ABSTRACT

DEVELOPING ONLINE HELP AT BLUESPRING SOFTWARE: AN INTERNSHIP

by Stacey P. Scott

This report discusses my internship at Bluespring Software Inc. in Cincinnati, Ohio, where I worked as a User Services Analyst from March to June 2004. In Chapter 1, I discuss the organization and culture of Bluespring Software. I then describe my position at Bluespring and my responsibilities as an intern in Chapter 2. In Chapter 3, I discuss my major project—documenting BPM Suite 4.0, Bluespring’s latest software application. This included writing online help and creating online demonstrations for the application. In Chapter 4, I analyze my internship by describing how I used Paul Anderson’s problem solving model to create my documentation plan, and I reflect on how my experience with a volatile start-up company like Bluespring cemented my views on the corporate culture in which I would like to work.
DEVELOPING ONLINE HELP AT BLUESPRING SOFTWARE: AN INTERNSHIP

An Internship Report

Submitted to the
Faculty of Miami University
in partial fulfillment of
the requirements for the degree of
Master of Technical and Scientific Communication
Department of English

by
Stacey P. Scott
Miami University
Oxford, Ohio
2004

Advisor___________________________
(Dr. Jennie Dautermann)
Reader____________________________
(Dr. Janel Bloch)
Reader____________________________
(Dr. Glenn Platt)
DISCLAIMER

This internship report includes information, data, copyrighted work, trademarks, trade names, logos and other intellectual property (collectively "Intellectual Property") of Bluespring Software, Inc. ("Bluespring Software"). Bluespring Software has granted the author of this report permission to use this Intellectual Property solely within and in the context of this report. None of the Intellectual Property may be extracted, copied, reproduced, distributed, republished, downloaded, displayed, posted or transmitted in any form or by any means from or outside of the context of this report, including but not limited to electronic, mechanical, photocopying, recording, or other means, without the prior express written permission of Bluespring Software. Except for the limited right to use described above, nothing in this report shall be construed to confer to the author hereof or any other person or party any license or other rights in or to any of the Intellectual Property, whether by estoppel, implication, or otherwise. Any unauthorized use of any of the Intellectual Property may violate copyright laws, trademark laws, the laws of privacy and publicity, and communications regulations and statutes.
# TABLE OF CONTENTS

**Chapter 1: Introduction** ........................................................................................................ 1  
The Organization and Culture of Bluespring................................................................. 1  
Bluespring before the Split ...................................................................................... 2  
Bluespring after the Split ......................................................................................... 2  
Corporate Culture at Bluespring .............................................................................. 4  
Technical Communicators at Bluespring .................................................................. 4  

**Chapter 2: Internship Overview** .................................................................................. 6  
My Position at Bluespring ....................................................................................... 6  
My Responsibilities during my Internship ............................................................... 6  
Learning about Bluespring’s Day-to-day Operations .............................................. 7  
Developing Documentation for BPM Suite 4.0 ....................................................... 8  
Editing Other Projects .............................................................................................. 8  
Working on Bluespring’s Human Resource Initiative ............................................. 9  

**Chapter 3: Major Project – Documenting BPM Suite 4.0** ....................................... 10  
BPM Suite 4.0 ........................................................................................................ 10  
The Status of the Project when I Arrived .................................................................. 11  
  The Scope ........................................................................................................ 11  
  The Purpose .................................................................................................. 11  
  The Audience ............................................................................................... 11  
  The Medium ................................................................................................. 12  
My Role in Documenting BPM Suite 4.0 ................................................................. 12  
  Researching the BPM Suite 4.0 ................................................................ 12  
  Writing the Online Help Topics .................................................................. 15  
  Submitting the Topics for SME Review ..................................................... 16  
  Editing the Online Help Topics ................................................................. 17  
  Developing the Online Demonstrations ..................................................... 17  
  Testing the Online Demonstrations ............................................................ 18  

**Chapter 4: An Analysis of the Internship** .................................................................. 20  
Using Paul Anderson’s Problem-Solving Model ......................................................... 20  

iii
LIST OF FIGURES

Figure 1. Bluespring Organizational Chart in March 2004. .............................................. 2
Figure 2. Bluespring Organizational Chart after April 15, 2004. ......................................... 3
CHAPTER 1: INTRODUCTION

To fulfill my internship requirements for the Master of Technical and Scientific Communication at Miami University, I worked as a technical communicator at Bluespring Software, Inc. in Cincinnati, Ohio. My official title was User Services Analyst – Intern. I performed this internship from March 1, 2004 to June 11, 2004. For the first ten weeks of the internship, my supervisor was Chris Berger, and for the remainder of the internship, my supervisor was Karl Trier. My writing mentor was Shawn Mummert.

THE ORGANIZATION AND CULTURE OF BLUESPRING

Bluespring Software Inc. was founded in 1997 as a software management group. Bluespring’s five founding members were telecommunications executives who wanted to develop and sell a software application that managed telecommunication business procedures. These procedures included analyzing telecommunication orders for commercial customers, placing and tracking orders, and billing customers for the telecommunication services provided. In 1997, Bluespring’s target market included tier-1 telecommunication providers like AT&T, Cincinnati Bell, and Verizon, although that emphasis has changed somewhat now.

In light of the contracting of the telecommunications industry in 2001, Bluespring has now refined its business goals and produced an application that can be used within other industries to manage any business processes. This technology is referred to as Business Process Management (BPM) software and its application is generic enough to be applicable to most industries. For example, in the property management industry, BPM software can be used to manage the tracking of tenant service order requests. Apartment tenants can enter service requests using their landlords’ Web site, and their requests will then be directed to a maintenance supervisor, placed automatically in a queue, and assigned to a maintenance provider. The maintenance provider then completes the request and updates the request status on the landlord’s Web site. Both the maintenance supervisor and the tenant are notified of the status of the request. The BPM software integrates user inputs and automates business rules to make business processes run seamlessly.
During my internship, increased pressure from Bluespring’s investors to meet profitability targets led Bluespring’s shareholders to split the company into two entities: telecommunications services and BPM services. This split affected all 24 of Bluespring’s employees and reduced employee morale. As an intern, I wondered what this split would mean for me in terms of long-term employment at Bluespring and the overall stability of working for such a small, volatile company.

**Bluespring Before the Split**

When I began my internship in March 2004, Bluespring employed 23 full-time employees, one part-time employee, and two technical communicator interns—I was one of them. The organization chart below in Figure 1 shows Bluespring’s structure when I began.

![Bluespring Organizational Chart in March 2004.](image)

I reported directly to the Vice President of Solutions Delivery and Business Development. On April 15, 2004, we learned of the shareholders’ decision to split the company into two groups during a rare company-wide staff meeting.

**Bluespring After the Split**
After the split in April, Bluespring was repositioned as primarily a BPM solutions provider, and its ties to the telecommunication industry were eliminated. This split resulted in the loss of three full-time employees, one part-time employee, and five senior managers. The five managers later found positions with one of Bluespring’s partners, and other employees found positions elsewhere. My internship supervisor and the company’s Chief Administrative Officer both left for other opportunities. The chart in Figure 2 below shows the leaner structure of the resulting organization.

![Bluespring Organizational Chart after April 15, 2004.](image)

As a result of broadening its target market and making its product less telecommunications industry specific, Bluespring now focuses on other markets such as finance and property management. Although I understood that this new focus could
potentially increase profitability, as an intern I was confused by the leaner organization, unsure of my chain of command, and unclear of my place in and value to the organization. The low employee morale after the layoffs and the corporate culture at Bluespring added to my sense of confusion and uncertainty.

**CORPORATE CULTURE AT BLUESPRING**

Bluespring’s offices are located on the third floor of the Hanke building on Main Street in downtown Cincinnati. The loft-style suite consists of three conference rooms, one enclosed office for the CEO, and cubicles for all other employees. This layout promotes a very open atmosphere. Employees communicate using a free instant messaging service and by speaking to each other over the cubicles. They rarely use e-mail except for meeting notices, product updates, and company-wide announcements. Work times are very flexible and I enjoyed the casual and relaxed feeling promoted by the casual dress code.

Bluespring’s employees are self-starters who require little supervision. Most positions are filled internally, and because of its size, many employees perform multiple roles for various projects. For instance, as an intern I had three roles: as technical writer developing the online documentation for BPM Suite 4.0, as a Human Resources assistant helping to determine Bluespring’s HIPAA (Health Insurance Portability and Accountability Act of 1996) compliance, and as an editor editing a document specification sheet for a prospective client. I discuss these projects in more detail in the next chapter.

Bluespring’s motto, “Whatever it Takes ... Right Away ... All the Way ... With a Good Attitude ... Everyday! …” is posted on most of the office walls. The CEO promoted a positive and rewarding work environment by sponsoring employee lunches at nearby restaurants whenever client or production deadlines were met. During the reorganizations, these CEO-sponsored lunches also represented attempts to deal with declining employee morale, and gave me the opportunity to speak with the SMEs (Subject Matter Experts) outside of the workplace.

**TECHNICAL COMMUNICATORS AT BLUESPRING**

4
The two full-time technical communicators, Kathryn (Kayte) Rudolph and Angela Shealy, typically worked together on documentation projects, training initiatives, marketing projects, human resource projects, and on the Web site. They first reported to the VP of Solutions Delivery and then to the CIO (Chief Information Officer).

The relationship between the technical communicators and the SMEs was tenuous at times; the SMEs at Bluespring often did not view documentation as a part of the product, but rather considered documentation as a complex task that often slowed software development efforts and lengthened the production lifecycle. As the documentation I prepared underwent SME review, I experienced some of the friction the SMEs associate with developing documentation. Some SMEs suggested we take a system-oriented approach to documenting the software. I describe how I managed this friction in Chapter 3. In spite of the attitudes of the SMEs, the work of technical communicators was highly valued by our managers and was often rewarded with positive mention via e-mail to the CEO. As expected, employee morale among the technical communicators was low after the announcements about the reorganization and the layoffs, but for the most part the positive corporate culture persisted.

The remainder of this report is divided into three chapters: An overview of my internship experience and other minor projects, a discussion of my major project—Documenting BPM Suite 4.0, and an analysis of the internship, the major project, and my use of Paul Anderson’s problem solving model during the major project.
CHAPTER 2: INTERNSHIP OVERVIEW

In this chapter, I discuss my position and my responsibilities as an intern at Bluespring over the 14-week internship period.

MY POSITION AT BLUESPRING

I was hired as a User Services Analyst to develop online help for BPM 4.0 Suite—an application that takes human inputs and business processes and uses logic to automate those processes. I worked with another technical communicator and Miami University BATSC (Bachelor of Arts Technical and Scientific Communication) graduate, Christopher Schapman, to develop the help for a scaled-down version of what was first known as the Workflow Designer. The Workflow Designer was later renamed the BPM Process Designer—the “canvas” on which the automated business processes are developed.

In addition to creating the online help system, I also developed Flash movies and included those with the online help. The online help-authoring tool I used was Macromedia RoboHelp. To create the Flash movies, I used Macromedia RoboDemo, and employed Macromedia RoboSource Control for content and version control management. When needed, I used Microsoft Word as my word processor and FullShot 99 for Windows as my screen capture application. These applications were purchased in advance of my starting at Bluespring and they were provided to by my project manager at the beginning of the internship.

My work on the online help project will be included in the released software. Bluespring’s revenue primarily comes from license sales: therefore, it is critical that this software is licensed and released so it can be sold by both Bluespring and its partners to customers interested in BPM software to automate their business practices.

MY RESPONSIBILITIES DURING MY INTERNSHIP

As a User Services Analyst, my time was devoted mainly to the following four tasks:

- Learning more about Bluespring’s day-to-day operations and business practices
- Developing documentation for BPM Suite 4.0
• Editing other minor projects as needed
• Working on an HR initiative

Figure 3 below show the percentage of time I spent per internship activity over the 14-week period. As the chart shows, I worked mainly on documenting BPM Suite 4.0 for about 9 weeks—65 percent of my time.

![Percentage of Time Spent per Internship Activity](image)

**Figure 3.** Percentage of Time Spent per Internship Activity

**Learning about Bluespring’s Day-to-day Operations**

The reason I was interested in learning more about Bluespring’s primary day-to-day operations was to get a better sense of how technical communicators help Bluespring meet its primary goal of increasing profitability. Meeting this goal was very important since Bluespring’s investors were becoming concerned about Bluespring’s ability to cover its operating costs—a point the CEO discussed openly after bi-weekly shareholder meetings. After speaking with several managers and the two full-time technical communicators, I learned that Bluespring’s two primary operations are refining its product—BPM Suite 4.0—and redefining its target audience and marketing BPM Suite 4.0 to a more diverse audience.

To keep its product up-to-date, Bluespring’s software developers revise programming code daily. These programming updates, called software builds, help to make BPM Suite 4.0 a more intuitive application that works seamlessly in any Microsoft environment and that users find easy to use. Technical communicators write build notes that describe the software enhancements and the benefits of these enhancements. These build notes are sometimes written daily.
To redefine and market to a more diverse audience, the Marketing and Sales managers aggressively marketed the BPM Suite 4.0 to local companies in downtown Cincinnati and Northern Kentucky. Appendix A is a press release featured on their Web site and on MSNBC.com, an international news and business Web site. The Marketing manager released this document to get international exposure for BPM Suite 4.0. Technical communicators are also involved in the marketing efforts. The senior technical communicator, Kayte, worked with the Marketing manager to write this press release and post it to the Bluespring Web site.

**DEVELOPING DOCUMENTATION FOR BPM SUITE 4.0**

My major internship project was to document Bluespring’s BPM Suite 4.0. I worked with another intern to write online documentation and later develop Flash demonstrations for the application. This work included researching the product, writing the online help topics, submitting the topics for SME review, editing the help topics, and finally testing the topics. The tasks involved in developing this documentation are detailed in Chapter 3. Developing this documentation was similar to other documentation projects I completed for MTSC classes and during my assistantship at Miami University’s IT Services, and they involved documenting an application that was in a constant state of change. Both Bluespring and IT Services seemed to view documentation as secondary and not an integral part of the application. In both instances, the way SMEs viewed documentation troubled me since in my role as a user advocate, I would often have to defend my documentation. Both at Miami and at Bluespring, SMEs often tried to persuade me to develop system-centered rather than user-centered documentation. In Chapter 3, I discuss how I was able to reach a compromise with the SMEs about writing help topics that were user-centered.

**EDITING OTHER PROJECTS**

My other editing projects at Bluespring included editing a software specification sheet for one of our customers and editing training materials. The software specification sheet was later used to secure a five-year contract with a Canadian customer. I also edited and tested training materials the technical communicators had created for BPM Suite 4.0. These training materials will be used when customers request on-site software training.
WORKING ON BLUESPRING’S HUMAN RESOURCE INITIATIVE

Because I wanted to add value at Bluespring beyond the Documentation and Training department, so during the first week of the internship I approached the Human Resource (HR) manager about working with him on upcoming projects. He suggested I make recommendations about Bluespring’s compliance with the HIPPA regulations. The main objective of the HIPPA act of 1996 is to maintain and secure the protected health information of employees covered by employee-sponsored group health plans. The level of compliance with these regulations depends on many factors that I outlined and included in a customized decision tool for the HR manager. Using the decision tool, the HR manager decided on the appropriate level of compliance for Bluespring and later created the supporting documentation.

Although my position at Bluespring was very well defined, my responsibilities varied slightly and I performed my roles. As a User Services Analyst, I often performed writing and editing tasks. I also researched the BPM business and the human resources issues that Bluespring faced. Working on my major project and on other minor projects broaden my knowledge of the BPM business and how BPM Suite 4.0 could help other businesses automate their business processes and increase their efficiency and profitability. Working on the human resource initiative also helped me understand how Bluespring and other corporations must comply with government and federal regulation, such as complying with the HIPPA regulations.

In the following chapter, I discuss my activities during the major project of my internship—documenting BPM Suite 4.0.
CHAPTER 3: MAJOR PROJECT – DOCUMENTING BPM SUITE 4.0

The major project during my internship was developing an online help system for BPM Suite 4.0. This help system was to be included with the application and would be available to users who selected the Help menu in the taskbar or who activated the Help button within the application.

BPM SUITE 4.0

BPM Suite 4.0 is a Windows application that automates a series of business processes and turns them into workflows. The workflows combine human inputs and the business logic to create automated business processes.

In my opinion, the unique benefit of BPM Suite 4.0 is that it automates business processes that are usually poorly managed when they are executed manually since human error can be introduced at every stage of the process. Another benefit of this application is that it provides a single repository that intelligently stores human input and reorganizes it into information that business decisions managers can use. Because of these benefits, users are motivated to use BPM Suite 4.0 to make sense of complex business logic and to produce output that they can use to achieve their business goals.

In addition to the property management application I describe in Chapter 1, BPM Suite 4.0 works in the telecommunications industry with order placement. Once a corporate customer initiates an order for telecommunications service (Internet, phone, and fax lines), the customer service representative from the telecommunications provider will enter the customer’s data into a predefined order fulfillment workflow. That workflow’s protocol then assigns a technician from a predefined queue and that technician receives e-mail assignment to that customer’s account. After the technician visits the customer site and determines their needs, he or she can record recommendations in an internal tracking system that is also managed by BPM Suite 4.0. A customer service representative can then contact the customer to offer service packages and estimate prices based on the recommendations made by the BPM Suite 4.0 software. In this example, BPM Suite 4.0 uses human inputs (customer requests for service and the technician’s technical assessment) and uses the business logic of that company (packaged
and bundled service offerings) to create output (cost of service estimates) that is meaningful to both the company and its customer.

THE STATUS OF THE PROJECT WHEN I ARRIVED

When I arrived, my project manager, Kathryn (Kayte) Rudolph, had already defined the main objectives of the project including its audience, purpose, and medium. My coworker, Chris, and I were to validate the project’s objectives and modify the project plan, if necessary. Kayte prepared a preliminary task list for the project, see Appendix B for an excerpt from the task list. After conducting task and audience analyses using that task list, we concluded the following about the scope, purpose, audience, and medium for the project.

THE SCOPE

The scope of the project was to prepare online documentation for the BPM Suite 4.0 application over a 14-week period. If time allowed, I would prepare online demonstrations to complement the online help text. Complete, accurate, and user-centered online help was the first priority.

THE PURPOSE

The purpose of the online documentation was to help Bluespring meet its customer service and support goals. The main objective of the documentation was to provide just-in-time assistance to the application’s users and to reduce the number of customer service calls to the Bluespring TAC (Technical Assistance Center.) The documentation would accompany the application and would eventually be used to create training materials for the instructor-led modules and e-learning solutions Bluespring could market separately to its customers.

THE AUDIENCE

The intended audience was BPM Suite 4.0 users who convert their business processes into automated BPM processes using the application and then macro-manage those processes. For this project, we made the following assumptions about the audience:

- Users range in computer skill from basic to expert.
- Users are familiar and comfortable working in the Microsoft Windows environment.
• Users will use the application and explore the online documentation without receiving training about the application and its use.
• For a few of the tasks in the application, users will need special assistance from software developers and from database administrators.

**THE MEDIUM**

The documentation was to be a part of the application, would be available as online help text, and was to be developed using Macromedia RoboHelp. If time allowed for their development, online demonstrations should complement the online help text, and demonstrate the fundamentals of BPM Suite 4.0. The demonstrations should be easily viewed within the online text and should be in a format—preferably Flash files—that could be easily compressed and viewed by all users.

Clearly defining the scope, the purpose, the audience, and the medium is a strategy I’ve employed in my past documentation projects in the MTSC program and at my assistantship with IT Services. This practice has successfully helped me to clearly define documentation needs or business challenges in those environments and was useful at Bluespring as well.

**MY ROLE IN DOCUMENTING BPM SUITE 4.0**

Our main objective while documenting BPM Suite 4.0 was to deliver just-in-time assistance to users, thus reducing the number of technical assistance and support calls to the Bluespring TAC. To meet this objective, I completed the following tasks:

• Researched BPM Suite 4.0.
• Wrote the online help topics.
• Submitted the topics for SME review.
• Edited the online help topics.
• Developed the online demonstrations.
• Tested the online demonstrations.

**RESEARCHING THE BPM SUITE 4.0**
To research BPM Suite 4.0, I attended a training session, reviewed older documentation for its predecessor the Priority CS 3.1 application, and interviewed the Training Specialist, Angela Shealy.

On the first day of my internship, Kayte and Angela conducted a training session to introduce Chris and me to the first generation of the BPM Suite 4.0—Priority CS. The Priority CS application was then being streamlined and renamed BPM Suite 4.0. The training session was held in Bluespring’s main conference room and lasted approximately one hour. Kayte and Angela introduced us to the two modules we were to document – the Configuration Management and the Process/Workflow Management modules.

Attending the training session helped me to understand the application and begin to assess the needs of the users. I was grateful that Kayte and Angela volunteered to conduct the training sessions since the sessions helped me understand the business situations in which the application could add value and the complex logic the application uses. The most important lesson I learned from this session was that Bluespring’s customers were accustomed to seeing tasks grouped into modules and that the documentation I produced should follow the application’s groupings to provide a consistent look and feel.

After the training session, I still struggled with the conceptual model the application uses. Mostly I struggled with the need to automate tasks that I felt were simple and routine, tasks like sending e-mail and assigning tasks based on available resources. After reviewing the older documentation, I better understood that automating even the simple and routine tasks was essential to having workflows adequately represent entire business processes. As the SMEs explained to me, reducing human intervention within a workflow increases the speed at which tasks are performed and produces more predictable outcomes.

I spent the majority of my first week reviewing previous documentation for the Priority CS 3.1 application. This documentation included incomplete Priority CS 3.1 online help, portions of which are included in Appendix C. Additionally, I also reviewed a copy of the Priority CS 3.1 training materials consisting of which include presentation slides and training labs. I have included samples of these in Appendix D. Most of this documentation was outdated and incomplete, but it was still very helpful because it
helped me grasp some of the conceptual information I struggled with after the training session and helped me find answers to questions that still lingered. After reviewing the older documentation, I decided to meet with Angela again to perform a preliminary audience analysis.

I met informally with Angela by the end of the first week of the internship to conduct an audience analysis. As the Training Specialist, Angela has the most experience with the application’s users since she provides on-site training to Bluespring’s software users at customer requests. No potential audience members were available for interview at the time I was writing the documentation (no licenses for the product had been sold), so Angela was the only person able to discuss the needs of the audience with me. To better analyze the audience, I asked Angela the questions below. Summaries of her responses follow the questions:

- **Who will use the documentation we produce?**
  The people who will use this software will build the workflows that other employees will use. These users will have knowledge of their business processes, but not necessarily expert knowledge of computer systems and programming.

- **Where will users use the application and the online help?**
  Users will typically use the application and documentation at their workstations.

- **What are some of the comments users have made about the online help and the training materials in the past?**
  The main thing the users have said is that they did not understand why they were completing tasks; therefore, it is important that the new documentation clearly explain the rationale for performing software tasks.

I used this information to inform the choices I would make as I wrote the online help. For instance, since users would have varying computer skill levels, I decided that About sections would be necessary for each module. In each About section, I would explain concepts at a high level and urge users to go on to individual sections for more detailed explanations. Using this strategy, the online help could inform audience members with either basic or expert computer skills since they had a choice between high-level explanations and more detailed explanations.
After performing the audience analysis, Chris and I established conventions for writing the online help, see Appendix E for the style guide we created.

**WRITING THE ONLINE HELP TOPICS**

The style guide and good practices document for the project reflected what we learned from the audience analysis. For example, our audience analysis assumed that BPM Suite 4.0 users would have varying computer skill levels and that some would be interested only in reading the high-level conceptual information while others would need to have concepts defined in detail and have their own roles in the workflow clearly defined. To accommodate the needs of most in our audience, we used several different heading levels anticipating that we would have to “drill down” from the very general to the very specific.

Using the preliminary task analysis and using the style guide, we began writing the topics in the following format:

- **Writing an About section** – in these sections, we described the category of tasks and outlined and linked the tasks that were in this category.
- **Writing the individual tasks instructions** – for each task, we briefly described the task’s functionality and use, and then wrote step-by-steps instructions for that task.
- **Linking related topics** – using the Related Topics category, we linked other topics that directly related to the current topic.

Appendix F reproduces the module About Advanced Tasks, which contains the following topics: Adding Logical Expressions Rules Tasks, Editing Logical Expressions, More about Logical Expression Rules, and Tips for Writing Logical Expressions. These examples show how our audience analysis helped to inform the way we wrote the help topics. The About Advanced tasks topic introduces very high-level concepts and users can find more details about Logical Expressions Rule tasks. They can also read about specifics such as editing Logical Expression Rules. Our audience analysis also indicated that users might need the help in the middle of task for just-in-time assistance. Therefore, I also included a Tips for Writing Logical Expression section to offer users tips based on some of the frequently asked questions received by the SMEs in the Bluespring TAC. After writing the topics within each module, we submitted them for SME review.
SUBMITTING THE TOPICS FOR SME REVIEW

To submit the topics for technical review, we printed each topic and hand-delivered it to the SMEs. There were primarily four SMEs for this project, virtually all of the Bluespring developers. To ensure that we did not receive all the SME reviews at the same time, we arranged staggered deadlines so that we could discuss comments and suggestions with each developer when they returned the topics to us. Examples of the request for the SME reviews we submitted to the Director of Software Development appear in Appendix G. Since there was no original SME review form, I designed the documents in Appendix G so that the SMEs would have some guidance as they reviewed the online help. I also wanted to ask some very specific questions, about whether the help accurately represented the tasks users must complete. Since I knew the developers were working on other projects, I estimated the time it would take them to review each topic and included that on the review sheet as well. In the request, we identified each module, the SME who should review it, and the time it would take to review each module. I wanted to be very sensitive to the fact that they were working on other projects and they needed to know approximately how much time they needed to review each module. Along with a review sheet (also in Appendix G), we attached printed copies of the modules each SME was to review.

Several SMEs had expressed interest in helping me produce the online help because they believed that documentation was an integral part of the BPM Suite 4.0. Later as I sat down to discuss their feedback with them I began to see a shift in their views. They clearly saw documentation as separate from the application, and the documentation process as a process that trailed the development process. Therefore, when changes were made to BPM Suite 4.0, I was often not informed and my documentation quickly became inaccurate and outdated. Although this was frustrating, I understood that this is a common approach to software documentation—especially when documenting software that is unstable because it is still in development.

Three of the four SMEs requested more system-centered documentations that described the application’s many functions, instead of its benefits to users. Referring to my initial audience analysis, I stood as the users’ advocate and reminded the SMEs that the audience was likely to be more concerned about how the application benefited them
than about the many great features of the application. After several discussions and gentle persuasion from my project manager, the SMEs eventually agreed that the documentation should be guided by a user-centered approach.

**Editing the Online Help Topics**

After receiving the SME review documents, I performed two types of editing. The first was a technical review based on the feedback from the SMEs. I edited text and tested the instructions to ensure they were technically accurate with the latest version of the software. During the week we sent the topics over for SME review, we learned that the application was to undergo more major changes. These changes included changing key terminology, revising naming conventions, and eliminating tasks we had previously documented. One specific change renamed the “canvas” on which the business workflows were built—Workflow Designer became Process Designer. Thus every reference to the word “workflow” had to be replaced with the word “process”. Although these changes added to our editing tasks, we were able to make them in a single day of updating work.

A subsequent copy edit checked for consistency, grammatical accuracy, brevity, and audience appropriateness. One of the major comments we received from the SMEs was that the basic concepts needed to be explained more thoroughly. Although we agreed with the suggestion, we were concerned about the readability of dense paragraphs in an online help system, especially since our audience analysis indicated that users would have varying technical knowledge. To compromise, I suggested using RoboDemo movies to visually explain more complex concepts and inserting the movies as demonstrations into the online help. Users could then choose to view the demonstrations if they wanted topics explained further, or they could continue with the instructions for completing tasks. The SMEs agreed with this idea, and I began developing the online demonstrations.

**Developing the Online Demonstrations**

The online demonstrations were easy to develop because the SMEs had provided good feedback on the online help. For example, they gave me some of the frequently asked questions they had received from customers in the Bluespring TAC.
Appendix H outlines the modules for which I planned to create RoboDemo movies. I created all but one of the topics in the plan. After reviewing the plan, Kayte and Angela decided one of the demonstrations was too complex for the online help project and instead made it the basis of the e-learning and training sessions.

One of the frequently asked questions received from the customers was “How do I link tasks?” I addressed this question using an online demonstration of the instances when the different types of links are used. Appendix I shows a section of one of the movies I created for the Linking Tasks module. In it I outline how users can create logical branching links on the Process Designer canvas to link the tasks that create their processes. Accurately modeling tasks and the links that connect them is the essence of BPM Suite 4.0 and I felt that it was vital that users understand this.

**TESTING THE ONLINE DEMONSTRATIONS**

After developing the online demonstrations, I asked Chris to review and test them to see that they accurately modeled the concepts they were meant to convey. Using RoboDemo’s functionality, I converted the movies to Flash file format and sent them to him via e-mail so that he could easily view them using a Flash player. After reviewing each of the seven movies, he offered feedback about their content and design. His feedback helped me to see when the demonstrations worked well and when they didn’t. At the highest level, he thought the demonstrations worked well, but he suggested that we increase the time between each of them so that the audience could read and absorb the concepts presented. We discussed his feedback, and I made changes to the demonstrations. I then asked Angela to review the movies. She also offered feedback—the demonstrations were accurate and helpful. After both reviews, I was confident that I had developed demonstrations that user would find useful and usable—they accurately demonstrated concepts and processes and addressed the users’ questions about the context of the concepts and processes.

During the final week of my internship, I handed the project off to Kayte. She reviewed the online help system and demonstrations and declared the project a success. Since the application was still under development, the documentation and demonstration I prepared would have to be updated and later added to the CD with the application.
Looking back, I believe the project was a success. I attribute the success of the project to thorough task and audience analysis, astute project planning and care with meeting our deadlines. Keeping the users in mind while I documented BPM Suite 4.0 helped me meet this project’s main objective – to deliver just-in-time assistance to BPM Suite 4.0 users. In the following chapter, I explain how I applied Paul Anderson’s problem solving model to this project.
CHAPTER 4: AN ANALYSIS OF THE INTERNSHIP

In this chapter, I reflect on my activities during my internship at Bluespring, beginning with a discussion of how I used Paul Anderson’s problem solving model to plan the documentation for my major project. I end with my reflection on what I learned about technical communication and the corporate culture in which I would like to work. Although I was involved with four main activities during my internship, my main focus was to document BPM Suite 4.0. To address this documentation challenge, I used Paul Anderson’s problem solving model.

USING PAUL ANDERSON’S PROBLEM-SOLVING MODEL

Paul Anderson’s problem-solving model was the major planning tool I used for documenting BPM Suite 4.0. The model proposes five phases for developing technical communications:

1. Define Problem
2. Design Solution
3. Test Solution
4. Implement Solution
5. Evaluate Solution

In completing this project, I used the first four phases of Anderson’s model to help me plan and develop documentation. In the first phase – Define Problem, I analyzed the situation/context, the purpose, the audience, and the medium for the documentation, and I set objectives for the project—all of which I’ve outlined in Chapter 3. During the second phase – Design Solution, I gathered information, created a style guide and good practices document, and outlined the development plan for the RoboDemo movies. When this phase was complete, I had the first draft of the online help ready for technical review. During the third phase – Test Solution, I submitted the first draft to the software developers for SME review, and I compiled and discussed their feedback with them before I implemented any changes. Throughout, my goal was to be the user advocate—to analyze the feedback from the SMEs and only make changes that would benefit the users, based on the information I gathered from the audience analysis.
Before I could get to the fourth phase – Implement Solution, the application changed several times. Whether it was a functionality change (the application performed differently) or a more cosmetic change (button and icons were renamed, but functions stayed the same), my documentation also had to change. These changes meant I often had to rewrite and retest the instructions and the online demonstrations to ensure both correctly depicted the users’ experience. As is the nature of software development, I rewrote and retested the documentation several more times before I was able to implement the solution.

Because the transition from the third phase to the fourth phase was so dependent on whether the application was stable and ready for release, I have just one criticism of Anderson’s problem-solving model.

**ONE LIMITATION OF THE PROBLEM-SOLVING MODEL**

My criticism of the heuristic is that it is not iterative enough for software documentation planning. During my internship, the application—BPM Suite 4.0, changed several times. After the initial Design Solution and Test Solution phases, I would insert another phase – Reevaluate Solution before Anderson’s fourth phase Implement Solution to make this model more iterative. I found that when software updates were communicated to me, I often had to reevaluate my documentation to ensure it was still accurate. In some cases, this reevaluation meant rewriting modules based on the application’s new functionality.

Although I have this criticism of the problem-solving model, I still believe it is a good heuristic for defining a stable documentation challenge, designing possible solutions to meet that challenge, and testing those solutions.

**REFLECTING ON THE INTERNSHIP AND MY EXPERIENCE AT BLUESPRING**

My experience at Bluespring cemented my views on the type of corporate culture in which I would like to work. Since Bluespring was a young, start-up company trying to find its niche in the BPM software industry, its culture was very volatile. When I worked there, I often had the sense that Bluespring’s board of directors dictated how Bluespring should do business with little buy-in from the employees. I sensed a feeling of exclusion
from the company-wide meetings where the CEO and senior managers simply reported directives from the board of directors and only solicited feedback about whether we understood what the new directives meant for day-to-day operations. That feeling of being excluded from the decision-making process often affected employee morale in general, and affected my motivation to go beyond what was expected of me. My motivation diminished more when I was told about the layoffs, and the layoffs strengthened my desire to work for companies that value employee buy-in.

In my opinion, many of Bluespring’s business practices seem haphazard and unorganized. While I worked there, none of they best practices, success, or failures were captured with the expressed purpose of reusing the best practices and improving on the failures. Instead of developing a solid application that could be modified to each customer’s need, the shareholders at Bluespring were more concerned with customizing the application to a customer’s needs and then working out the software bugs once they sold the software. In the future, I know now that I only want to work with companies that have well-documented design and development workflows, standardized processes, and well-developed best practices. I think I would rather work for an organization that documents both its failure and successes and promotes reuse of best practices by first offering a well-built application that can be customized rather than selling a flawed product.

My experience at Bluespring also heightened my desire to be more involved with developing the technology users see. To truly be a users’ advocate and promote user-centered design, I believe that I should be more involved in the early stages of application development. Ideally, I would like to work with application design team to promote usability issues, such as consistent and meaningful naming conventions and providing tools like tool tips that provide text explanation when users mouse over icons on the screen. In the case of Bluespring’s BPM Suite 4.0, I felt that I had no opportunity to offer feedback about the application and that I was not a part of its development. As I mentioned in the previous chapter, the software developers, and later the managers, viewed documentation as an addition to the application and not as a crucial part of the application. For my major project, I felt that I was matching the documentation to the application, regardless of whether I believed the application was user-centered. For
example, when I suggested creating a style sheet for naming icons for consistency, my suggestion was ignored. Having consistent naming conventions was considered a quality assurance or documentation issue that did not affect functionality, and sadly only issues that directly affected functionality were immediately or eventually addressed. It was also challenging being the users’ advocate when the SMEs proposed that I write system-centered documentation. In this situation, it was difficult to reach a compromise that we knew users would appreciate (based on the audience analysis) and that the SMEs would also applaud. It was also disheartening to be the users’ advocate and to know that my suggestions for improving the quality and usability were not a priority. In the future, I hope the software developers and managers at Bluespring see some of the many benefits of considering the audience when they develop software—benefits like increased customer loyalty and increased profitability.

With reference to the tools I used during the internship, the MTSC program and my graduate assistantship at Miami’s IT Services prepared me well for the internship experience. For my major project, I used a help text authoring tool and a demonstration and simulation tool. While in the MTSC program and at my assistantship at IT Services, I used both those tools on various projects. When I began my internship, I was excited to learn that the same tools I was exposed to in the classroom were being used in the industry. I also had a great experience working with former MTSC students and a BATSC student because we shared the same understanding of and zeal for software documentation. Together, we often used many of the preparatory strategies we learned in the MTSC program, like researching topics to clarify the business challenge, brainstorming ideas to address the documentation need, and applying collaborative problem-solving skills to reach a viable solution to address the documentation need.

My major project was not without challenges. Chief among them was managing the project scope. I was grateful that my project manager helped control the scope of the project by monitoring and recording my progress. Each week we communicated my accomplishments, goals for the next week, and any concerns (and their proposed solutions) in a weekly report, see Appendix J for a sample. Creating the weekly report helped me to identify some of the challenges with the project before they became unmanageable, and the report also helped me to suggest possible solutions to overcome
those challenges. Whenever possible, my project manager responded to the challenges I raised within two business days, either confirming my proposed resolution or suggested other ways I could overcome the challenges I described.

Overall, I was grateful for the opportunity to add value in a company like Bluespring Software. I feel that its customers appreciate the work of technical communicators and that the company has potential for growth for technical communicators, especially in the areas of technical training, project management, and human resource management. In a market where technical communicators are also required to be learning strategists, project managers, and Web site developers, I believe the MTSC program, the assistantship experience, and the internship experience adequately equip students of technical communication with real-world skills to meet today’s business challenges.
CINCINNATI, OHIO (May 3rd, 2004)

Bluespring Software announces general availability of its BPM Suite 4.0, best-of-breed “pure play” in the business process management (BPM) space. The core execution engine is a web services oriented architecture built on a .NET framework, ensuring maximum scalability and stability while offering ease in integration with existing infrastructures. Highlights include:

Ease of Integration

Web services-based API’s and XML data representation provide seamless integration with client applications and infrastructure. Automatic creation of VB.NET, C#, VB6 or Java Proxy code minimizes integration development efforts.

Efficient Human Worker Collaboration

Manual tasks are efficiently orchestrated into the process, factoring in worker role, geography, workload and business calendar. Task notification occurs via email, SMS or IM with a hyperlink directing workers to the task user interface. Integration with common Microsoft desktop tools Excel, Word and InfoPath ensures worker acceptance and yields maximum productivity.

Optimized People and Systems Interaction

Tight integration with Microsoft desktop tools provides smooth data flow from people-to-people and people-to-systems. A generic database reader and writer, along with a web services component, enhance system-to-system integrations.

Incorporation of Mobile Devices

Connected or disconnected Windows Pocket PCs allow workers to interact with their task lists and submit real-time or batch updates via WiFi, GPRS or Bluetooth.

“BPM Suite 4.0 is an enterprise-grade BPM offering that can handle complex processes and efficiently manage high levels of human interaction, which is the grail for effective business process management,” said Rob Daly, President and CEO of Bluespring Software.
**Task-Based AMC Workflow Help**

**Introduction/Executive**

**Summary**

Basics
- Using drop-down lists
- Right clicking
- Minimizing and maximizing windows
- Closing windows
- Checking and unchecking boxes
- Understanding the tree structure
- Expanding
- Using scroll bars
- Dragging and dropping

Understanding Workflows
- How workflows work
- Definitions
- Workflow
- Service order
- Work order
- Task
- Turning a business process into a workflow
- Overview of workflow creation
- Defining your requirements
- Creating a workflow
- Launching the workflow

Accessing AMC
- From the desktop
- From the Programs list

Logging In
- Entering your user name
- Entering your password
- Defining your login settings
- Selecting a user class

Understanding the
Accessing workflow
<table>
<thead>
<tr>
<th>Workflow Management screen</th>
<th>Good place for a RoboDemo Overview Workflow management toolbar Table of icons and button descriptions Toolbox Canvas Debug tab Design tab Status bar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creating a New workflow</td>
<td>Definition</td>
</tr>
<tr>
<td>Opening an Existing Workflow</td>
<td></td>
</tr>
<tr>
<td>Searching for a Workflow</td>
<td>Searching for a workflow by Flow Searching for a workflow by Task Entering search criteria</td>
</tr>
<tr>
<td>Viewing/Managing SOs and Workflows</td>
<td>Viewing by service order Viewing by workflow Refreshing the screen Narrowing your search to from date and to date Selecting a status Showing cancelled service orders and flows Showing empty SOs Scrolling backward and forward by 200 records Going to an account Jumping to your task list Showing the Color Key Accessing workflow properties</td>
</tr>
</tbody>
</table>
Viewing content information
Viewing flow definition
Viewing system related information
Accessing task properties
Viewing task definition
Viewing system relation information
Accessing workflow information

Viewing/Managing Your Workflow Task List
- Filtering tasks
- Sorting tasks
- Accessing your assigned tasks
- Viewing your assigned tasks
- Refreshing the screen
- Completing a task
- Saving your work
- Accessing unassigned tasks
- Viewing unassigned tasks
- Refreshing the screen
- Taking a task
- Saving your work

Editing Workflow Scripts
- Creating a new workflow script
- Removing a workflow script
- Exporting a workflow script to a file
- Importing a script from a file
- Adding parameters
- Adding parameter values
- Editing a script
- Testing a script
- Viewing available workflow scripts

Editing Logical Expressions
- Showing all expressions

Adding parameter values
- Editing a script
- Testing a script
- Viewing available workflow scripts
Loading expressions
Saving expressions
Since tasks cannot exist alone, they must be linked together. There are two types of task links: the Standard Link and the Logical Branching Link. Your requirements will dictate which link should be used for each task. To link tasks, follow the procedures below.

Select by clicking the task icon that begins the workflow. To indicate that the task is selected, selection handles appear.

Click the center handle and drag the mouse over the task icon you are linking to.

Release the mouse. A line is drawn connecting the two tasks.

Right-click (or double-click) on the line to access the Link Properties window.

Note: Link Properties can also be accessed by double-clicking on the line or selecting View Link.

Properties can also be accessed by double-clicking on the line or selecting View Link.

By default, the link will be a solid blue line with the arrow pointing from the initiating task toward the following task.

Setting up Standard Link Properties

Standard Link properties define a simple link with no branching criteria. Standard Links must be executed in order for a flow to complete.
Field

Link Type

Description

Default is *Standard Link* meaning that the link must be executed in order to complete the flow.

Critical Path?

Options: Yes or No.

*Yes* = link displays with a dotted line; task must be completed in order for the workflow to complete successfully.

*No* = the link will be displayed with a solid line (default).

Wait on Task

Used only when more than one link is going into a **Connection Node**. When there is only one link going into the task, it will not matter whether *Yes* or *No* is selected.

Options: Yes or No.

*Yes* = Standard link must wait for the previous task to complete before launching the next task.
Setting Up Link Properties for Logical Branching

Logical Branching Links will branch based on expressions that you build into the properties. Logical Branch Expression link properties define which direction a flow should follow based on the data item values passed into the flow.

To set up link branching, follow the procedures below.

1. On the Data Item Management toolbar, click View Link Properties. The Link Properties window appears with the Standard Link appearing in the Link Type text box.

2. To set up the link for logical branching, in the Link Type text box, click the drop-down menu.

3. Choose Logical Branch Expression. The Link Properties window changes. Edit the window.
Field
Link Type
Critical Path?
Wait on Task
Available Data Items

Description
*Select Logical Branch Expression* from the drop down list. The additional properties will be displayed.

Select *Yes* or *No* from the drop down list.
*Yes* - the link will be displayed with a dotted line, to show tasks that must be completed in order for the flow to complete successfully. The critical path is also used in calculating the *Average Flow Completion Time*.

*No* - the link will be displayed with a solid line, the default.
Select *Yes* or *No* from the drop down list. If you select *No*, the flow will not wait for the branched tasks to complete before continuing the flow.
Note: The only time *Wait on Task* should be set to *No* is when two or more logical branches are recombining at a connection node.

Select data item from the drop down list. You must add data items through the **Data Item Management** window before setting up a *Logical Branch Expression*.

Expression Builder Toolbar

Use these buttons to build your expression.
Module 15:
Advanced Tasks

- Logical Expression Rules
- Scripting Task
- Web Service Invoker
Advanced Tasks, like Standard Tasks, are not specifically assigned to any particular user. They are tasks performed by the system and require no human interaction to execute.

Advanced Tasks either solve complex logical expressions or execute scripts that process Business Rules.
**OBJECTIVES**

**Module Objectives**

The objectives of this module are to:

- understand what Logical Expression Rules tasks do and what their properties are
- and understand what Scripting Tasks do and what their properties are.

By the end of this module, you should be able to:

- configure a Logical Expression Rules task
- and configure a Scripting Task with a pre-written script.

This module will introduce you to Logical Expression Rules, Scripting Tasks, and the Web Service Invoker and explain how each is configured.
The Logical Expression Rules tool allows the user to collect data items’ values, evaluate the chosen values against a defined table, and populate a specific data value into another data item. Subsequent business logic can then use the newly-populated data values, replacing complicated, code-driven logic. This capability is very powerful, as it can replace complicated, code-driven logic within a scripted function. This is a table-driven, value-based capability, providing flexibility and reusable logic.
The Logical Expression Rules tab provides the user with the opportunity to configure the data items used in the value comparison, along with an AND or OR condition required for data comparison.

Two separate radio button groups allow the user to control the Logical Expression. The first radio button group (on the left) specifies when data items for value comparison, called keys, are set. These keys can either be assigned at the workflow design time or at run time, based on the requirements established by business rules.

The second radio button group specifies the condition required for the value comparison – whether all values need to be met (an
AND condition), or whether only one of the values need to be met (an OR condition).

The Use these Keys to choose Set of Rules group box contains three drop-down boxes to define data items for the logical comparison. After selecting the appropriate keys for comparison, you can then view or input the values used for evaluation by clicking the View/Input Rules button, which will open a window, shown on the next slide, to configure the data values used for comparison.

The Write the output of the Logical Expression to the following DataItems group box contains two drop-down boxes to define where the output of the task will be stored. The DataItem to Hold the Expression Result drop-down box allows the user to select which data item will hold the “true” or “false” value associated with the result of the comparison. The DataItem to Hold the Expression Data drop-down box lets the user select which data item will hold the value associated with the “true” or “false” outcomes of the comparison.
Logical Expression Rules (continued)

Logical Expressions Selection Window
- There are two options for viewing input data in the window:
  - it can be viewed by all data combinations, or
  - data combinations can be filtered by a particular key.
- The user can add, edit, and delete all conditions directly from this window.

Within the Logical Expressions window, two options are available for viewing input data. The user can either view all data combinations expressed in rows of data, or filter data combinations by a particular key. The **Key 1**, **Key 2**, and **Key 3** boxes are auto-populated from the previous window. The user can also load the data value combinations already stored using the **Load Expressions** button or save new data combinations by clicking the **Save Expressions** button.

To add new conditions, scroll to the bottom of the Logical Expressions array, place the cursor in the first column of the array, and begin to enter data. Use the tab key to navigate across the row, moving from column to column. Upon arriving at the Logical Expression column, an ellipse button will appear. This
button will open the **Logical Expression** window, where these expressions can be edited.

**DEMONSTRATION**

Now you will view a demonstration of the Logical Expression Rules task.
The Scripting Task allows the user to write custom-coded logic (in either VBScript or JavaScript) to support a particular task within a workflow. Scripts must be written in accordance with the naming and access conventions of the workflow engine. To effectively create scripts for workflows, you must have prior knowledge of VB Script or JavaScript, as well as experience manipulating the Priority CS Object Model.
The Scripting Task tab contains four specific components that impact the use and creation of workflow scripts: the scripting toolbar, the **Currently Assigned Script** field, the **Script Parameters for the Currently Assigned Script** array, and the **Available Workflow Scripts** list.

The scripting toolbar contains five buttons which allow the user to (in order of appearance on the toolbar):
- Create new scripts,
- Delete the currently selected script,
- Change currently selected script’s properties, such as name, path, and category,
- Assign the currently selected script to the current task,
- Edit the currently assigned script.
If the currently assigned script requires parameters, they will be listed in the parameters section of the screen. Simply enter the values required.

**SCRIPTING TASK (CONTINUED)**

The file management toolbar has three main capabilities: Save, Export Script to File, and Import Script from File.

The first tool on the Edit Tools toolbar, **Script Editor Options**, represented by a crossed hammer and wrench, allows the user to adjust settings found the script editor, such as font size, color, and indent width. This impacts only the visual appearance of the script editor; all other functionality remains the same.
The second tool contains three buttons for **Find**, **Repeat find**, and **Replace**, which all behave as in most text editors, such as Windows Notepad.

An icon of a hand holding a sheet of paper represents the third tool, a listing of valid workflow and scripting Objects. This tool allows the user to view valid Objects, entities, and functions while writing code, depending upon any predecessor scripting language that has been written.

The fourth tool, an **Entity Definer**, specifies the Object type, entity, or function the user has selected. This tool is represented by a caption with an arrow pointing from it and allows the user to verify the definition of a particular line of code.

The next two tools allow the user to flag lines of code as either breakpoints or bookmarks. A hand and a flag represent them, respectively.

Finally, the **Script Editor** itself is a Windows Notepad-like area where the script is actually edited.
Demonstration

Now you will view a demonstration of the Scripting task.
The Web Service Invoker Task allows a web service to be launched synchronously from a flow. The Web Service Invoker allows a workflow to use information generated from an external application. The applications, called web services, can be called from the web to do any necessary. Information returned from web services can then be put into data items and used later in the workflow.
The configuration of the Web Service Invoker Task takes three main steps. First, enter the URL of the web service the task is supposed to use and click on the **Read Def** button. Next, select which method of the web service to use (if applicable). Finally, select data items to use for the service’s parameters.

The “Service Properties” section displays additional parameters that may or may not affect the web service.
SUMMARY

In this module, you learned:
- what a Logical Expression Rules task is and how to configure one
- and what a Scripting Task is used for and how one is configured with a previously-written script.

Review Questions:
In the Logical Expression Rules task setup, what is the difference between the AND and the OR condition?
What is the difference between “Design Time” and “Run Time?”
What scripting languages are available to write workflow scripts in?
How can a script’s properties be viewed and edited?
APPENDIX E – STYLE SHEET AND GOOD PRACTICES DOCUMENT
## Style Guide & Good Practices Document

<table>
<thead>
<tr>
<th>Date</th>
<th>Change made</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>March 31, 2004</td>
<td>Original draft</td>
<td>Chris Schapman</td>
</tr>
<tr>
<td>April 5, 2004</td>
<td>Added “e-mail” and “time zone”</td>
<td>Chris Schapman</td>
</tr>
<tr>
<td>April 12, 2004</td>
<td>Added Naming Conventions section</td>
<td>Stacey Scott</td>
</tr>
<tr>
<td>April 13, 2004</td>
<td>Updated all sections to say “Process”, not “workflow”</td>
<td>Kayte Rudolph</td>
</tr>
</tbody>
</table>
Above

Only use “above” when directing the user to a spot on the UI
Do not use “above” when referring to information on the Help topic or TOC

Below

Only use “below” when directing the user to a spot on the UI
Do not use “below” when referring to information on the Help topic, instead use “following”

Cancel

Usually speaking, when writing a task, do not give the user the option to click Cancel.

Example: After completing the information, either click OK or Cancel.

Check box

Click

Use “click” and not “click on”
Use “click” with check boxes, buttons, lists, drop-down lists

DataItem

Use “DataItem” as one word, not “Data Item” as two words.
Capitalize the ‘D’ and ‘I’ in DataItem.

Drop-down list

Use “drop-down list” when a user has to click a field or down arrow to see the contents of the field.
Example: Click the Status drop-down list, and then select Complete.

Ellipses

Do not use ellipses even if it appears on the UI.
If a button only has ellipses on it, call it the Ellipses button.

E-mail or e-mail

Do not use “email”

Example

Write example text on line following example in bold with a colon.

Future tense

Do not use future tense, unless referring to an action the user might do some other day.

Highlight

Use “select” when the user has to select text, do not say “highlight”

Icon

Do not use “icon” when referring to a clickable image.
Instead, use “button.”

Example: Click the Grid button.

Important:

Write important text on line following important in bold with a
colon

List  A “list” refers to an array of elements already displayed on the screen from which the user can make a selection.

**Example:** From the list of DataItems, select **UserID**

Message box  Use “message box” when referring to a confirmation type of window that appears, which asks the user for a Yes/No, OK/Cancel, etc. type of response.

Do not use “window” to refer to a “message box”

Note:  Write note text on line following note in bold with a colon

Optional:  Write optional text on the same line as optional in bold with a colon

Pane

Radio button

Select  Use “select” when prompting a user to make a selection from a list or drop-down list.

**Note:** MS Manual of Style says go “click,” but we decided to go with “select”

Use “select” when prompting a user to highlight text. Do not say “highlight.”

System  Use “System Administrator” and not “System Configurator”

Administrator

System  Do not use “System Configurator.” Instead, use “System Configurator Administrator”.

Configurator

Tables  Preferred width: 90%

First row: Bold White text are to be 90% and in line with text above table.

Left align table with text under a step or bullet; otherwise, indent table once.

No bold in left column unless talking about a specific field, button, etc.

Time zone  Time zone is two words.

Tip:  Write tip text on line following tip in bold with a colon
<table>
<thead>
<tr>
<th>Window</th>
<th>Use “window” when referring to anything that is not a “message box” See Message box for more details.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Designer canvas</td>
<td>Use “on the <strong>Process Designer</strong> canvas…” and not “in the <strong>Process Designer</strong> canvas…” Bold Process Designer, but not canvas.</td>
</tr>
<tr>
<td>Process Designer toolbar</td>
<td>Bold Process Designer, but not “toolbar” Use “on” when referring to objects in the toolbar. <strong>Example:</strong> On the <strong>Process Designer</strong> toolbar, click the <strong>Show Grid</strong> button</td>
</tr>
<tr>
<td>Process Designer toolbox</td>
<td>Bold Process Designer, but not “toolbox” Use “from” when referring to object in the toolbox. <strong>Example:</strong> From the <strong>Process Designer</strong> toolbox, click the <strong>Nested Flow</strong> task</td>
</tr>
</tbody>
</table>
**GOOD PRACTICES**

The following lists good practices for documenting in RoboHelp X5.

When you change the title of the topic, make sure you change the **File Name** and **Topic Title** on the **Topic Properties** window.

When multiple people are editing a project, remember to **Check in** files often and **Get latest** often, so others can see the changes you’ve made and you can see the changes other have made.

Make sure the user feels in control of Priority CS. The user acts, not the program.

Assume all fields are necessary, unless you explicitly tell the user it is optional.

**NAMING CONVENTIONS**

The following lists the naming conventions used for documenting BPM Suite 4.0.

**Windows** – when referring to a window by its name, always use bold and title case (**Name in Title Case** window) for the window name.
ABOUT ADVANCED TASKS

Advanced tasks are not specifically assigned to any particular user. They can be performed by the system and thus require no human intervention. Advanced tasks either solve complex logical expressions or execute scripts for processing business rules.

ADDING ADVANCED TASKS

- Adding Logical Expression Rules
- Adding scripting tasks
- Adding process scripts
- Adding Web Service Invokers
**ADDING LOGICAL EXPRESSION RULES TASKS**

Using the Logical Expressions Rules task and your business logic, you can collect DataItem values, evaluate selected values against a set of rules you define, and populate the new data values into new DataItems.

Read more about Logical Expression Rule tasks.

**TO ADD A LOGICAL EXPRESSION RULES TASK**

From the Process Designer toolbox, click the Logical Expressions Rules task. The task appears on the Process Designer canvas.

Double-click the task. The Properties for Task window opens. The Logical Expressions Rules task properties window contains five tabs: General, Rollback Information, Logical Expressions Rules, Display, and Related Tasks.

Complete the General task property tabs.

Click the Logical Expressions Rules tab. Determine when you will set the DataItems you will use as keys:

- Setting keys at Design time.
- Setting keys at Run time.

Determine whether the expression in each group will use AND or OR logic:

- Click the AND radio button if all the expressions need to be true.
- Click the OR radio button if only one of the expressions needs to be true.

Note: The expression will only return a True or False result.

Click the View/Input Rules button to view or input your logical expression rules. The Logical Expressions window opens.

Create your logical expressions.

To reuse the default DataItemValue, leave the Do Not Reuse the Default DataItemValue check box empty.

From the DataItem to Hold the Expression Result drop-down list, select the DataItem that will hold the actual true or false value the expression results return.

From the DataItem to Hold the Expression Data drop-down list, select the DataItem that will hold the data the expression results return.

From the Error Recovery options, select one of the following for this task:

- Mark task as failed and continue process implies that an error is rather unimportant to the business process.
Mark task as failed, raise With Exception Flag and continue process provides low-level notification that an error has occurred but continues the process.

Mark task as failed, raise With Exception Flag and stop process provides low-level notification that an error has occurred and stops the process.

Rollback and cancel the process and all subprocesses encountered is the most severe and instructs the process and any subprocesses to come to an immediate halt. This strategy is extreme and immediately impacts the business processes. Selecting this option implies that there is a strategy to resolve the processing error as well as dedicated resources to ensure that it is resolved quickly.

Click OK. The Properties for Task window closes.

Related topics
- About advanced tasks
- Editing Logical Expressions
- Tips for writing logical expressions
EDITING LOGICAL EXPRESSIONS

You can edit the logical expressions that you use to collect data and drive your business logic.

Read more about Logical Expression Rule tasks.

TO EDIT A LOGICAL EXPRESSION

From the process Mgmt menu, select Edit Logical Expressions. The Logical Expressions window opens.

Filter the expression you want to see:

To see all expression, select the Show All radio button.

To see only selected expressions, select the Filter By Key radio button and select the keys to use as filters.

Click the Load Expression button. The expressions load.

Click the cells you want to edit.

For cells with blinking cursors, type directly into the cell.

For the Logical Expression cell, click the ellipse to open the Logical Expression window. In this window, you can directly edit the logical expression. Tip for writing logical expressions.

When you have finished editing the logical expression, click OK. The Logical Expression window closes.

Click the Save Expression button to save your edits.

Related topics
Adding Logical Expression Rules tasks | Tips for writing logical expressions
MORE ABOUT LOGICAL EXPRESSION RULES

Using the Logical Expressions Rules task and your business logic, you can collect DataItem values, evaluate selected values against a table you define, and populate the new data values into new DataItems.

You can use the new DataItems you create in other logical processes. Using Logical Expression Rules, you have the flexibility to make table-driven, value-based decisions and the flexibility to reuse logic by simply editing expressions.

Logical expressions can range from very simple to very complex comparisons; use the tips for writing logical expressions to help you create your logical expressions. This powerful task replaces the complicated, code-driven logic of scripting functions.

LOGICAL EXPRESSION RULES

The Logical Expression Rules tab allows you to configure the DataItems you will when making comparisons, along with the conditions required for that comparison:

Set keys at design or run time - these options allow you to specify when you will identify the DataItems you will use as keys.

Select the AND/Or logic for this expression - these options allow you to specify the condition needed for the value comparison.
Use AND if all the values in the expression need to be true, and use OR if only one of the values needs to be true.

Select the keys to choose sets of rules - these drop-down lists allow you to choose up to three keys to define the DataItems you will use in the value comparison.

View and input rules - this button allows you to view the Logical Expressions window. In the Logical Expressions window, you can view input data using all data combinations or in data combinations that are filtered by a particular key. You can also add, edit, and delete conditions for the expression by clicking the expression. To load saved data value combinations, click the Load Expressions button. To save new data combinations, click the Save Expressions button.

Write the output of the expression - these two drop-down lists allow you to define where the output of the task will be stored.
The DataItem to Hold the Expression Result holds the "True" or "False" outcome of the value comparison.
The DataItem to Hold the Expression Date allows you to select the DataItem that will hold the value associated with the outcome of the value comparison.

In addition to the options above are the Error Recovery options. These options allow you to specify the recovery method in the event a task fails to execute:

Mark task as failed implies that an error is rather unimportant to the business process.

Mark task as failed, raise With Exception Flag and continue process provides low-level notification that an error has occurred but continues the process.
Mark task as failed, raise With Exception Flag and stop process provides low-level notification that an error has occurred and stops the process.

Rollback and cancel the process and all subprocesses encountered is the most severe and instructs the process and any subprocesses to come to an immediate halt. This strategy is extreme and immediately impacts the business processes. Selecting this option implies that there is a strategy to resolve the processing error as well as dedicated resources to ensure that it is resolved quickly.

**Note:**
To save the changes you make in this tab, click **Apply** before moving on to another tab.

**Related topics**
[Adding a Logical Expression Rules task] | [Editing Logical Expressions] | [Tips for writing logical expressions]
TIPS FOR WRITING LOGICAL EXPRESSIONS

Below are tips that can help you write and edit logical expressions:

Accessing the Logical Expression window

Writing and testing logical expressions

Read more about Logical Expression Rule tasks.

TO ACCESS THE LOGICAL EXPRESSION WINDOW

From the Logical Expressions Rules tab, click the View/Input Rules button. The Logical Expressions window opens.

Click the ellipse in the Logical Expression cell of any key. The Logical Expression window opens.

Use the table below to help you understand the buttons on the Expression Builder Toolbar.

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ - / =</td>
<td>Arithmetic functions.</td>
</tr>
<tr>
<td>&lt;string literal&gt;</td>
<td>Alphanumeric text that do not include numbers used in calculations. Clicking this button opens the String Literal window. In the String Literal window, you can enter string literals.</td>
</tr>
<tr>
<td>=</td>
<td>Equal to</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>Not equal to</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater than</td>
</tr>
<tr>
<td>&lt;</td>
<td>Less than</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Greater than or equal to</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Less than or equal to</td>
</tr>
<tr>
<td>AND OR NOT</td>
<td>And, Or, and Not</td>
</tr>
<tr>
<td>( )</td>
<td>Parentheses</td>
</tr>
<tr>
<td>Parse Expression</td>
<td>This button looks for possible DataItems in the expressions and puts them into the Populate Variable to Test Expression array.</td>
</tr>
<tr>
<td>Test Expression</td>
<td>This button tests the logical expression.</td>
</tr>
</tbody>
</table>

Click OK.
The Logical Expression window closes.
TO WRITE AND TEST A LOGICAL EXPRESSION

Click the Expression Editor pane.
The editor pane is below the Expression Builder toolbar.
Enter the logical expression.

Note:
As a rule, do not use a question mark (?) to build an expression.

Add DataItems, string literals, and their values as necessary.

Use the buttons on the Expression Builder toolbar to help you build expressions.

Click the Parse Expression button.
The DataItems you used in the expression will appear in the Populate Variable to Test Expression array.

Note:
Before you can test an expression, you must enter the DataItem values first.

Click the Test Expression button.
The test results appear in the Expression Results pane.

Check the results of the expression to make sure the expression works correctly.

If the expression works correctly, click Apply.

If the expression works incorrectly, edit the expression and retest.

Related topics
Adding Logical Expression Rules tasks | Editing Logical Expressions
APPENDIX G – SAMPLE SME REVIEW REQUEST
Hello Bob,

Chris and I completed the first draft of the online help for Priority CS. Can your developers help us review a few topics? Because of the concentration on Batch, we have included the table below that identifies the topics we would like reviewed, the developers we need to review them, and the estimated time for each.

<table>
<thead>
<tr>
<th>Module to be reviewed</th>
<th>Subject-matter expert</th>
<th>Estimated review time in minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Communication tasks:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Creating E-mail tasks</td>
<td>Clay</td>
<td>20</td>
</tr>
<tr>
<td>• Using the advanced e-mail options</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Creating FTP tasks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Creating Group Note tasks</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Macro View:</strong></td>
<td>Allan</td>
<td>40</td>
</tr>
<tr>
<td>• Saving views of the workflow</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Auto-arranging child flows</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Printing macro views of workflows</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Viewing the workflow on the Workflow Designer Canvas</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Viewing your ToDo list from the macro view</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Nudging and aligning workflows</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Using the Workflow Viewer to displaying workflows in a macro view</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Office Integration Tasks:</strong></td>
<td>Erik</td>
<td>15</td>
</tr>
<tr>
<td>• Creating Excel tasks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Creating Excel Range Reader tasks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Creating XML to Excel tasks</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total: 72
<table>
<thead>
<tr>
<th>Task Type</th>
<th>Details</th>
<th>Responsible</th>
<th>Time</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML Tasks:</td>
<td>• Adding Query by Example tasks</td>
<td>Chris</td>
<td>15</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>• Adding Merge XML tasks</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Adding Document Designer tasks</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automatic Tasks:</td>
<td>• Adding Calendar Delay Tasks</td>
<td>Clay</td>
<td>25</td>
<td>115</td>
</tr>
<tr>
<td></td>
<td>• Adding Delay tasks</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Adding Multi-Instance Flows</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Adding Nested Flows</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Setting Product Status</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>To Do List:</td>
<td>• Adding items to workflow ToDo lists</td>
<td>Clay</td>
<td>10</td>
<td>125</td>
</tr>
<tr>
<td></td>
<td>• Updating ToDo list items</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Export Tasks:</td>
<td>• Creating WriteBack tasks</td>
<td>Chris</td>
<td>10</td>
<td>135</td>
</tr>
<tr>
<td></td>
<td>• Creating Table Writers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Advanced Tasks:</td>
<td>• Adding Logical Expression Rules tasks</td>
<td>Erik</td>
<td>20</td>
<td>155</td>
</tr>
<tr>
<td></td>
<td>• Tips for writing logical expressions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Adding scripting tasks</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Adding Web Service Invokers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Workflow Scripts:</td>
<td>• Assigning available scripts</td>
<td>Chris</td>
<td>30</td>
<td>185</td>
</tr>
<tr>
<td></td>
<td>• Creating new scripts</td>
<td>Erik</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Editing assigned scripts</td>
<td>Clay</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Importing and Exporting scripts</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Removing scripts</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Testing scripts</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linking Tasks:</td>
<td>•</td>
<td>Chris</td>
<td>10</td>
<td>195</td>
</tr>
</tbody>
</table>
• Setting up links for logical branching
• Using Connection Nodes to link tasks

Erik, or Clay

Manual Tasks:
• Prior Manual tasks
• User Type Assignment

Erik 10 215

We anticipate the technical review would take the developers 3.25 hours:
Please let us know when we can distribute printed copies to your developers for review.
Thanks,
Stacey Scott and Chris Schapman
Thank you for helping us review the Priority CS online help. Our overall objective for the online help is to provide just-in-time assistance to Priority CS users. The table below lists the attached topics we would like you to review and the estimated time for each review:

<table>
<thead>
<tr>
<th>Topics to be reviewed</th>
<th>Subject-matter expert</th>
<th>Estimated review time in minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Automatic Tasks:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Adding Calendar Delay Tasks</td>
<td>Chris</td>
<td>25</td>
</tr>
<tr>
<td>• Adding Delay tasks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Adding Multi-Instance Flows</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Adding Nested Flows</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Setting Product Status</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Data Export Tasks:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Creating WriteBack tasks</td>
<td>Chris</td>
<td>10</td>
</tr>
<tr>
<td>• Creating Table Writers</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Workflow Scripts:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Assigning available scripts</td>
<td>Chris</td>
<td>30</td>
</tr>
<tr>
<td>• Creating new scripts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Editing assigned scripts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Importing and Exporting scripts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Removing scripts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Testing scripts</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>XML Tasks:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Adding Query by Example tasks</td>
<td>Chris</td>
<td>15</td>
</tr>
</tbody>
</table>

Total: 35 minutes, 65 minutes, 80 minutes
Primarily, we are interested in your feedback about the processes; therefore, please ignore minor typographical errors. After reviewing the topics, please answer the following questions:

Are there any topics that do not reflect the processes accurately? Please list them below and write your comments on the topics.

__________________________________________________________________
__________________________________________________________________

Where applicable, are all the required steps indicated? Please list the topics you believe should have required steps indicated below:

__________________________________________________________________
__________________________________________________________________

Where can we add Notes to provide further explanation to the users? Please write the notes on the topic.

__________________________________________________________________
__________________________________________________________________

Where used, are the Examples accurate? Please list any topics we could add Examples to below:

__________________________________________________________________
__________________________________________________________________

Please write on the topics and insert any comments about the accuracy of the instructions. After your review, we will add your changes and then add images to the topics. Please return the topics to us by Friday, April 9.
Again, we appreciate the help and so will the users.
APPENDIX H – PLAN FOR DEVELOPING ROBODEMO MOVIES
### Linking Task

<table>
<thead>
<tr>
<th>Standard links (between to standard tasks)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical branch links (between two standard tasks)</td>
</tr>
<tr>
<td>Connection nodes (between parent and child flows/tasks)</td>
</tr>
</tbody>
</table>

**Specific example:**

Use the connection node to connect 2 Query by Example tasks with a Merge XML document task, and Document Designer task.

### Creating workflows

The nine step process: (text only)

- Turn a business process into a workflow
- Open a new workflow definition
- Include the DataItems for this workflow
- Define the flow properties
- Define the task properties
- Link the tasks together
- Save your completed workflow
- Trigger your workflow
- View service orders, work orders, and workflows

**NOTE:** This topic will be covered in training.

### Using the Workflow Designer

Pop up windows explaining the components of the Workflow Designer window:

- The Priority CS Workflow Menu
- The Workflow Designer canvas
- The Workflow Designer toolbox
<table>
<thead>
<tr>
<th>Started:</th>
<th>Finished:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workflow Management &gt; Using workflows &gt; Using the Process Management window to manage service orders and workflows &gt; <strong>Understanding the Process Management window</strong></td>
<td>Pop up windows explaining the components of the window.</td>
</tr>
<tr>
<td>Started:</td>
<td>Finished:</td>
</tr>
<tr>
<td>Workflow Management &gt; Using workflows &gt; <strong>Using the Workflow Designer toolbar</strong></td>
<td>Using the Workflow Designer toolbar to do the following: Align tasks, Nudge tasks, View tasks on a grid, Zoom in or out of workflow</td>
</tr>
<tr>
<td>Started:</td>
<td>Finished:</td>
</tr>
<tr>
<td>Workflow Management &gt; Using workflows &gt; Using the Workflow Designer toolbox &gt; <strong>Using the Workflow Designer toolbox</strong></td>
<td>Pop up windows explaining the components in the toolbox: Standard tasks, Manual tasks, Templates</td>
</tr>
<tr>
<td>Started:</td>
<td>Finished:</td>
</tr>
<tr>
<td>Workflow Management &gt; Task Management &gt; <strong>About the Task Management window</strong></td>
<td>Pop up windows explaining the components in the Task Management window.</td>
</tr>
<tr>
<td>Started:</td>
<td>Finished:</td>
</tr>
<tr>
<td>Workflow Management &gt; Using workflows &gt; Working with existing workflows &gt; Using the macro view &gt; <strong>About using workflows in a macro view</strong></td>
<td>Pop up windows explaining the components of the Workflow Viewer, emphasizing its use in project management.</td>
</tr>
</tbody>
</table>
APPENDIX I – SAMPLE ROBODEMO MOVIE
Linking tasks within a process

In this demonstration, we will show you how to do the following:

• Use standard links.
• Use logical branching links.
• Use connection nodes.

Use the navigation bar below to pause, go forward, or go back to any point on this demonstration.

Exit  Continue demo
Using standard links

This section demonstrates how you can use standard links to link tasks that have no branching criteria.

Continue to standard tasks
Using logical branching links
Using connection nodes

After adding the Manual task to the Process Designer canvas, click the Start node.
After adding the Manual task to the Process Designer canvas, click the Start node.

Nine handles appear around the node.
Click the center handle and drag it to the End node.

An arrow connects the task and the End node.
To define the link properties, double-click the handle of the link.

The Link Properties window appears.
For standard links, leave the default settings in the Link Type, Critical Path, and Wait on Task drop-down menus.
To save these properties and close the Link Properties window, click OK.

The Start node and the Manual task are now connected by a standard link.
Using logical branching links

This section demonstrates how you can use logical branching links. This type of link determines the direction of the process based on the DataItem value that is passed into the process.

In this demonstration, we will show you how to link the Start node to two New Query By Example tasks.
After placing and linking the tasks on the Canvas, click the handle of one of the links.

The Link Properties window opens.
From the Link Type drop-down menu, select Logical Branch Expression.

The Link Properties window expands.
Click OK to assign the logical expression to this link and close the Link Properties window.
The Link Properties window opens.

From the Link Type drop-down menu, select Logical Branch Expression.
The Link Properties window expands.

Enter the logical expression. This expression should be the logical opposite of the first.
Click OK to assign the logical expression to this link and close the Link Properties window.
The logical expression appears over the second link.

The process can now progress based on the value of the data item that passes through the logical expression.
Using connection nodes

This section demonstrates how you can use connection nodes to link multiple tasks to the same task.

Continue to connection nodes
Using logical branching links
Using standard links
Thank you for viewing this demonstration, brought to you by
Bluespring Software, Inc.
APPENDIX J – WEEKLY REPORT TO MY PROJECT MANAGER
To: Kayte Rudolph
From: Chris Schapman and Stacey Scott
Date: March 5, 2004
Re: Weekly status report #1 – Week ending Friday, March 5, 2004

The following report discusses our accomplishments from this past week, our goals for next week, and our concerns about the project and proposed resolutions to address those concerns.

**ACCOMPLISHMENTS FOR THIS WEEK:**

Single-sourced the project file, so both of us can work from a single file
Developed a preliminary style sheet
Created a plan for organizing the project folders
Documented this week’s and some of next week’s tasks, based on the project plan and preliminary task analysis you provided

**GOALS FOR NEXT WEEK:**

Continue documenting tasks on the task list and any tasks we may discover during our investigation of Priority CS
Develop a weekly report with QA issues for the developers
Continue developing the style sheet
Discuss software issues with you, Angie, and the SMEs for this project
Continue learning more about Bluespring’s operations
## Concerns and Proposed Resolutions:

<table>
<thead>
<tr>
<th>Concern</th>
<th>Proposed Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning the Priority CS system.</td>
<td>Ask questions to get a better understanding of Priority CS and how each module works.</td>
</tr>
<tr>
<td>Agreeing on style sheet guidelines.</td>
<td>Discuss stylistic views with one another and, when necessary, rely on a 3rd party source.</td>
</tr>
<tr>
<td>Documenting tasks not included on the project plan or preliminary task list. We believe these tasks are essential to users’ understanding of and interaction with Priority CS. Our major concern here is that including these tasks may extend our delivery date.</td>
<td>As of today, we are ahead of schedule by a day or two. This allows time to communicate additional tasks we discover with you and Angie in order to determine the tasks’ necessity.</td>
</tr>
</tbody>
</table>