since the container creation is dynamic, the containers exist and are filled only when the process flows down the path where they are defined. For paths not taken in a particular instance (called dead paths), no containers are created.

A.4 Other Things

The items mentioned above are the major factors in determining the size of an instance. But there are others. As mentioned earlier, the use of a subprocess results in another process instance being created when it is encountered in the process flow. Bundle activities result in real-time updates (additions) to the database. And the notification server, if used, can add to the database contents.

Appendix B. The FlowMark Internet Site

If you would like more information on FlowMark, visit the Internet site:
<http://www.software.ibm.com/ad/flowmark>

Here you will find lots of information on what is happening in the world of FlowMark, frequently updated.

Appendix C. Special Notices

This publication was written to give system architects more information to plan for the number of servers needed for their FlowMark system, and to design it for better performance. The information in this publication is not intended as the specification of any programming interfaces that are provided by FlowMark. See the PUBLICATIONS section of the IBM Programming Announcement for IBM FlowMark for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter 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 the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

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, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

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

AIX	AIX/6000
DB2	FlowMark
IBM	Operating System/2
OS/2	

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

Pentium, MMX, ProShare, LANDesk, and ActionMedia are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

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

Appendix D. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

D.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see "How to Get ITSO Redbooks" on page 33.

- *FlowMark Installation and Administration*, SG24-4614
- *FlowMark and VisualInfo with Windows*, SG24-4712
- *Integrating VisualAge Generator and FlowMark*, SG24-4239
- *Integrating IBM FlowMark with Lotus Notes*, SG24-4851
- *Sample Applications for FlowMark*, SG24-2184

D.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
Lotus Redbooks Collection	SBOF-6899	SK2T-8039
Tivoli Redbooks Collection	SBOF-6898	SK2T-8044
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
RS/6000 Redbooks Collection (PDF Format)	SBOF-8700	SK2T-8043
Application Development Redbooks Collection	SBOF-7290	SK2T-8037

D.3 Other Publications

These publications are also relevant as further information sources:

- *IBM FlowMark: Modeling Workflow*, SH19-8241
- *IBM FlowMark: Managing Your Workflow*, SH19-8243
- *IBM FlowMark: Programming Guide*, SH19-8240
- *IBM FlowMark: Installation and Maintenance*, SH12-6260
- *IBM FlowMark: Application Integration Guide*, SH12-6267
- *IBM FlowMark: Diagnosis Guide*, SH19-8239

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at <http://www.redbooks.ibm.com/>.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet** - type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

To get LIST3820s of redbooks, type one of the following commands:

```
TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
```

To get BookManager BOOKs of redbooks, type the following command:

```
TOOLCAT REDBOOKS
```

To get lists of redbooks, type the following command:

```
TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
```

To register for information on workshops, residencies, and redbooks, type the following command:

```
TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1998
```

For a list of product area specialists in the ITSO: type the following command:

```
TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
```

- **Redbooks Web Site on the World Wide Web**
<http://w3.itso.ibm.com/redbooks/>
- **IBM Direct Publications Catalog on the World Wide Web**
<http://www.elink.ibm.link.ibm.com/pb1/pb1>

IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to announce@webster.ibm.link.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (<http://www.redbooks.ibm.com/redpieces.html>). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** — send orders to:

In United States:	IBMMAIL usib6fpl at ibmmail	Internet usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications Publications Customer Support P.O. Box 29570 Raleigh, NC 27626-0570 USA	IBM Publications 144-4th Avenue, S.W. Calgary, Alberta T2P 3N5 Canada	IBM Direct Services Sortemosevej 21 DK-3450 Allerød Denmark
--	--	--

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States) or (+1)001-408-256-5422 (Outside USA)** — ask for:

Index # 4421 Abstracts of new redbooks
Index # 4422 IBM redbooks
Index # 4420 Redbooks for last six months

- **Direct Services** - send note to softwareshop@vnet.ibm.com

- **On the World Wide Web**

Redbooks Web Site	http://www.redbooks.ibm.com/
IBM Direct Publications Catalog	http://www.elink.ibm.link.ibm.com/pbl/pbl

- **Internet Listserver**

With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to announce@webster.ibm.link.ibm.com with the keyword `subscribe` in the body of the note (leave the subject line blank).

Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (<http://www.redbooks.ibm.com/redpieces.html>). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

Glossary

Note: This glossary defines terms and abbreviations for IBM FlowMark. For more information about the differences and other terms, not defined here, refer to the respective publication as listed in Appendix D, "Related Publications" on page 31.

A

activity. A unit of work that is performed by one person in one place and at one time. An activity consumes time and resources and has a defined duration with an explicit start and end time. In LOVEM-ProModeler: on a PLOVC or JLOVC, an activity is a manual process path component, which receives input from an upstream component, acts upon it, and sends output to a downstream component. Activities can be added to manual bands. In FlowMark: an activity is one of the steps that make up a process. See also *program activity* and *process activity*.

activity block. In FlowMark: a modeling construct that enables the grouping of related activities into a lower-level diagram. It also enables the modeling of loops and bundles. See also *activity bundle*.

activity bundle. In FlowMark: a type of activity block that supports multiple instances of a single program or process activity during run-time. The number of instances of an activity is determined during run-time by a special program activity called the planning activity. See also *planning activity*.

AIF. Application Integration Feature: a CICS/ESA product which helps you integrate your existing applications without writing any code. Applications communicate across platforms using MQSeries for MVS/ESA, which keeps them insulated from network complexities.

AIR. See *assumptions/issues/recommendations*.

AIX. Advanced Interactive Executive (UNIX)

ALOVC. See *architecture line of visibility chart*.

animation. A facility for dynamically verifying workflow models. Animating a workflow model lets the user simulate the flow of work through its activities.

API. See *application program interface*.

APPC. Advanced program-to-program communication. A commonly used protocol for various network

environments, such as Internet, Host communications, and LANs. FlowMark uses either APPC or TCP/IP. See also *TCP/IP*.

application development models. Process and data models required for application systems development.

application program interface (API). An interface provided by the workflow manager that enables programs to be started, processes to be controlled, and operators to work with data containers.

architecture line of visibility chart (ALOVC). The graphical representation of the essential customer and enterprise processes and key data needs and their main characteristics arranged in sequence. See also *enterprise architecture*.

As Is. The current state of a process or process path. See also *To Be*.

As Is view. A chart or diagram showing the current state of a process or process path. See also *To Be view*.

assumptions/issues/recommendations (AIR). In LOVEM-ProModeler: a business control parameter that can contain assumptions, issues, and recommendations.

attribute. The characteristic or property that defines an entity, such as the attribute of a unit of information. See also *representative attribute*.

audit trail. A facility for recording events that occur when process instances are run.

automation band. The horizontal section on a PLOVC or JLOVC that contains systems, system functions, or computer data stores.

Automation Manager. The Automation Manager is responsible for connection of remote clients to OLE servers.

B

band. The horizontal sections of an LOVC. Examples of bands:

- Customer band
- Internal/external organization band
- Internal business area or department and job bands
- Manual/automation band
- Automation band

bar code. Industry standard pattern of vertical lines. You can use bar codes to indicate the beginning of a new folder, the beginning of a new document, or to provide a value to be used in indexing the folder or the document.

base product. The product that provides the functionality required for the operation, for example, FlowMark, Lotus Notes. This is the product called via the Service Broker Manager.

benchmarking. Comparing something with a standard, like comparing the performance of a business process with another process of the same kind.

best of breed (BOB). A company that offers the same or similar products and services as other companies in that category, but at higher levels of performance in one or more area of competition (for example, price, quality, competence, customer service, and so on).

block. See *activity block*.

blueprint. The exact graphical representation of As Is business processes or To Be designs. A blueprint can be created at the architecture, logical, physical, or job level. It can be used for implementing new processes and for ongoing process management.

blueprinting. The procedure used to document a company's process design in graphical form using ALOVCs, LLOVCs, PLOVCs, or JLOVCs.

BOB. See *best of breed*.

bottom-up. Starting the modeling or design of business processes from their lowest level of abstraction and detail and then integrating lower-level models or designs into a higher-level whole. See also *top-down*.

BPM. See *business process management* and *business process modeler*.

BPR. See *business process re-engineering*.

Buildtime. In FlowMark: the component used to define processes. See also *Runtime*.

bundle. In FlowMark: a type of block that supports multiple instances of a single program or process activity at run time. The number of instances of the activity is determined at run time by a special program activity called the planning activity. See also *bundle activity*, *pattern activity*, and *planning activity*.

bundle activity. In FlowMark: one of the multiple instances of the pattern activity created for an activity bundle during Runtime. The number of instances is determined by the input to the planning activity. See also *planning activity* and *pattern activity*.

bus. A term borrowed from electrical engineering (or computer design), which signifies a continuum with concrete contact (start and exit) points. In this manual, it represents a continuum of repetitive and unpredictable processes, a set of sequential data stores, or a continuum for capturing critical process quality measurement points.

business. An entity that engages in commerce. A business produces or sells goods and services, has goals, processes, and personnel.

business area. Part of an enterprise implementing a group or processes that support one aspect of furthering the mission of the enterprise. Business area is part of the logical model (LLOVC).

business control parameter. Goals, predictions, targets, observations, and measurements of the enterprise or individual organization units, such as CSFs, AIRs, and CMPs. These business objects are of primary importance to the purpose of the enterprise and can be assigned to processes and activities in logical and physical models and blueprints.

business process. See *process*.

business process management (BPM). The ongoing management of business processes, from day-to-day operational process management to radical business process re-engineering.

business process modeler (BPM). An IBM business modeling tool that is based on IBM LOVEM. Short name: ProModeler.

business process re-engineering (BPR). A disciplined approach to radically changing business processes.

C

CABE. Computer Aided Business Engineering.

call flow management. The automated management of telephone calls; especially applicable for call center applications, where phone calls are treated as units of work and are tracked and measured.

cardinality. In data modeling: an attribute of a relationship that describes the membership quantity. There are four types of cardinality:

1. One-to-one
2. One-to-many
3. Many-to-many
4. Many-to-one

CASE. Computer Aided Software Engineering.

change management bus. On the ALOVC, a continuum of repetitive and unpredictable processes for enabling customers to request and affect changes (for example, a proposal, a contract, or an order at any time during the relationship).

child organization. In FlowMark: an organization within the hierarchy of administrative units of an enterprise that has a parent organization. Each child organization can have one parent organization and several child organizations. The parent is one level above in the hierarchy. See also *parent organization*.

CICS/ESA. IBM Customer Information Control System/Enterprise Systems Architecture.

common specification language. A means of communicating across various functions, organizational units, or levels of management in a precise language using graphical representations, such as the four types of LOVCs.

computer data store. A business object representing computer files, databases, or other media that store information.

condition. In FlowMark: an expression that determines the flow of control through a process instance. See also *start condition*, *exit condition*, and *transition condition*.

connector. An arrow drawn between two nodes in a process diagram to signify the flow of control or data. See also *data flow*, *information flow*, *material flow*, *control flow*, *control connector*, *data connector*, and *default connector*.

constrained. Pertaining to, or characteristic of, the PLOVC and JLOVC. Representative of the factors that define *how* a process path or job is performed and by whom (for example, time, money, organization, and technology are constraining factors).

container. See *data container*.

control connector. In FlowMark: the graphical representation of the flow of control from one activity or block to another. See also *control flow*, *data connector*, and *default connector* and *transition condition*.

control flow. In LOVEM: on PLOVCs and JLOVCs, a trigger that is generated by an activity. It shows the flow of control from one activity or task to another, provided the transition condition, if specified, is true. See also *information flow* and *material flow*.

In process-based applications: a control flow can consist of a workflow, a task flow, and an event flow. See also *workflow*, *task flow*, and *event flow*.

critical measurement point. Any point on the process path or within a job that is of critical importance, and where a measurement should be taken. Measurements can be in units of time or quantity, for example, time to answer a customer inquiry, cycle time, number of invoices per unit of time, error rate, and so on.

critical path. Taking into account all the dependencies and processing requirements for achieving a major goal or target, the critical path is that sequence of events that takes the longest time to reach the final goal.

critical success factor (CSF). A qualitative or quantitative measure that defines the quality or performance of an enterprise, a process path, or a job, such as the required skills of employees to perform a certain task. A measurable internal or external business result that has a major influence on whether or not an enterprise achieves its goals. CSFs can be assigned at the following levels:

- The entire industry
- The enterprise
- An organizational unit, such as a business area, department, or job
- Individuals, such as managers or employees

CSD. Corrective Service Diskette (software updates for OS/2 programs).

CUA. Common User Access.

customer. A person or business that acquires products or services from an enterprise.

customer activity. In physical models or blueprints, it depicts the activities performed by customers. A customer activity can start or end a chain of activities.

customer expectation. A description of what a customer needs or wants. This can be in terms of products, services, or the performance of an activity or a system.

customer process. On logical models or blueprints, it depicts the actions or processes carried out by customers.

customer satisfaction. The goal of a customer-oriented business. Customer satisfaction occurs when a customer receives as much as, or more than, expected from a product or service. It is usually measured as the number of satisfied customers as a percentage of the total number of customers. Customer satisfaction can be graphically shown on LOVCs by an icon indicating whether a customer is *happy* or *unhappy* with the operations and results of an activity.

cycle time. The time it takes to complete a process path. For example, for the order fulfillment process, the cycle time is the time it takes to fulfill an order (from

when a customer places an order to when the customer receives the product).

D

DASD. Direct Access Storage Device. A device in which access time is effectively independent of the location of the data.

data bus. On an ALOVC or a LLOVC, a logical set of data. A logical, dynamic data store. It Starts at the beginning of a logical model, such as ALOVC and LLOVC and continues until the end of the model. It reflects the most current data at any point in a relationship with a customer. Examples of data buses:

- Contact or customer data bus
- Offering or products and services data bus
- Promise or order and contract data bus

data component. A packet of data with which a process deals.

data connector. In FlowMark: the graphical representation of the flow of data in a process diagram. See also *data flow*, *information flow*, *control connector*, and *default connector*.

data container. In FlowMark: storage for the input and output of an activity, a block, or a process. See also *input container* and *output container*.

data dependency. Data that a process requires to proceed. An *upstream data dependency* is data required from the preceding process. A *downstream data dependency* is data required by the next process.

data entity. See *entity*.

data flow. A packet of data in motion. It can consist of data groups, data entities, and data elements. A data flow identifies the data that is either generated from or required by, the customer or internal processes to which it is connected. See also *information flow*.

data store. A logical set of data or a physical place where you can keep information (desks, filing cabinets, databases, and personal computer files are examples of *physical data stores*).

DB2/2. IBM Database 2 for OS/2, a relational database manager.

DDE. Dynamic data exchange. An OS/2 or Microsoft Windows feature that enables data exchange between applications.

decomposition. The process of breaking a large entity into smaller components. For example, a process can be decomposed into subprocesses, or an activity into tasks.

default connector. In FlowMark: the graphical representation of a special kind of control connector, shown as in a process diagram. Control flows along this connector if no other control path is valid. See also *connector*, *control connector*, and *transition condition*.

department. A subdivision of an enterprise that shows reporting lines and performs one or more activities. Department is part of the physical models, such as PLOVC or HSD.

design point. The primary focus of a design; for example, the customer, or a workflow solution.

designing. The creative process used to plan, sketch, and model the new business processes, process paths, and jobs in order to create a set of detailed blueprints from which the new design can be implemented.

DLL. Dynamic link library. A module containing a routine that is linked at load time or run time. FlowMark uses the *.DLL filetype under OS/2 as well as under MS Windows.

document. A transmission medium for, or depository of, information, such as a report or an invoice.

document flow. The flow of documents through an organizations. This can be through a document management system in digitized form or in hardcopy form.

document storage. A physical data store where hard-copy documents are stored. Can be part of the physical models or blueprints, such as PLOVC and JLOVC. See also *data store*.

downstream. Subsequent processes or activities, for example the *distribute* process is downstream from the *order* process. See also *data dependency* and *upstream*.

E

enterprise. An organization, usually comprised of several lines of business, whose purpose it is to perform a mission and to achieve goals by working with customers and suppliers.

enterprise architecture. A business process architecture at the enterprise level as expressed through an ALOVC. It is at the logical, unconstrained level and shows the essential process and data needs in sequential form. See also *architecture line of visibility chart (ALOVC)*.

enterprise process. The actions or processes performed by the enterprise.

entity. A thing or object of importance to a business about which the business wants to keep information, such as customer or product.

event flow. In process-based applications, including FlowMark, an event flow is part of the control flow. It triggers the continuation of activities that are in a wait status. See also *control flow*, *workflow*, and *task flow*.

exit condition. A control setting for an activity of a PLOVC or JLOVC that determines when an activity is complete and control is passed to the next activity. It also determines when a process path or workflow is completed. See also *start condition*, *start criteria*, and *exit criteria*.

exit criteria. The conditions that determine when an activity has completed its actions. See also *start criteria*.

external organization. In a PLOVC or JLOVC, an organization unit, such as a government agency, a bank, or a supplier, that is part of a process path, but outside the organization of the enterprise.

F

FDL. See *FlowMark Definition Language*.

FlowMark. An IBM program product that manages and controls the execution of a process path or workflow.

FlowMark Definition Language (FDL). An external format for defining staff, programs, data structures, and workflow models in a flat file. The definitions in the FDL file can then be imported into a FlowMark database.

FRL. FlowMark Runtime Language. An external format for templates, instances, and work items in a flat file. The export utility program EXMPFREX.EXE located on the FlowMark database server exports the runtime database to an FRL file. Using the import utility program EXMPFRIM.EXE, an FRL file can be imported into a FlowMark runtime database.

formalism. The strict attention to rules and symbols (for example, the LOVC rules and symbols).

G

goal. The statement of an enterprise's objectives or direction. A business target that is to be met within a given time. Goals can be qualitative, such as *to become the best-of-breed*, or quantitative, such as *to increase revenue by 20% over the next 12 months*. Goals exist at the following organizational levels:

- The enterprise
- An organizational unit, such as a business area, department, or job

- Individuals, such as managers or employees

H

hand-off. The passing of a product, information, or other materials from one department or workstation to another.

hierarchical structure diagram (HSD). A graphical modeling technique, which shows the hierarchical structure of organizations, processes, activities, goals, CSFs, and systems. Its major use is the systematic refinement of objects.

HLLAPI. High-level language application program interface. HLLAPI is used by application programs for host communication. FlowMark makes use of the HLLAPI building block under OS/2 or MS Windows.

HSD. See *hierarchical structure diagram*.

I

IBM LOVEM. An IBM business process engineering or re-engineering methodology, which can be applied at the following levels:

- Enterprise architecture
- Logical process path model or blueprint
- Physical process path model or blueprint
- Job model or blueprint

Note: There are two levels of IBM LOVEM:

1. Line of Visibility Engineering Methodology

This level contains the full methodology, which is documented in the IBM LOVEM Consultant's Guide, and which is only available to IBM consultants.

2. Line of Visibility Enterprise Modeling

This level contains the graphics and applications of the methodology that are implemented in ProModeler. This level is documented in the IBM LOVEM User's Guide and is available to the general public.

icon. A picture that represents the actual image of information flow media or means of transportation, such as a telephone, truck, or computer terminal.

information. Facts or objects that have meaning to human beings; as opposed to *data*, which requires context and interpretation in order to become *information*. Information is formatted data. Business objects that are produced by or moved between activities and systems using information flows.

information flow. An ordered packet of data. Input to, or output from, any object on a PLOVC, or JLOVC,

such as an order, a shipping document, or an invoice. Information flows can use various media, such as FAX machines, telephones, or electronic mail, which can be represented on the LOVCs by icons. See also *data flow*, *material flow*, and *control flow*.

information system. See *system*.

input container. In FlowMark: storage for data used as an input for activities, processes, or blocks. See also *output container*.

inquiry management bus. On the ALOVC, a continuum of repetitive and unpredictable processes for enabling customers to request information on anything about the company, its products and services, or a past service encounter (for example, an inquiry about a proposal, a contract, or an order). See also *bus* and *change management bus*.

internal interface. An interface between two internal business functions, departments, or jobs where a dependency exists or a transfer takes place.

issue. A matter at dispute that needs to be discussed and resolved.

J

JLOVC. See *job line of visibility chart*.

job. The physical implementation of business processes, as expressed through manual activities and interfaces with customers, systems, and internal or external organizations. A series of one or more activities in a department that are performed by one employee. A job is represented by the JLOVC and can be part of a PLOVC. Examples of a job: marketing representative or customer service representative.

job line of visibility chart (JLOVC). A physical modeling and blueprinting technique that shows the *effectiveness and efficiency* of one particular job. Using the JLOVC, you focus on *how* an individual job is or will be implemented, and *who* is or will be executing the manual activities. It also shows the relationship of each activity to customer activities, systems, and internal or external organizations. See also *line of visibility chart*.

L

LAN. Local area network.

line of business (LOB). A family of products or services, having common characteristics.

line of visibility (LOV). On an LOVC, the line

between your customer and internal processes or activities where all points of contact (service encounters) are shown. See also *service encounter*.

line of visibility chart (LOVC). The graphical representation of all aspects of your business processes that are required to provide your customer with a specific product or service. It shows all points of contact with your customer. The LOVC is implemented in four different types of charts:

- ALOVC. See *architecture line of visibility chart*.
- LLOVC. See *logical line of visibility chart*.
- PLOVC. See *physical line of visibility chart*.
- JLOVC. See *job line of visibility chart*.

Line of Visibility Engineering Methodology (LOVEM). See *IBM LOVEM*.

Line of Visibility Enterprise Modeling (LOVEM). See *IBM LOVEM*.

LLOVC. See *logical line of visibility chart*.

load balancing. Attempting to ensure that each worker gets the same amount of work to perform. Can be automated or manual.

LOB. See *line of business*.

location. A physical place where activities are performed or where information or materials are stored.

logical. The abstract or generic nature of business processes or data before any physical constraints are applied. Logical defines *what* process or data is required, not *how* it is implemented. See also *unconstrained*.

logical line of visibility chart (LLOVC). A logical modeling or blueprinting technique that shows the *effectiveness* of the process path that you are studying. Using this technique, you focus on *what* needs to be done, not *how* it is done. See also *line of visibility chart*.

logical model or blueprint. The depiction of the effectiveness of a process: *doing the right thing*. See also *logical*, *model*, and *blueprint*.

logical-to-physical transformation. The translation of a logical process into its physical implementation components, such as manual activities or systems functions. See also *logical transformation list*.

logical transformation list (LTL). A technique for transforming logical processes into physical implementation scenarios. For example, a logical process can be transformed into any number of manual activities, any number of systems functions, or both.

loop. A loop is an iteration of activities on a PLOVC or JLOVC. There are two sets of exit criteria for a loop:

1. One set contains the criteria for exiting the loop through the normal flow when the exit conditions are met.
2. The other set contains the criteria for how often the flow can go through the loop before terminating it if the first criteria are not met. This set also has to describe where the flow continues in case of an abnormal termination.

loop connector. A symbol on a PLOVC or JLOVC that points back to the starting point of a loop. The loop connector contains the short and long names of the activity, where the loop starts.

LOV. See *line of visibility*.

LOVC. See *line of visibility chart*.

LOVEM. See *IBM LOVEM*.

LTL. See *logical transformation list*.

M

manual/automation interface. The manual-automation line on a PLOVC or JLOVC between manual activities and systems. Systems placed on this line show user interactions with the systems. Systems placed below this line are batch systems with no direct user interaction.

material. A commodity that is of value for the business process. Materials are used or worked on by an activity or system and are transported between activities and systems. See also *material flow*.

material flow. Any tangible product or document that is generated by an activity or system.

In LOVEM: input to, or output from, an activity or system on a PLOVC or JLOVC; for example, a car, a mortgage document, or a consultant's report. The mode of transportation, such as courier, airplane, or truck, can also be shown on the diagram as icons. See also *information flow*, *data flow*, *document flow*, and *control flow*.

measurement. The extent, quality, or size of an object. For example, the measurement of the object *box* can be expressed by *volume*, *height*, *weight*, and the measurement of the object *process* can be expressed by *effectiveness*, *cost*, *duration*, or *maturity*. Measurements can be used for benchmarking. See also *benchmarking*.

measurement point. A designated point in a process path where measurements are to be taken. See also *critical measurement point*.

methodology. A collection of related techniques and notations based on a common philosophy to solve a series of major tasks. See also *IBM LOVEM*.

metrics. The definition and description of a procedure for taking measurements. Metrics can be assigned to activities as *actual*, *target*, or *ultimate* values.

mission. The purpose and nature of an enterprise.

model. The graphical and written representation of observations and predictions of how a design could or should be implemented. Models are usually built at various levels of abstraction and detail. For example, a business model depicts a defined business area that is important to the enterprise; it can be shown as different views of the same business area, such as:

- Process or process path
- Organization
- Performance

modeling. Part of the design process used to create alternatives or *what if* scenarios before committing to the final design. See also *model*.

moment of truth. Your customer's perception resulting from any contact that your company has with that customer either in person or through a document, product, or system. See also *service encounter*.

MQSeries. Message Queue Series: a communications layer that establishes connection between two systems. With MQSeries, messages can be sent and received through queues.

N

navigation. The movement from a completed activity to downstream activities. The paths followed are determined by control parameters, their associated transition conditions, and by the start conditions of activities. See also *control connector*, *start condition*, *exit condition*, and *transition condition*.

navigation evaluation. The process FlowMark uses to decide which path to take. See *navigation*.

node. A point at which one or more functional units connect. In a process diagram, nodes are the symbols or objects that can be joined by connectors or flows, such as activities, systems, blocks, sources, and sinks.

O

opportunity area (OA). A point in a process or process path where possibilities, advantages, or other positive factors can help an enterprise meet its goals.

organization. An administrative unit of an enterprise. In FlowMark: organization is one of the criteria that can be used to dynamically assign activities to people. See also *role*, *child organization* and *parent organization*.

organization unit. An administrative subdivision with reporting lines that implement processes; for example, a department.

OS/2. IBM's Operating System/2 for workstations.

output container. In FlowMark: storage for data produced by an activity, process, or block for use by other activities. It can also be used for evaluation of conditions. See also *sink*.

P

parent organization. In FlowMark: an organization within the hierarchy of administrative units of an enterprise that has one or more child organizations. A child is one level below its parent in the hierarchy. See also *child organization*.

pattern activity. In FlowMark: the single program or process activity in a bundle from which multiple instances, called bundle activities, are created. See also *bundle activity*.

PC. Personal computer or workstation.

physical. The concrete, specific, or constrained nature of business processes and data. Physical defines the *how*, *where*, *when*, or *by whom* a process is performed or data is used. See also *constrained*.

physical constraints. See *constrained*.

physical line of visibility chart (PLOVC). A business process modeling or blueprinting technique that shows the *effectiveness and efficiency* of a process path. A PLOVC is a sequence of physical business objects, such as activities and systems. See also *physical* and *line of visibility chart*.

physical model or blueprint. The depiction of the efficiency of a process: *doing the thing right*. The physical model or blueprint shows the *how*, *where*, *when*, or *by whom* aspects of an enterprise, such as the resources required to perform a process or process path. See also *model*, *blueprint*, and *physical*.

planning activity. In FlowMark: a special program activity that creates, at run time, the required number of bundle activities for a specific bundle. The planning activity must use a program that refers, in its registration, to the bundle-planning tool supplied with the FlowMark product. See also *program activity*, *program registration*, and *bundle activity*.

platform. The operating system environment in which a program runs. FlowMark is a distributed, cross-platform application, which can run on OS/2, AIX, and Windows.

PLOVC. See *physical line of visibility chart*.

policy. A principle, plan, or course of action pursued by an enterprise.

problem. An obstacle to meeting a goal or fulfilling a CSF. A problem can also be a situation or issue that is uncertain, complicated, or difficult to deal with.

problem area (PA). A point in a process or process path where difficulties, constraints, or other negative factors prevent the enterprise from meeting its goals or CSFs.

procedure. A series of steps or activities required to perform a process.

process. A business function or operation that achieves results for customers with input from suppliers. A process transforms the nature, status, or composition of input to produce output according to business rules and policies. A process is a means to achieving the goals and strategies of an enterprise. See also *subprocess*.

In LOVEM-ProModeler: a process is a logical component on an ALOVC or LLOVC.

In FlowMark: a process is a set of activities that must be completed to accomplish a given task.

process activity. In FlowMark: an activity to which a separate process is assigned. Starting this activity creates an instance of the referenced process and starts it. See also *program activity*.

process administrator. In FlowMark: the person responsible for the execution of a process instance. A process administrator can be specified in the workflow model; otherwise it is the person who starts the process.

process category. In FlowMark: an attribute that a process modeler specifies for a process. Only users who are authorized for this category can start and control instances of the process as top-level. See also *top-level process*.

process cycle time. The elapsed time required to receive, process, and forward a transaction.

process diagram. A graphical representation of a process or process path that shows all its components.

process instance. In FlowMark: an executable copy of a process template in Runtime.

process management. In FlowMark: the Runtime tasks associated with process instances, such as creating, starting, suspending, resuming, terminating, restarting, and deleting process instances. See also *business process management (BPM)* and *process path management*.

process manager. A manager who has the delegated responsibility from a process owner to manage the day-to-day operations of a process or process path. See also *process owner*.

process owner. A senior manager who is responsible for managing all aspects of a business process or process path. A process owner usually delegates the operational management to process managers. See also *process manager*.

process path. A sequence of processes or activities and flows of data or information that produce a specific product or service. A process path usually starts and ends with a customer service encounter. See also *service encounter*.

process path management. The management of a business across the traditional vertical processes or organizations; for example, in a traditional order fulfillment company, the vertical processes are *sell, order, supply, distribute, settle*.

process quality management bus. On the ALOVC, a continuum across the enterprise process path to capture any process quality parameters, such as CMPs, CSFs, goals, strategies, or policies in relation to customer service encounters. See also *bus*.

process status. In FlowMark: the status of a process instance. The status can be one of the following:

- Ready
- Pending
- Running
- Suspended
- Terminated
- Finished

process template. In FlowMark: the translated form of a workflow model in Runtime.

program. In FlowMark: a computer-based application that supports the work to be done in an activity. Program activities reference executable programs using the logical names associated with the programs in FlowMark program registrations. See also *program registration*.

program activity. In FlowMark: an activity to which a registered program is assigned. Starting this activity invokes the program. See also *process activity*.

program registration. In FlowMark: identification of a program to a FlowMark database, so that it can be assigned to a program activity in a workflow model. See also *program activity*.

R

recommendation. Advice or suggestion on how to meet a goal, solve a problem, evaluate a CSF, or carry out a strategy or policy.

refinement. A standard modeling technique used to view the parts of a whole in increasing amounts of detail. See also *hierarchical structure diagram (HSD)*.

relationship. A descriptive association between two data entities or relationships in a data model.

report. Formatted text and graphics, usually generated by a system.

report layout. The design and specifications for the format of a printed report. See also *screen layout*.

repository. An organized, shared collection of data or information that supports business process re-engineering, application development, or business or systems management. It is usually automated and is implemented as a database.

REXX. Restructured extended executor language. A procedures language.

RISC. Reduced instruction set computer (runs AIX).

role. In FlowMark: a responsibility that is defined for staff members. Role is one of the criteria that can be used to dynamically assign activities to people. See also *organization*.

root cause analysis. The analytical determination of the cause of the symptom of a problem.

Runtime. In FlowMark: the component used to execute process instances. The Runtime component consists of:

- Runtime server
- Program execution client
- Bundle server
- Notification service
- Delivery server
- Runtime client

See also *Buildtime* and *Runtime client*.

Runtime client. In FlowMark: the user interface for working with process templates, process instances, work lists, and work items. See also *Runtime*.

S

screen layout. The design and specifications of the image that the user sees on the screen of a system. See also *report layout*.

service encounter. Any point of contact with your customer. See also *moment of truth*.

shredder. A machine for the destruction of documents.

simulation. A mock-up version or prototype of the new process and job design used to test assumptions before final implementation.

sink. In FlowMark: the symbol that represents the output container of an activity, process, or block. See also *output container* and *source*.

skill. An ability, proficiency, expertness of a person that comes from training, practice, and experience.

source. In FlowMark: the symbol that represents the input container of an activity, process, or block. See also *input container* and *sink*.

staff. In FlowMark: the people (and their roles, organizations, and levels) who execute the process instances. Staff is defined in a FlowMark database. See also *role* and *organization*.

staff resolution. The process FlowMark uses to decide which worklists to add a process to.

start activity. In FlowMark: an activity that has no incoming control connector. A start activity becomes ready when the process or block that contains it starts. There can be more than one start activity in a process or block.

start condition. A control setting that determines when an activity with incoming control connectors can start. It also determines when a process path or workflow can start. See also *condition*, *exit condition*, and *transition condition*.

start criteria. A control setting for activities on PLOVCs or JLOVCs that determines when an activity or a process path can start. See also *exit criteria*.

strategy. A pattern of goals, policies, and plans that specify how an enterprise is to function over a given period of time. A strategy can specify areas for product development and marketing as well as techniques for responding to competition.

subject expert. A specialist in a particular area of expertise, such as workflow management.

subprocess. A lower level process. Processes can be refined into subprocesses through the HSD modeling techniques. See also *hierarchical structure diagram (HSD)* and *process*.

In FlowMark: a process instance that is started by a process activity.

substitute. In FlowMark: the person to whom an activity is automatically transferred if the person to whom the activity is assigned is flagged as absent.

symbol. A graphical representation of an object or thing, which may be abstract in nature; for example, a line with an arrow is the symbol for a data or information flow.

system. A set of processes with a common aim that act on data or information using input and producing output. Usually used in the context of *information system* or *data processing system*.

system development. The design, code, test, implementation, and maintenance of an information system. Can also denote a business function that performs systems development.

system function. A component or module of a system. A system can be refined into system functions using the HSD modeling technique. See also *hierarchical structure diagram (HSD)* and *system*.

T

task. The lowest level of activity or unit of work. Tasks belong to the physical business models. There is no implied sequence or order in performing tasks within an activity. Activities can be refined into tasks using the HSD modeling technique. See also *hierarchical structure diagram (HSD)* and *activity*.

task flow. In process-based applications, a task flow is part of a control flow. See also *control flow*, *workflow*, *event flow*, and *document flow*.

TCP/IP. Transmission control protocol/Internet protocol. A commonly used protocol for various network environments, such as Internet, Host communications, and LANs. FlowMark uses either TCP/IP or APPC. See also *APPC*.

technique. A procedure for doing anything in an orderly way; a method.

time line. A notation on the PLOVC and the JLOVC that shows the time between individual activities as well as for the entire process path or job; for example, the

order cycle time. The time line shows both actual (As Is) and target (To Be) times.

To Be. The desired state of a process or process path: how it could be or should be. See also *As Is*.

To Be view. A chart or diagram showing the desired state of a process or process path. See also *As Is view*.

top-down. Modeling or designing business processes from their most abstract level down to their most concrete and constrained levels of detail. See also *bottom-up*.

top-level process. In FlowMark: a process that is started from a user's process list or from an application program.

transition condition. In FlowMark: a logical expression associated with a control connector. If specified, it must be *true* for control to flow along the associated control connector. See also *control connector*, *default connector*, *condition*, *start condition*, and *exit condition*.

trigger. An event or condition that starts or ends an activity, process, or process path. See also *start condition* and *exit condition*.

U

unconstrained. Pertaining to, or characteristic of, the ALOVC and LLOVC techniques, or representative of the factors that define *what* a process is doing not *how* it is being done. See also *logical*.

unit of measure. A standard dimension, extent, or quantity, such as days, hours, or minutes; dollars or cents; or meters or centimeters. A unit of measure is used for measurement purposes.

upstream. Preceding processes or activities; for example, the *order* process is upstream from the *distribute* process. See also *downstream*.

V

VHLPI. Visually high-level programming interface. The service broker for Visually. It can be used to integrate FlowMark with Visually for document management.

VisualBasic. A programming language under MS Windows.

VisualInfo. An IBM product for document management.

V2R1. Version 2 Release 1.

V2R2. Version 2 Release 2.

V2R3. Version 2 Release 3.

W

WISDDM. World-wide integrated solution design and delivery methodology. WISDDM is a set of I/T methodologies, such as:

- The application development methodology, which is based on:
 - Solution/2000
 - End-to-End
 - Full Life-Cycle Testing
 - Redevelopment
 - Package Selection
- The project management methodology, which is based on:
 - managing implementation of the total project
 - project management for project executives
 - other IBM project management approaches

workflow. A sequence of activities (units of work). A movement of units of work through a process. In process-based applications, it can be part of a control flow. See also *control flow*, *task flow*, and *event flow*.

workflow management. The art of controlling or administering a sequence of activities.

workflow model. In FlowMark: a complete representation of a process. It consists of the process diagram and settings and the definition of staff, programs, and data structures that are associated with the activities of a process.

work list. In FlowMark: a list of work items assigned to a staff member.

Index

A

- activity 1, 3, 6, 7, 8, 11, 12, 13, 14, 15, 17, 19
 - automatic 12
 - block 1, 15
 - networks 1
 - notebook 13
 - staff pages 13
 - volumes 11
- activity block 2
- activity, automatically started 1
- activity, unattended 1
- additional hardware 3
- AIX 5, 6
- application 1, 3, 5, 6, 7, 11, 12, 17, 19
- assign 1
- audit trail 3, 9
- automate 11
- automatic 9, 12
- automatically started 1

B

- bibliography 31
- Boolean expression 1
- Buildtime 1, 2, 6
- business process 7
 - automating 7
- business processes 1
- business requirements 5
- business rules 3

C

- capacity 3, 5, 7, 10, 11, 14, 17, 19
- cleanup interval 10
- client 11
- client response time 21
- client workstation, no change 5
- client/server 5
- communications protocol 6
- conflicts 13
- connectors 1
- construct 15
- container 7
 - source 7
- control flow 1
 - conditional 1
 - unconditional 1
- cost 5, 15
- customer 3, 7, 12, 19
 - design 12

D

- data container 2, 14, 19
 - input 19
 - output 19
- data elements 19
- data flow 1
- data structure 2, 6
- Data Structure Definition facility 2
- database 3, 7, 10, 12, 13, 14, 17, 19
 - size 7
- database space 10
- definition facility, FlowMark 1
- delete finished items 9
- deleting 10
- design 5, 11
- disk I/O 6

E

- end-user training 21
- exception 7, 8, 12
 - example of 7
- exceptions 3, 7, 11, 17
- exit condition 8, 15, 17
- ExmcStartProcess API 9
- ExmDeleteProcess API 10
- ExmTerminateProcess API 9

F

- filtering 19, 21
- find 19
- flow 1, 3
 - control 1
 - data 1
- FlowMark 4
 - constraints 4
 - enhancements 4
 - new functions 4
- FlowMark API 7, 9, 14
- FlowMark definition facility 1
- frequently asked questions 3

G

- glossary 37

H

- hardware cost 5
- hardware, additional 3

headquarters 5, 6

I

impact 3, 6, 11, 14, 19, 22

include 19, 21

input container 6

instance 3

K

knowledge workers 11

L

LAN 3

load balancing 13

M

memory (swapping) 6

monitor 9

N

navigation 1, 19

navigation evaluation 11

network 13, 21

network address 6

nodes 1

notebook, personal data settings 9

notification servicer 10

O

off-load 10

organization 13

OS/2 5

overhead 11, 14, 15, 19

P

pass data 3

peak times 19

people 1

performance 3, 5, 7, 10, 11, 12, 14, 17, 19

performance capacity 15

personal data notebook 22

personal data settings notebook 9

pointers 19

price/performance 5

process 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 17, 19, 21

definition 1

deleting 9

automatically 9

manually 9

process (*continued*)

design 1, 2, 3, 6, 11, 12, 13, 21

ongoing 3

example 8, 12, 14

example of 5, 7

exception 8

exceptions 7

finished 9

flow 3

guidelines 7

instance 1, 9, 19

invoking 17

models 1

overhead 8

parent 17

performance 3

short 8

size of 7

starting 9

template 5, 7, 9, 12, 14

terminated 9

Process Definition facility 1

process flow 1

process list 21

process model 7

process-relevant data 19

processor utilization 6

program 1, 12

design 12

programs 1

assignment of 1

R

regional 6

resources 6

response time 1, 6, 19

poor 6

response time, client 21

response, runtime client 19

reusability 17

role 13

roles 1

RS/6000 5

runtime client response 19

S

server 1, 2, 3, 5, 6, 7, 9, 10, 11, 17, 19, 21

AIX 5

client workstation, no change 5

dedicated machine 6

Intel 5

multiple 5

dividing the work 5

independence of 6

mix of users 5

- server *(continued)*
 - multiple *(continued)*
 - no communication 6
 - subprocess 6
 - OS/2 5
 - planning 5
 - regional 5
 - registration 2
 - single 5
 - step up 5
 - Unix 5
- Server Definition facility 2
- shutdown 21
- signoff 21
- signon 21
- skill level 13
- staff 1, 5, 13
 - allocation of 1
- staff definitions 17
- staff resolution 6, 7, 11
- structure 19
- subprocess 2, 5, 6, 15, 17
 - activity notebook 17
 - on another FlowMark server 17
 - remote 2

T

- thinking 11
- trade-offs 17
- training, end-user 21

U

- unattended 1
- user 1

V

- volumes, workload 3

W

- work list 1, 8, 10, 11, 13, 19, 21, 22
 - large 22
 - roll 13
 - shared 22
- work lists 13, 21
- workflow 1, 4
 - engine 1
 - management 1
- workload volumes 3

ITSO Redbook Evaluation

Image and Workflow Library: FlowMark V2.3 Design Guidelines
SG24-4613-02

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at <http://www.redbooks.ibm.com>
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction _____

Please answer the following questions:

Was this redbook published in time for your needs? Yes____ No____

If no, please explain:

What other redbooks would you like to see published?

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

