This report explains the technical and scientific communication internship I performed after completing my MTSC classes at Miami University. In the report, I orient the reader to the context in which I performed the internship by introducing Ontario Systems, its products, organization, corporate culture, and the work I performed there. I proceed to describe the activities and projects in which I spent time during the internship. I then extensively detail my largest project, the letter management online help, outlining the process I used to complete the project. In the final chapter, I analyze the process I used on the letter management project by using the Anderson Problem-Solving Method for Technical and Scientific Communication. I show how the process I used at Ontario Systems represents an example of the problem-solving method.
# Table of Contents

## Chapter 1: Internship Overview
- Introduction .................................................................................................................. 1
- Products and Services ................................................................................................. 1
- Organization ................................................................................................................ 3
- Corporate Culture ...................................................................................................... 5
- My Job and Contribution ............................................................................................ 6
- Report Contents .......................................................................................................... 6

## Chapter 2: Summary of Activities and Projects
- Introduction ................................................................................................................ 7
- Activities .................................................................................................................... 7
- Projects ...................................................................................................................... 11

## Chapter 3: The Letter Management Project
- Introduction ................................................................................................................ 12
- The Letter Management Documentation Project ...................................................... 12
- The Writing Process .................................................................................................. 13
  - Planning .................................................................................................................. 13
  - Research ............................................................................................................... 15
  - Researching the Letter Phrases Document .......................................................... 17
  - Researching and Writing the Letter Addresses Document .................................... 18
  - Writing ................................................................................................................... 21
  - Editing .................................................................................................................... 21
  - Editing the Letter Definitions Document ............................................................ 24
  - Delivery .................................................................................................................. 27

## Chapter 4: Documentation Process Observations
- Introduction ................................................................................................................ 32
- Reflections on the Anderson Problem-Solving Model ............................................... 32
  - Define the Problem ................................................................................................ 32
  - Design the Solution ................................................................................................ 33
  - Test the Solution .................................................................................................... 33
  - Implement the Solution .......................................................................................... 35
  - Evaluate the Solution ............................................................................................. 35
- Final Thoughts ............................................................................................................ 35

## Appendix A: Technical Communicator Job Description .................................................. 37
## Appendix B: Letter Management Documentation Plan .................................................. 43
## Appendix C: SME Email Conversation about Letter Addresses .................................. 48
## Appendix D: Department Edit Form ............................................................................. 53
## Appendix E: Technical Editor Job Description .............................................................. 56
## Appendix F: Technical Edit Checklist .......................................................................... 61
This Work Is Dedicated To My Wife,

For Whom I Am Thankful.
Chapter 1: Internship Overview

Introduction
I performed my internship for the Master’s Degree in Technical and Scientific Communication by working as a Technical Communicator at Ontario Systems, LLC from February 24 to June 8, 2003. I worked at the Muncie, Indiana office, which is the headquarters of the company. Ontario Systems also has offices in Berlin, Ohio, and Cle Elum, Washington. Ontario Systems employs approximately 400 individuals, with close to 350 located in Muncie.

At the beginning of the internship period, I had already been a full-time technical communicator at Ontario Systems for nearly a year, so this report represents a small slice of what I had already been doing and what I continued to do at Ontario Systems for the next several months. In October 2003, I was promoted to a supervisor position within the department, which is the position I currently hold.

This chapter of the report introduces the company’s products and services, the organization and corporate culture of Ontario Systems, my contribution to the work done at Ontario Systems, and a forecast of the following chapters of the report.

Products and Services
Ontario Systems incorporated in October 1980. Since that time, the company has created, marketed, and sold software, hardware, and services to the receivables management industry. Any organization that processes a large amount of accounts receivable (or consumer or business credit accounts) could be an Ontario Systems customer. Customers include collections agencies, in-house collections departments (for example, the collections department for a credit card issuer such as a department store), and collection attorneys. In the third-party market comprised of collections agencies that collect on behalf of other organizations, Ontario Systems is the market leader, with 6 of the top 10 largest agencies using Ontario software. Ontario Systems also serves many of the nation’s largest hospitals. Approximately 50,000 individuals use Ontario Systems products daily.

Ontario creates and sells products to meet the needs of the company’s various markets. The discussion contained in the next page of the report introduces several of Ontario’s prominent products.

Artiva Architect
Artiva Architect is a platform of software tools produced by the systems architecture division of Ontario Systems. Artiva Architect is for application developers at Ontario Systems that build the Artiva family of products, which includes Artiva Agency, Artiva Healthcare, Artiva Legal, and Artiva Recovery. When customers purchase the Artiva Healthcare, Legal, or Recovery packages, they also receive Artiva Architect, which allows them to customize their software in alignment with their unique business needs.
Artiva Agency
Ontario Systems markets Artiva Agency to the third-party collections industry (i.e., agencies that collect on behalf of another party). The product helps agencies determine which accounts have the greatest potential for collection, and includes many tools to help account representatives work accounts more efficiently. The third-party market has been the largest portion of Ontario’s customer-base since the 1980s.

Artiva Healthcare
Ontario Systems markets Artiva Healthcare to healthcare providers (i.e., hospitals). The product is designed to handle complex transactions that are affected by government regulations, the healthcare insurance agency, and other variables to help hospitals improve cash flow through more efficient tracking and collections of accounts receivables. The healthcare market might be Ontario’s fastest growing market in the next few years. Customers include a number of the largest and most prestigious hospitals in the country.

Artiva Legal
Ontario Systems markets Artiva Legal to collections law firms. The product allows the law firms to track court cases for lawsuits initiated to settle their client’s accounts. The product also enables the firms to collect receivable accounts that have not yet entered the court system in the same manner as a third-party collection agency. The legal market is a new and growing market for Ontario Systems.

Artiva Recovery
Ontario Systems markets Artiva Recovery to first-party credit grantors, which includes any organization that offers credit cards. Recovery helps clients to collect receivables before outsourcing collections to a third-party agency, and to track receivables the organization decides to outsource using a third-party agency. The Recovery market is projected have the greatest growth potential of all of Ontario’s markets.

Flexible Automated Collections System (FACS)
FACS has been Ontario Systems flagship product since 1984. Ontario Systems markets FACS to collection agencies, healthcare institutions, debt buyers, and other third-party organizations that collect on behalf of others. FACS is slowly phasing out in favor of the Artiva family of products, though some agencies will likely continue using FACS for the next 10 years or more.

Guaranteed Contacts (GC) Telephony Dialers
GC dialers are hardware Ontario Systems builds and sells to be used with all of the previously described software packages. The GC dialers are computerized telephony machines that automatically connect account representatives using Ontario Systems software, to the parties responsible for the account balances, so representatives do not have to manually dial telephones. The dialers integrate with the software so that when a dialer connects a responsible party with an account representative, the account information for the responsible party appears on the screen of the account representative. Guaranteed Contacts products use sophisticated telephony technology and provide Ontario a high-margin source for the company’s revenue stream.
Ontario Systems provides add-on features and client support services for all products. Additionally, Ontario designed the Artiva family of products as individual applications that can be combined as components of a larger application for customers that work in more than one receivables management niche. For example, Artiva Recovery and Artiva Agency can be combined for companies that work in the first and third party markets. Ontario’s competitive advantage is enhanced by this ability as the receivables management market grows and consolidates, because no competitor offers a product that serves all niches.

**Organization**

I work in the Technical Communications department, which currently includes 17 individuals, including the department manager, two supervisors, 4 technical editors, and 10 technical communicators (aka technical writers). Each supervisor directs 7 individuals. The Technical Communications department is a part of Education Services, which also includes the Training Services team and the Knowledge Management team. Altogether, Education Services includes approximately 30 employees. Education Services is a part of the Client Services Division, which also includes the Application Support and Project Management departments. The division includes over 100 employees. Other divisions of the company include application development, systems architecture, administrative services, sales, and marketing. All division Vice Presidents report to Wil Davis, the President, Chairman of the board, and co-founder of Ontario Systems. Figure 1 shows the organizational chart for the Technical Communications department.
Figure 1: Technical Communications Organizational Chart
Corporate Culture
Ontario Systems has a corporate culture that affects how all employees work. The culture has similarities with other software companies and hi-tech organizations in that the organizational structure is fairly flat and most decisions are made in a decentralized and collaborative manner. The most unique aspects of the culture derive from the corporate core values of the company. The values listed below are taken from the Ontario Systems Web site:

Ontario Systems is defined by its SPIRIT of service. This company philosophy guides every business decision we make.

The Meaning of S.P.I.R.I.T.

Stewardship
We believe that all of life is a gift, and work is a part of that gift. We should use the resources we are given wisely. These resources include time, money, opportunities and talents.

Professionalism
We strive to treat our clients, vendors and each other in a professional and courteous manner. We believe this contributes to a positive atmosphere and helps to maintain our position as the market leader.

Innovation
As a software development company, it is our responsibility to constantly provide our customers with the latest technology available. As the market leader, we must provide innovative solutions our clients cannot find elsewhere.

Responsibility
Employees should act as if the company is their own. We expect employees to take responsibility for their work, professional development and contributions to the team. We also expect employees to professionally address issues where they feel the company is falling short of its goals and principles.

Integrity
We desire to be known as a company of integrity. We expect each employee to act with integrity during every interaction with a customer, fellow employee or other business associate.

Tolerance
Each person has been given the gift of a unique spirit and personality. We expect employees to accept and embrace the different gifts and abilities that have been given each of us.

Copyright © 2004 Ontario Systems, LLC. All rights reserved.

Employees are encouraged to think of our jobs as one of the gifts we have in life and to believe that we’re responsible to steward our jobs towards success. The culture encourages employees behave as if they own the company. In turn, the company recognizes that every employee has stewardship responsibilities outside of the company. At Ontario Systems, nearly all employees work only 40 hours per week; however, nearly all employees work very diligently for those 40 hours. The company recognizes that employees who have a strong stewardship ethic can produce more in a shorter amount of time than employees who work longer hours (in my experience at another software company, many employees worked more than 60 hours per week) but do not have a strong sense of stewardship and spend excessive amounts of time socializing at the water cooler. Employees know that the company will not put unreasonable
demands on them if they manage their responsibilities conscientiously, so that they can continue to positively steward their lives outside of work. The company helps enhance employee’s sense of ownership and loyalty through profit sharing and stock ownership. The corporate value system helps foster a high degree of respect and understanding between individual employees and the company management. The work environment becomes comfortable and productive at the same time, and the company has great success in the retention of employees.

**My Job and Contribution**

All of Ontario Systems’ products require extensive documentation, and the Technical Communications department is responsible for planning, researching, creating, and delivering the documentation for all products. The department is responsible for researching the best tools and processes to carry out the mission of the department:

“To produce and deliver timely, accurate, and relevant documentation to the internal and external customers of Ontario Systems by way of multi-purpose media.”

Ontario Systems internal and external customers use the documentation produced by the Technical Communications department in various ways:

- Application support consultants use our documentation to learn the products so they can provide support to outside customers and resolve support calls more quickly.
- The Training Services team reuses Technical Communication documentation from our content management system in order to build training materials used to train internal and external users.
- The Knowledge Management team reuses our documentation from the content management system to aid in the creation of knowledge base solution documents, which help internal and external customers solve problems.
- External customers use online help and manuals produced by the department to perform specific tasks and learn how to use, administer, and customize the Ontario’s products.

During the internship, I performed all the roles for which the department owns responsibility.

**Report Contents**

The following chapters of this report explain my experiences during my internship at Ontario Systems. Chapter 2 provides a broad overview of the activities I was involved in during the internship; Chapter 3 examines my largest project during the internship in detail: the production of the online help for the letter management functionality of Artiva Architect; Chapter 4 includes an evaluation of the process I used on the letter management project. Chapter 4 also summarizes my thoughts on the experience of working as a technical communicator at Ontario Systems, and it notes a few things I learned that might be helpful for the audience of this report. The report also includes various appendices, which should serve to illuminate experiences and ideas discussed in the report.
Chapter 2: Summary of Activities and Projects

Introduction
This chapter describes the activities and projects to which I devoted my time during the internship.

Activities
Every activity I undertook during the internship falls under one of the responsibilities outlined in Ontario Systems’ job description for a technical communicator. The job description identifies the following high-level responsibilities, and specifies the percentage of time the employee should spend on each activity (Appendix A contains the full job description for a technical communicator at Ontario Systems):

- Create printed and online documentation (60%)
- Manage assigned projects (20%)
- Communicate information to team members accurately and professionally (10%)
- Respond to company requests in a timely manner (10%)

Each of these responsibilities is connected to the others. For example, project management normally refers to the management of the technical communicator’s documentation projects (though sometimes team members may be required to manage other projects). Likewise, the information technical communicators communicate to a team member is often information learned while creating documentation or managing various projects and to help others complete their projects.

The company assigns technical communicators to particular projects based on the company’s need for the skills possessed by the technical communicator for a prioritized task or project. For example, because of the user advocate role and my knowledge of graphical design (acquired in the MTSC Information Design course), I was asked to be a part of a company-wide committee responsible for creating graphical user interface (GUI) design standards.

The following paragraphs describe my activities during the internship.

Development
I logged 451.25 hours of development during the internship. Development time includes a number of activities. For each document, I spent a few hours researching, reading developer design documents and obsolete documentation. Development time also includes writing: creation of the documents and the development of the content within the documents. Development time includes a few hours of revision for each document to incorporate comments and changes from the editing process. Finally, development includes time spent publishing the finalized documentation. Publishing tasks include integrating documentation into the online help system, testing the integrated documentation in the online help system, and the production of hard copy documentation.
Administration

I logged 121 hours of administrative time. A technical communicator typically uses an hour of ADM time a day. Administrative time includes project tracking and management. Technical communicators document progress and the amount of time spent on each document in the project tracking system. The timesheets described here were printed from the project tracking system. Administrative time also includes time spent on email, time management, and general organizational tasks.

Meetings

I logged 83 hours of meetings. During the internship, my team within the department had a one-hour meeting every week. The weekly team meeting is for the supervisor to share information with the rest of the team. Information can include notes about how the department is progressing, company news and events, and software upgrades to the department tools. I also had a monthly one-on-one meeting with my supervisor for an hour each month. The one-on-one is an opportunity for the supervisor to give feedback to the employee and for the employee to ask questions of the supervisor regarding documentation projects, vacation requests, or company guidelines. Other meetings included committee reviews, which are the meetings where subject matter experts (SMEs) review documentation, to ensure accuracy and to approve the documentation for publication. Sometimes, my department held all-staff meetings to announce or explain things that will affect everyone in the department. Once a quarter, Ontario Systems hosts a quarterly financial update for all employees.

Vacation

I used one-half vacation day during the internship, for which I logged 4 hours. Full-time employees at Ontario Systems accrue one day of vacation at the end of every month (or 12 days a year). After 5 years of service, employees accrue 1.25 vacation days a month, and after 10 years of service, employees accrue 1.5 vacation days a month. Employees can accrue a maximum of 30 vacation days at one time.

Investigations

I logged 1.5 hours to investigation tasks. I logged investigation time when I found bugs in the software. If anyone in the company finds a bug in the software they can log an investigation task in the project tracking system. The task goes into a queue, and a development manager assigns the task to a programmer to fix. During the internship I found a few bugs in the software, including misspelled words, and some problems with the actual design or logic in the software.

Sick

I used one half sick day during the internship, for which I logged 4 hours. Ontario Systems gives employees 6 sick days a year, and if an employee doesn’t use all of their sick days, the sick days roll over to the next year.

Quality Control

I spent 2 hours during the internship on quality control (or software testing) tasks. I logged quality control time to test the programming done to fix investigations that I’d logged. After
ensuring that the bug was fixed, I closed the investigation task for the bug in the project tracking system.

**Training**

During the internship I spent 2.5 hours in formal training. Ontario Systems offers employees a number of internal and external training opportunities. Employees can receive product training, software training, soft-skills training, and market-place training. The company also provides funds for employees to go to conferences, such as the STC conference. During the internship, I spent a lower portion of time on training activities because of tighter-than-usual deadlines for my projects.

**Holiday**

One paid holiday occurred during the internship, for which I logged eight hours. Nine or ten paid holidays normally occur a year.

Table 1, on the following page, lists the activities I performed during my internship. The table includes the type of activities, the number of hours on each activity, and the percentage of time spent on each activity. I recorded the data using the company’s project tracking system. The internally developed system tracks projects, individual tasks, and timesheets. This table is based on the timesheets I logged in the tracking system for each day during the internship.
<table>
<thead>
<tr>
<th>Activity</th>
<th>Week 1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>Total</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development</td>
<td>33</td>
<td>36</td>
<td>32.5</td>
<td>28.75</td>
<td>38.5</td>
<td>34</td>
<td>31.25</td>
<td>25</td>
<td>27.5</td>
<td>25.5</td>
<td>28.75</td>
<td>23.75</td>
<td>26.25</td>
<td>25.25</td>
<td>26.25</td>
<td>451.25</td>
<td>67%</td>
</tr>
<tr>
<td>Administration</td>
<td>6.5</td>
<td>6.25</td>
<td>7</td>
<td>7.75</td>
<td>5</td>
<td>7.75</td>
<td>8.25</td>
<td>11.75</td>
<td>11.75</td>
<td>7</td>
<td>7.5</td>
<td>8.25</td>
<td>8.25</td>
<td>8.5</td>
<td>9.5</td>
<td>121</td>
<td>18%</td>
</tr>
<tr>
<td>Meetings</td>
<td>0.5</td>
<td>2.75</td>
<td>5.5</td>
<td>8.5</td>
<td>1.5</td>
<td>4.25</td>
<td>5.25</td>
<td>10.25</td>
<td>4.25</td>
<td>8.5</td>
<td>6.75</td>
<td>7</td>
<td>9</td>
<td>2.75</td>
<td>6.25</td>
<td>83</td>
<td>12%</td>
</tr>
<tr>
<td>Vacation</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Investigations</td>
<td></td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.5</td>
<td></td>
</tr>
<tr>
<td>Sick</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quality Control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training</td>
<td></td>
<td>1</td>
<td>1.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Holiday</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8</td>
<td>8</td>
<td>1%</td>
</tr>
</tbody>
</table>
Projects

Most of the activities in which I engaged myself during the internship, I undertook as part of work on a documentation project. During the internship, I worked on the following projects:

- Architect Letter Management
- Architect System Functions
- Smaller Architect documents including User Reports and System Filters
- Other technical communicator’s projects

The letter management project was my highest priority during the internship and took up more time than all my other projects combined. Chapter 3 explains my work on the letter management project and the process that I used to complete the project.

The system functions project was my first project in the department, and I had worked on that in the nine months I spent in the department before the internship. The system functions project was also for the Artiva Architect product. Artiva Architect includes approximately 200 system functions, which are small reusable chunks of programming code that a client’s internal software development department can use to customize the product their company bought. I enjoyed the system functions project because it gave me the opportunity to use my software programming experience, and I had opportunities to write my own small scripts (or programs) to test the functions and illustrate how to use them in the documentation. I worked on system functions sporadically during the internship, at the times I could afford to take myself away from the letter management project.

I also intermittently worked on a few other smaller projects as time permitted, including the user reports and system filters. The user reports and system filters were also for the Artiva Architect project. They were also parts of the product that Product Management requested for the Artiva Recovery beta release. The user reports documentation was designed to help our client’s management use Artiva Architect-generated reports that measure employee performance. The system filters document was similar to the system functions documentation, because it was also primarily for programmers or other technical personnel who customize our products for their businesses. These projects only consisted of one or two documents, so I could complete them rather quickly.

I logged some time to other team member’s projects, usually when I completed a peer edit for one of their documents. I primarily performed peer edits for technical communicators that worked on projects related to my Architect projects (i.e., documentation projects that were for the same product I was working on). For example, two other technical communicators were assigned a number of documents for the Artiva Recovery beta release. They performed peer edits on my documents, and I performed peer edits on their documents.
Chapter 3: The Artiva Architect Letter Management Project

Introduction
I spent the largest amount of time during my internship preparing online help for the letter management component of Artiva. The department manager assigned the project to me on February 21, and the internship period began the next Monday, February 24. Documentation tasks on this project consumed at least 50% of my time at work during the internship. The table in Chapter 2 shows that documentation tasks consumed 67% of my time at work (not including some meetings); approximately 75% of my documentation time was for the letter management project (or just over 50% of my total time at work, still not including meetings and administration activities for the project). This chapter describes the letter management project and explains the process I used to complete the project. I explain each phase of the documentation process employed by the Technical Communications department at Ontario Systems and how I applied each phase to the letter management project.

The Letter Management Documentation Project
Letter management refers to a set of features in the Artiva Architect product that enable users to control letter requesting and printing in the Ontario Systems’ products that use Artiva Architect (i.e., the Artiva family of products explained in Chapter 1). Using letter management, the products can generate letters for clients, vendors, employees, responsible parties or anyone else for whom the system contains information. Customers use the features to create letter templates, use letter forms, print batches of letters, and merge information from their databases into the letters. Since all applications built on the Artiva Architect platform can use the letter management functionality, users of all the Artiva family products represent the audience for the letter management online help. Users include account representatives that collect receivables from responsible parties, agency managers, systems administrators, computer programmers employed by customers, and numerous internal users.

I was assigned the letter management project so that I could contribute to a major department push to complete an extremely large amount of documentation in a short period of time. The department needed to produce the documentation for preliminary publication for the March 31 beta release of the Artiva Recovery product. At Ontario, the product directors prioritize Technical Communications’ projects. The product director for Artiva Recovery produced a list of the documentation that the company needed to have available for preliminary publication for Artiva Recovery beta clients. Technical Communications provides the product directors with all of the department’s documentation plans, and then the product directors determine which plans and what parts of the plans are the highest priorities.

Approximately 60 new conceptual documents were required for the release, most of which hadn’t been written, including all of the letter management documents. Seventeen of the documents came from the letter management plan, which was included in the request because the letter management features of Artiva Architect include features the company emphasizes in
the marketing materials and sales presentations for the Artiva Recovery product. One technical communicator had been dedicated to the Artiva Recovery documentation, but there was too much work for her to complete by the deadline working alone. Throughout a year, most technical communicators average about 5 documents a month. Another technical communicator and I were pulled off our other projects and assigned a number of documents for the Artiva Recovery beta release. I was assigned the letter management project and a handful of other Artiva Architect documents that were a part of the effort. Completion of these documents remained my highest priority throughout the remainder of the internship. For the first five weeks of the internship I worked to prepare the letter management documents for preliminary publication, and for the remainder of the internship I worked on the stages of the documentation process that occur after preliminary publication. I did not complete the documentation process for the letter management project during the internship, but I did perform a preliminary publication of all conceptual help documents for the beta release of the Artiva Recovery product. I did not perform a final publication of the letter management documents until February 2004.

The Writing Process

The rest of this chapter explains each phase of the Technical Communications department writing process, and how I implemented each phase to complete the letter management project. Each member of the department uses the same process for their projects. The Ontario Systems writing process includes the following phases:

- Planning
- Research
- Writing
- Editing
- Delivery/Publication

The process begins with the planning phase and ends with delivery of the finished documentation (usually involving integration of documentation into the online help system). However, the process is not completely linear or one-way. For example, I often needed to conduct research during the writing and editing phases because I was unable to identify all areas of research at the beginning of the process. Feedback from users or subject matter experts (SMEs) sometimes necessitates that I move a project back to an earlier phase, and each stage of editing required more revisions or additional writing.

Planning

The goal of this stage is to create a documentation plan that outlines a proposed solution to meet the documentation needs of a product or set of product features. For online help projects, such as the letter management project, the documentation needs are normally very similar, so the documentation plans are usually similar as well. For example, all plans for online help projects propose documentation that contains the same types of information (i.e., all projects use the same types of concept and procedure headings to organize the information). Product plans are broken into sub-plans or units; the plan for letter management represents one unit of the Artiva Architect plan. Appendix B contains the documentation plan for the Letter Management unit of the Artiva Architect product.
To create the documentation plan, a technical communicator conducts research to identify all the documents that need to be created to cover the features, and to determine what kinds of conceptual and procedural information belong in each document. Conceptual information answers important questions about a feature, for example: What Is a Letter Definition, Why Use a Letter Definition, When to Update a Letter Batch, Where to Request a Letter. Procedural information explains how to perform a task, for example: How to Request a Letter.

After a technical communicator creates a documentation plan, the Writing Committee, made up of supervisors, technical editors, subject matter experts, and other stakeholders, must approve the plan. For larger plans, the committee might need to meet more than once. The Writing Committee reviews every plan to ensure the plan is consistent with other plans and fits with them. For example, since the letter management plan is one unit of the overall Artiva Architect plan, the committee had to ensure that information in the letter management plan didn’t overlap with other units in the Artiva Architect plan. The content for one plan usually includes hyperlinks to content in other plans, so the Writing Committee must also play a content management role to ensure the online help system contains consistent types of documentation and behaves in a predictable manner. If the concepts and procedures contained in the help system aren’t consistent, users could become confused and frustrated because they wouldn’t know what to expect. If users don’t know what to expect they might not continue to use the documentation. When the plan is approved, the technical communicator can begin the other phases of the documentation process. If documentation needs change during other phases of the process, the technical communicator updates the plan, with further approval of the Writing Committee.

When my supervisor assigned the letter management project to me, she gave me a copy of the letter management documentation plan, which the Writing Committee had already approved. My supervisor created the documentation plan and was therefore mostly responsible for the planning phase of the documentation project. In the Technical Communications department, the creator of the documentation plan is often not the same individual that actually creates and delivers the documentation. Usually, the planner is an experienced member of the team that already has familiarity with the company’s products and the department’s processes. More experienced members of the team better understand how all of the documentation plans fit together. New members of the department usually aren’t asked to write a complete plan until after at least a year of experience in the department. New technical communicators become familiar with plans as they work on their first few projects, and once they start a project they are given the responsibility to maintain the plan (recommending inevitable plan changes as the project progresses).

When I began the project, I was given responsibility for maintaining the plan. Even though the plan was reviewed and approved by the Writing Committee, I had to recommend quite a few changes to it after I began the research phase of the project. The technical editor for the project and I tracked all changes made to the plan, and the technical editor periodically presented the updates to the Writing Committee, and the committee approved our changes. At the beginning of the project, the plan was four pages long and it consisted entirely of a list of the concepts and procedures that should be included in each document. It included 17 conceptual documents
with 73 concepts and procedures. Today, the plan contains 18 documents, with 93 concepts and procedures.

Research
The research phase of the documentation process begins after the Writing Committee approves the documentation plan and the department assigns the project to a technical communicator. The planner performs research to create the plan, but they only determine the types of information to be included in the plan; they do not conduct the in-depth research necessary to determine the actual content itself. The purpose of the research phase is to determine the content to fill in the outline of the documentation plan. Thorough research prior to writing, ensures quality documentation and an efficient editing process; when insufficient research leads to incomplete or incorrect documentation, technical editors often have to perform research while conducting their edits, which can cause the documentation process to grind to a halt. For the letter management project I had to conduct thorough research, and I had to research efficiently, so that I could write quickly enough to meet the March 31 deadline.

Members of the department employ a number of different activities to research a topic. Ontario does not provide formal training in research, though mentors usually give new employees ideas when they start their first projects. For example, when I started in the department my mentor went with me to interview SMEs the first few times, and she helped me learn to navigate the applications. I used the documentation plan as my guide; researching one document at a time, until I could formulate the content outlined in the plan for that document.

Throughout the research phase of the letter management project, I made notes of what I learned. My notes almost constituted a rough draft for my documents. My goal was to record all of the information needed for the project so that I could simply refine and rework the content during the writing phase of the project. After completing a thorough documentation plan and conducting comprehensive research, a technical communicator should easily transition to the writing phase of the documentation process. For the documents in the letter management project, I conducted the following research activities.

Interviewing SMEs
My primary SME for the project was systems analyst located at the Berlin, OH office, where the Systems Architecture department developed Artiva Architect. We typically communicated using email; each day, I wrote down any questions I had, and when the list grew long enough, I composed an email containing all the questions. I tried not to send more than one email a day to limit the number of times I interrupted her, show respect for her time, and enable her to concentrate more deeply on letter management when I did need her help. She generally replied to my requests the same day, which greatly helped my efficiency with the product. We talked on the phone whenever the subject matter was too complex for a short and simple email. When she couldn’t answer a question, she worked with other systems analysts in Berlin and followed up with me as soon as she could. Secondary SMEs for the project included customer support representatives and systems analysts from the Muncie office.
Reading obsolete documentation
The software development teams reuse some features in new releases, so some older versions of documentation can be helpful. However, new versions of software usually have new features or changes to older features (i.e., old documentation can be helpful, but it should be questioned). Also, our department’s writing standards and style have changed so much since the last Artiva Architect documentation was produced that it is impossible to copy any of the old documentation. The version of Artiva Architect for which I was assigned letter management project is version 11, and there were a few earlier versions of the product that included extensive letter management documentation. Some of the information was still relevant, but much was out of date. Some of the current letter management features were covered in the legacy documentation, but not all. The obsolete documentation familiarized me with the general concepts so I could put the pieces of the puzzle together to form an overall understanding of letter management.

Reading developer specifications, designs, and other documents
Specifications are the plans developers use to create software applications; the specifications show how the system will work when its developed. Aside from specifications, developers sometimes create documents that explain how the software works after completing development. My SME from Berlin provided me with a few documents recently produced by programmers that explained how to use some new letter management features, especially the new features that enable users to use Microsoft Word to create letter templates to use with Artiva Architect. These documents did not cover all of the functionality I needed to document, but they were very helpful in teaching me some of the newer features. For the letter management project, no specifications or design documents were available or known to me, though in some of my other projects at Ontario Systems design documents have been a valuable aid to me; when good designs are available I can often write entire documents without disrupting an SME at all, which makes everyone’s work much more efficient and enjoyable.

Exploring the user-interface of the software
Exploration is usually the first thing I do when I get a new assignment; I want to see what I’m working on so I can understand how it works and begin formulating the content of the documentation in my mind. Exploration ensures the technical communicator understands how different parts of the system affect other parts (for example, how using letter management features of the system affect reporting features in the system). Exploration also helps to ensure that I find all the areas of the system to document. Using the documentation plan as a guide, I found the areas of the system that interface with the letter management features of Artiva Architect.

Using and testing the software
Testing is especially important for writing procedural information. Obviously, you must know each step of a procedure in order to document it correctly. The best way to learn a procedure is to perform it. To test letter management, I began carrying out the kinds of tasks a user would want to do. I used a test directory with fake database records so that I wouldn’t disrupt other people’s work. I used a test directory so that I would not disrupt development or quality assurance tasks in development directories. For example, I used letter management to define
my own letter templates and database information to merge into letters, request letters, and print letters.

**Reading system code**

Only a few technical communicators can read code. The CSA classes I took at Miami have given me the ability to read code, which has helped me to learn how parts of the application work. Reading code helps determine how functionality works. It is a skill that is helpful when working documentation projects for which software developers are the primary audience, because it gives the technical communicator the ability to think like a programmer. It wasn’t necessary for me to read code for very much of the letter management functionality, but the ability to read code has helped me learn how the company’s software works in a number of instances, usually on the developer documentation projects. In some cases, reading the code has prevented me from having to schedule time with SMEs to learn about the system.

**Interviewing internal users**

Internal users such as application support analysts can tell the technical communicator how clients use or will use the functionality. They can also provide usability feedback. Some of the subject matter experts on my committee were internal users. They knew how the functionality worked, and they knew what kind of information users needed in the documentation. On the letter management project, I worked with a few internal users that supported Artiva Architect for the company’s clients. They helped me learn about letter management by sharing with me how users have implemented letter management in the past; and they also shared the frustrations that clients sometimes have with letter management, so that I could use the documentation to highlight the information that can help alleviate their frustrations, thereby improving client’s experience with Ontario’s products.

**Researching the Letter Phrase Document**

The evolution of the letter phrases document indicates how extensively I changed the documentation plan for the letter management project because of my research. When I began the project the plan included the following concepts and procedures in the letter phrases document:

- What is a Letter Phrase
- Why Use a Letter Phrase
- How to Create a Letter Phrase
- How to Add a Letter Phrase to a Letter Body

I discovered topics that required plan changes while researching letter phrases. In my research, I first read the obsolete documentation, which included information about letter phrases. I learned the conceptual idea of letter phrases: they are small sections of text that you can insert into a letter body. For example, you can create a letter phrase that merges client address information from the database. Then, in any letter that you send to a client, you can insert the same letter phrase into the letter definition. After reading the obsolete documentation, I explored the letter phrase functionality in the system by searching the menus and windows in Artiva Architect that compose the letter phrase maintenance area. In my exploration I created some test letter phrases and inserted them into letters to understand how the application behaves and how a user could, would, and should use letter phases.
I quickly realized that letter phrases included important features that were not included in the original plan. The original plan would have only showed a user how to create a text letter phrase and how to insert the letter phrase into a letter body. By using only those aspects of letter phrases, clients would spend more time setting up the letter phrases than actually benefiting from them (meaning, clients wouldn’t use letter phrases). Letter phrases save users time when the phrase include database fields, scripts, system variables and other elements that enable the letter phrase to generate dynamic content (merged from the database) in the letter body when a user requests a letter; however these aspects of letter phases weren’t included in the original plan. I shared my findings with my development SME and the technical editor on the project. They both agreed that adding the additional procedures to the plan was necessary to ensure user comprehension of letter phrases. I recommended and implemented similar changes to many of the other documents in the letter management plan.

When I completed the project, the final version of the document included the following concepts and procedures:

- What is a Letter Phrase
- Why Use a Letter Phrase
- How to Create a Letter Phrase
- How to Add a Field to a Letter Phrase
- How to Add a Script to a Letter Phrase
- How to Add a Printer Control Code to a Letter Phrase
- How to Add a Shared Variable to a Letter Phrase
- How to Add a System Variable to a Letter Phrase
- How to Add a Letter Phrase to a Letter Phrase

**Researching and Writing the Letter Addresses Document**

The Letter Addresses document was the most difficult document for me to understand and write about in the letter management project. Letter Addresses was the hardest document to write and understand because it was the hardest topic to research. It was the first document I began, and the last I completed. It involved 3 dedicated committee reviews. The document was originally planned to include the following concepts and procedures:

- What Is a Letter Address
- Why Use a Letter Address
- How to Create a Letter Address

The finalized document included the following concepts and procedures:

- What Is a Letter Address
- Why Use a Letter Address
- How to Create a Letter Address
- How to Add a Letter Address to a Letter Definition
- How to Display a Letter Address in a Letter Body
- How to Set a Letter Address to Send Letters Using Email
- How to Set a Letter Address to Send Letters Using OSC Link
- Why Use the SyAddress Script
- How to Display an Alternate Address on a Letter
Like the letter phrases document, I considerably changed the plan for the letter addresses document. Whereas I changed the letter phrases plan because a few procedures were needed to fully explain letter phrases, I changed letter addresses because I discovered that letter addresses were a profoundly different conceptual idea than the department had thought.

The plan only included two concepts and one procedure: What Is a Letter Address, Why Use a Letter Address, and How to Create a Letter Address. After my initial use of the system, I immediately realized that after you created a letter address, you couldn’t use the letter address unless you added it to a letter definition, so I added a procedure for adding a letter address to the plan. However, the plan still didn’t explain letter addresses, because when I added a letter address to a letter definition and then printed a letter, the letter address didn’t appear. In conversations with my SMEs, I learned that most users do not use letter addresses to display address on their letters; they use the letter phrase feature instead. I didn’t know why letter addresses were in the system and I didn’t understand how they worked. The SMEs didn’t know either (ergo, most clients probably didn’t know how to use letter addresses either). The two things I did know were that letter addresses were much more complicated than I’d originally thought, and that they were so complicated that without adequate documentation, Ontario’s internal and external clients could not use them. Ironically, at the point I spoke with the SMEs about my concerns, the letter addresses document had already been through a committee review (with the SMEs approval) and published.

I remembered that parts of the letter management portion of Artiva Architect were developed 10 or more years ago, so I started reading old documentation. I never found a document that explained letter addresses, but I found other documents that explained parts of the application that related to the conceptual idea of addresses. Finally, I learned that a user could insert a script called SYADDRESS into a letter body to display a letter phrase. I found the source code of the SYADDRESS script and it included logic that enabled users to specify an address when they request a letter. I added the script to a letter definition and requested a letter. When I requested the letter, the system asked what letter address I would like to add to the letter. I could choose the default address or an alternate address: settings that were on the letter address I had added to that letter definition. Enabling account representatives to specify a letter address when they request a letter was useful functionality that I hadn’t read about in any obsolete documentation or been told about by an SME.

Finally, I learned why letter definitions helped a user: they allowed the user to specify an alternate address when they printed the letter. The Letter Addresses document needed to explain how to create a letter address, add it to a letter definition, and how to use it by inserting the SYADDRESS script into a letter, all in order to let the letter-requesting user decide which address to add to the letter (for example, they might send a letter to the recipient’s home address instead of a work address). The SYADDRESS script also enabled users to send letters to email addresses using an add-on product, the OSC Link server, which distributes letter services to third-party vendors. Appendix C shows an email discussion with my primary SME on the project. The discussion is four pages long and spans several weeks. It shows how I interacted with the SME using email, and it shows how she did not always know much of the information I needed to include in the document.
The knowledge of how letter addresses worked had been lost in the company. Since the document needed to be changed so much, I decided to take the document through another committee review. I had to explain to the SMEs what I’d learned about letter addresses, and how I believed the functionality was designed to work. The SMEs agreed and approved the release of the document again.

Figure 2 shows the text of the What Is a Letter Address concept that was approved at the first committee review.

**Figure 2: What Is a Letter Address After Committee Review**

**What Is a Letter Address?**
A letter address is an address you can assign to a letter definition.

You maintain letter addresses at the Letter Addresses menu. Letter address contains the values of database fields, you determine. You add an address to a letter definition in the Address Information window on the Letter Definitions menu. You can set an option in the Address Information window of the letter definition to prompt the user to assign an address to a letter request when the user requests a letter from the account screen.

**Tip**
The user-assigned address overrides the address you on the letter definition.

You must specify the following information for a letter address:
- The source table from which to return field values
- The request field, which you can use to limit the number of addresses the user can assign to a letter request
- The key field for the SyAddress script to reference the print address in a letter
- The letter phase in which to print the address if you want to print the address in the letter body

You can optionally specify the following information for a letter address:
- A sort field to sort all letter requests by a letter batch, the use address sort fields and values to sort the fields if the SysReturnMail function executes for a returned letter
- A validation script to determine if the values of the address fields on a letter request are valid

The final version of the document contains very different conceptual information based on what I learned as I continued to investigate letter addresses. Figure 3 shows the final version of the same concept.

**Figure 3: What Is a Letter Address Final Version**

**What Is a Letter Address**
A letter address contains the settings and fields for an address.

You add a letter address to a letter definition. The letter address displays in the letter where you add the SyAddress script. The system parses the information in the letter address when the letter is printed. The fields and text in the letter body that follow the letter address move down to accommodate the expanded letter address.

A letter address identifies:
- A table that contains the fields for the address
- A table of the letter definition with which to use the letter address
- A field on the table of the letter definition that points to the table of the letter address. The SysAddress script you add to the letter body references the field to display the address in a letter
- A letter phase that contains an address to display in the letter body by using the SysAddress script

You can set a letter definition to prompt for an alternate address to display on the letter. This prompt appears when the user requests a letter from the account screen. You must set up the letter definition, one or more letter addresses, and one or more letter phases to use an alternate address.

A letter address can include an email address. To send a letter by using email, you must select Email as the device when you:
- Print the letter batch.
- Print the letter immediately when you request a letter from the account screen.

You can also use a letter address to send letters by using Link. To use Link to print a letter, you must select Link Server as the device when you:
- Print the letter batch.
- Print the letter immediately when you request a letter from the account screen.

You can use a script to prevent a letter from printing if invalid data exists for a letter address.

I researched all the documents in the project using similar methods. I always browsed the screens in the application, tried to use the application, talked with SMEs, and read obsolete documentation. In most cases, those research methods enabled me to learn the topics I needed to document. However, the letter addresses document shows that sometimes all of those sources still can’t provide all the information needed. I had to learn the technical details of letter addresses by reading code and extensively testing and retesting the application. Design specifications would have helped me immensely. Even if the original programmers had written design specifications, I might not have found the documents because most of letter management was created in the early 1990s.
Writing

The writing phase of the documentation process begins when the technical communicator completes research on a topic from the documentation plan. After creating a strong plan and conducting extensive research, writing the documentation should be a straightforward process; it should be the process of molding information learned and recorded during research into content that meets the department’s standards for online help. Technical communicators use department writing guidelines, the department style guide, the department list of words to use and avoid, and other internal resources to aid in writing the documentation. We also use external resources such as the Microsoft Manual of Style and the Greg Reference Manual. New technical communicators should be able to learn the common standards after a few projects, which should enable them to complete the writing process more quickly and easily. The department uses standards to ensure consistency, quality, and usability. Most of our standards respond to usability issues encountered in online help systems. In our product’s online help systems, users can search and hyperlink to thousands of concepts and procedures, which have been written by at least 15 different members of our department. The standards ensure that all of the department’s documentation appears to have been written by one person, so that users do not suffer a drop-off in quality or consistency when they link from one concept to another in the help system.

I began writing each conceptual help document after I researched the concepts and procedures contained in the documentation plan for the document. To create each document, I performed the following activities:
1. Created a new document using the module document type definition (DTD) in the XML editor on my PC. The DTD serves as a template that allows technical communicators to include only predefined elements such as concepts, procedures, paragraphs, and tables.
2. Added the concept and procedure elements contained in the documentation plan to the document.
3. Saved the document to the content management database. All content produced by the Technical Communications, Training Services, and Knowledge Management departments is stored in the database. The database provides version control, security, and the ability to reuse information from one document into another.
4. Wrote the content into the document for each concept and procedure using writing tools and resources and resave the document to the database.

For the letter management project, I created all of the documents in the documentation plan, added the concepts and procedures to the documents, and saved the documents to the database in the first few days that I started working on the project. I wrote the content into each document after I thoroughly researched the document. After writing a document and reviewing it for standards, I submitted the document into the department editing process.

Editing

When a technical communicator writes a complete draft of a document, he or she submits the document into the department editing process. The department editing process includes a number of different edits, which are performed by different people and have different goals. Each edit normally results in some modifications to the content; each modification should help the document meet standards for consistency, quality, and usability. Sometimes, after an edit,
the technical communicator might need to conduct follow-up research and/or change the documentation plan in order to incorporate edits. Appendix D contains the department edit form. The department uses the following edits:

**Supervisor Edit**
The technical communicator’s supervisor conducts an edit to catch obvious errors and to become familiar with the content of the documentation. For example, the supervisor ensures that the technical communicator includes each topic from the documentation plan. Supervisors do not perform in-depth editing tasks such as checking hyperlinks or test the procedural information contained in the documentation. The goal of this edit is for the supervisor to become familiar with the documentation which they will also review in the final edit and to ensure the technical communicator is generally on the right track with the document. A supervisor can usually complete the edit within 2 or 3 days. Since the letter management project had a tight deadline for preliminary publication, my supervisor completed supervisor edits for each document within one day.

**Peer Edit**
After the supervisor edit, the supervisor assigns another technical communicator in the department to complete a peer edit. The peer edit is similar to the supervisor edit, because the goals are similar: to catch obvious errors, check for continuity and completeness, and for the peer editor to become familiar with other documentation the department is producing. The peer edit also enables the peer editor to gain editing experience. Technical communicators often work on related projects that are linked in the online help system; the peer edit is one way in which writers can begin to see how their documentation fits into other documentation projects. By using consistent standards across all documentation, understanding how the documentation in one project fits into the documentation in other projects, and working to ensure all of the documentation within the online help system does fit together, we can produce documentation that appears to have been written by one person, rather than a department of 15. The peer edit normally takes one or two days to complete. For the letter management project, other members of the department that worked on the Artiva Architect or Artiva Recovery products normally completed my peer edits. My peer editor’s own projects were closely linked in the online help system with the letter management project (for example, their documents often contained hyperlinks to the letter management document).

**Committee Review**
After incorporating supervisor and peer edits, the technical communicator holds a committee review to check documents for technical accuracy. The committee review does not focus on grammar or style. The committee for each project is composed of the technical communicator, a technical editor, a Technical Communications department supervisor, a development SME, an application support SME, a quality assurance representative, a product director, and potentially others depending upon the project. At a committee review meeting, the committee reads through documentation to ensure the documentation is accurate. Members of the committee can ask questions about the documentation or the functionality documented. After the technical communicator completes revisions from the committee review the documentation is ready for preliminary publication and the department edit. The company uses documents in preliminary publication to send to internal and external customers (such as beta customers) that
need the documentation before the editing process concludes. Typically, a technical communicator can review 2-5 documents (or 20 pages of documentation) during a committee review. Technical communicators schedule the meetings and send electronic copies of the documents approximately 48 hours to a week in advance to give time to the SMEs to prepare for the meeting. The best committee reviews occur when the members of the committee come prepared for the review. They should read the documentation and come with questions. The technical communicator should explain the purpose of the meetings and each person’s role in the meeting. For the letter management project I scheduled 3 reviews in March right when I started the project. Since we needed to review all the letter management documentation by the end of the month, I wanted to reserve the meeting times on the committee member’s schedules. I sent the documentation out approximately a week before the review so that the committee members could read the documents before the meeting. After the March 31 deadline passed, I conducted a few subsequent reviews to go back over questions that arose during the editing process.

Department Edit
The department edit is the most extensive edit of the documentation. The technical editor assigned to the project conducts the edit. The job description in Appendix E explains the responsibilities of the technical editor position. The technical editor edits for grammar, style, consistency, continuity, technical accuracy, etc. Technical editors test the documentation, and sometimes perform research to determine if the documentation is accurate and complete. Sometime, the technical editor requests a second department edit. Department editors edit for 4 or more technical communicators, so they become familiar with the documentation across the department, familiarity they need to ensure that the department produces consistent documentation across all features and products. For the letter management project, I submitted all of the conceptual documents to department edit together on May 22. They were given back to me for revisions on July 15, and the editor requested a second edit. I submitted the documents for the second edit on September 11, and they were given back to me on November 10.

Technical Edit
This edit is conducted by an authoritative development SME who checks the document for technical accuracy. The department provides the SME with a checklist to fill out, sign, and return to the department. The technical edit checklist is included in Appendix F. I submitted the letter management documents to technical edit on December 19, 2003, and received them back, between January 8 and February 11, 2004.

Final Edit
The supervisor conducts the final edit to ensure that obvious errors have not entered the documentation as a result of the editing process and to ensure that the technical communicator addressed all previous edits. The supervisor reviews previous edits to ensure the technical communicator understood the edits and incorporated the edits appropriately. The documentation can be delivered after the final edit is complete. Final edits are usually returned in just a few days. All of the final edits for the letter management project were returned to me in one to three days.
Editing the Letter Definitions Document
The letter definitions module was the largest module in the project. To set up letter management, users must first understand letter definitions. The module went through all of the edits described in this chapter. The peer edit and supervisor edit did not suggest very significant changes to the module. After writing my initial draft of the document, I submitted the document to supervisor edit on March 6, 2003. I received the markup for the edit on March 11, 2003. Only small terminology changes and the addition of a hyperlink to another document are suggested. Figure 4 shows the marked up copy of the supervisor edit on the What Is a Letter Definition concept.

Figure 4: What Is a Letter Definition Supervisor Edit

Letter Definitions
What Is a Letter Definition?
A letter definition is a letter template you create from the Letter Definitions menu.
A letter definition consists of the following parts:

- The letter body you create in the Letter Body window or Microsoft Word
- The margins and page size you set in the General Information window.
- The form type you set in the General Information window.
- The table you set in the General Information window. You can insert fields into the letter body from the table and all of the table’s parent tables. You must use a Letter or General script to insert a field from a child of the table you specify.
- The address to send the letter to in the Address Information window
- The Pre-Print event and the Pre-Update event in the Letter Events window. You can also use letter events on the table of the letter definition.

You use the letter definition when you create a batch of letters for a number of accounts. You can use the SylLetter system service to request and print letters.
Use the letter definitions report to view a letter definition or all letter definitions.
The department edit normally requires extensive revisions. Figure 5 shows the markup for the department edit. The markup shows that the department editor focused on grammar, organization, sentence structure, and terminology among other things. The edits do not reflect changes in the meaning of the content. After making revisions based on the committee review, I submitted the document for the first department edit on May 22, 2003. I received the markup on July 15, 2003.

Figure 5: What Is a Letter Definition Department Edit

Letter Definitions

What Is a Letter Definition?
A letter definition is a letter template you create from the Letter Definitions menu.

A letter definition consists of the following parts:

- The contents of the letter body. You can use Microsoft Word to edit the letter body.
- The margins and page size of the letter.
- The letter form type.
- The table. You can add tables to the letter body from the table and parent tables.
- The letter address to send the letter.
- The preprint event and the presave event.

Tip

You can also use letter events on the table.

- Preprint event
- Presave event

You use the letter definition when you create a letter batch for a number of accounts.

You can use the SysLetter system service to request and print a letter definition.

Run the letter definitions report to view a letter definition of all letter definitions.
The technical editor for the letter management project requested that the modules be submitted for a second department edit. The edit did not require extensive changes. Figure 6 shows the markup for the second department edit. This excerpt only suggests two grammatical changes and one minor organizational change. After making revisions based on the first department edit, I submitted the document for the second department edit on September 11, 2003. I received the markup on November 10, 2003.

**Figure 6: What Is a Letter Definition 2nd Edit**

---

**Letter Definitions**

**What Is a Letter Definition**

A letter definition is a letter template you create that determines the:

- Contents of the letter body
- Margins and page size of the letter
- Letter form type of the letter
- Table that contains the data to use in the letter
- Letter address

You can set the letter definition to prompt for an alternate address to display on the letter. This prompt appears when the user requests a letter from the account screen. You must set up the letter definition, one or more letter address, and one or more letter phrases to display the alternate address.

You can set the following letter events on the letter definition:

- Preprint event
- Preupdate event

You select a letter definition when you create a letter batch and when you request a letter. The system uses the letter definition to create a letter when you request the letter.

Run the Letter Definitions Report to display the information about a letter definition.
The technical edit and final edit did not require more than cosmetic changes to the document. After making revisions based on the second department edit, I submitted the document for technical edit on December 19, 2003. I received the markup for the technical edit on February 11, 2004. After making revisions based on the technical edit, I submitted the document for final edit on February 13, 2004. I received the markup for the final edit on February 16, 2004. The following illustration shows the markup for the final edit. Figure 7 shows the marked up copy of the module after final edit. It includes only one question that asks whether I should link the word, request, to another document.

**Figure 7: What Is a Letter Definition Final Edit**

![Figure 7: What Is a Letter Definition Final Edit](image)

**Letter Definitions**

**What Is a Letter Definition**

A letter definition is a letter template you create that determines the
- Contents of the letter body
- Margins and page size of the letter
- Letter form type of the letter
- Table that contains the data to use in the letter
- Letter address

You select a letter definition when you create a letter batch and when you request a letter. The system uses the letter definition to create a letter when you request the letter.

You can set the letter definition to prompt for an alternate address to display on the letter. This prompt appears when a user requests a letter from the account screen. You must set up the letter definition, one or more letter addresses, and one or more letter phrases to display the alternate address.

You can set the following letter events on the letter definition:
- Preprint event
- Preupdate event

Run the Letter Definitions Report to display the information about a letter definition.

**Delivery**

When a document completes the editing process, the technical communicator must prepare the document for delivery and integration. For most online help projects, including all of the projects on which I’ve worked, delivery of the documentation involves the following activities:
- Set the task for the document in the project tracking system to final status. When the task is in final status, other members of the department can reuse information from the document or link to the document, with assurance that the author won’t continue revising the document. Training Services and Knowledge Management can also reuse information from the document once it is finalized.
- Print the conceptual documents in PDF form to be submitted for posting to the internal intranet for internal users and the Client Resource Center (Web site) for external users. All technical communicators email the PDF file to the member of the department who
is responsible for coordinating the posting of the PDF files with the IT department. The preliminary publication that occurs after the committee review uses the same process.

- Submit the task number of the conceptual documents or contextual help to the systems architecture team for integration into the online help system. The systems architecture group first integrates the help to a controlled testing environment, so the technical communicator can assure the help displays in the help system correctly and in the appropriate context. After quality is assured, the systems architecture team integrates the help to a release environment, from which the product releases are created to deliver to clients. The preliminary publication that occurs after a committee review does not include the integration process.

Figure 8 shows the What Is a Letter Address Concept displayed in the online help system after integration. The blue and underlined text represents hyperlinks that users can select to access other concepts or procedures in the online help system. Nouns link to other What Is concepts that define something in the software. Verbs link to procedures that explain how to perform some kind of operation in the system.
Figure 8: What Is a Letter Address in Online Help

What Is a Letter Address

A letter address contains the settings and fields for an address.

You add a letter address to a letter definition. The letter address displays in the letter where you add the SyAddress script. The fields and text in the letter body that follow the letter address move down when the letter prints to accommodate the letter address.

A letter address identifies:

- A table that contains the fields for the address.
- A table of the letter definition with which to use the letter address.
- A field on the table of the letter definition that points to the table of the letter address. The SyAddress script you add to the letter body references the field to display the address in a letter.
- A letter phrase that contains an address to display in the letter body by using the SyAddress script.

You can set a letter definition to prompt for an alternate address to display on the letter. This prompt appears when the user requests a letter from the account screen. You must set up the letter definition, one or more letter addresses, and one or more letter phrases to use an alternate address.

A letter address can include an email address. To send a letter by using email, select Email as the device when you:

- Print the letter batch.
- Print the letter immediately when you request a letter from the account screen.

You can also use a letter address to send letters by using Link. To use Link to print a letter, select Link Server as the device when you:

- Print the letter batch.
- Print the letter immediately when you request a letter from the account screen.

You can use a script to prevent a letter from printing if invalid data exists for a letter address.

Additional help:

- More about this topic
- All information about this topic

-----------------------
Table 2, on the following page, shows:

- When I submitted each document to each stage of the editing process.
- When each stage of the editing process was completed.
- When I submitted each document for final integration.
- When the final integration was completed.

This chapter described the work I performed on my largest project, the letter management online help. During the internship, I wrote the documents contained in the project’s documentation plan and began submitting the documents to various stages of the editing process. After the internship, I completed all revisions and published the documents. In the following chapter, I analyze the process I used to complete the letter management documentation.
<table>
<thead>
<tr>
<th>Document</th>
<th>Asgn:</th>
<th>Peer Edit</th>
<th>Comm. Review</th>
<th>Dept Edit</th>
<th>2nd Edit</th>
<th>Tech Edit</th>
<th>Final Edit</th>
<th>Published</th>
</tr>
</thead>
</table>
Chapter 4: Documentation Process

Observations

Introduction
Chapter 1 introduced the work I did for my MTSC internship and the company where I performed the internship, Ontario Systems LLC. Chapter 2 described the activities and projects in which I engaged during the internship. Chapter 3 explained the Architect Letter Management project, my main project during the internship.

This chapter concludes the report by showing how the documentation process I used during my internship is an example of the Anderson problem-solving model taught in the MTSC program. I conclude this chapter with a few additional thoughts about the MTSC program, the field of technical communications, and my work as a technical communicator.

Reflections on the Anderson Problem-Solving Model

The Anderson Problem-Solving model is a framework that outlines how technical communicators can attack any documentation problem. The model includes the following phases:

- Define the Problem
- Design the Solution
- Test the Solution
- Implement the Solution
- Evaluate the Solution

The model is heuristic so that it can apply to any documentation context. The contexts in which the model may be used can vary by employer, subject matter, medium, audience, and other factors. The following discussion shows how the problem-solving model applied to my context, that of producing online help for receivables management software for Ontario Systems’ clients.

Define the Problem

I did not perform this part of the problem-solving module specifically for the letter management project. We define the problem for all projects at once because all the projects come together as one very large deliverable (the online help which contains thousands of topics of documentation). The purpose of the online help is to explain to our clients how to use the software to meet their business needs.

When the company began producing online help approximately four years ago, the Technical Communications department determined what types of information need to be included in online help in order to show clients how to use the software. The department created a guide that explains what information types can be used in online help, what kind of information each type includes, and when to include the information types in a documentation plan to explain a
feature of the software. A committee meets to update the guide when we find that we can’t explain something with the information types available.

The department also had made other decisions that affect the problem definition of every online help project. For example, the audience for our projects is usually composed of the same types of users (account representatives, internal and external programmers, client managers, cashiers, etc), so we don’t have to reanalyze our audience for every project the department works on. We also don’t completely reconsider the context in which our users use the documentation. We decided to use online help years ago because users need the help while they are performing business tasks within the software. When we do change our thoughts on the problem (purpose, context, or audience), the changes affect all online help projects, not one project at time.

**Design the Solution**

Like problem-definition, some parts of the design process for the letter management project were predefined by existing department standards. I didn’t have to make any decisions about medium, form, style, production, or distribution. My work was part of the online help solution for Artiva Architect, and the department largely made those decisions when we began producing online help instead of hard-copy manuals. Even though these decisions had been made, I did a lot of work during this stage to gather information and draft documents for review. Chapter 3 explains how I gathered information in the research section and how I wrote the first drafts of the letter management documents to submit to the editing process. I was able to gather information quickly by using various resources, including conversations with SMEs and internal users, reading through obsolete documentation, reading programmer documents about the functionality, and exploring and using the software. I drafted the preliminary documents as I gathered information using the resources and tools the department uses for all online help projects.

Since the department doesn’t remake many decisions in the problem defining and designing stages of the problem-solving model for each project, we can operate in an efficient manner. Project planning is easier because, the planner doesn’t have to create a new style guide, determine how to deliver the solution, or define what types of information should be included. The planner simply applies the standards and processes the department has in place to the part of the software included in the project. The only significant difference from project to project is determining how to apply the standards (for example, deciding what information types from the guide are needed to explain a topic). While this is efficient, the department does need to be wary of thinking too narrowly about documentation. Today, we believe online help is the best medium for our audience and the context in which they use the company’s software. However, as the software, our clients, and technology change, we might need to augment or replace online help with another form of communication.

**Test the Solution**

The editing phase of the Ontario documentation process is a very good example of the Test phase of the Anderson problem-solving model. As I stated in chapter 3, the editing process ensures that all online help produced by the department is accurate, consistent, and standardized. The editing process includes the committee review, in which SMEs and internal
users review the documentation to ensure that it is accurate and helpful. After the committee review the department editing, performed by full-time technical editors, ensures that the documentation is consistent and standardized for online help. The Technical Edit, performed by an authoritative SME and the Final Edit, performed by a Technical Communications department supervisor, ensure that unintentional errors have not entered the documentation during earlier phases of the editing process. After each stage of the editing process the author analyzes the responses and edits and decides how to incorporate them into the documentation. All online help projects at Ontario follow the same process that I used on the letter management project.

Overall, I think that Ontario’s editing process does a good job of producing accurate, consistent, and standardized documentation. When I read the department’s documentation, or hyperlink from topic to topic in the online help, I cannot tell that the each more than 15 individuals helped to author the system. While I appreciate the quality produced by the editing process, we have had to take steps to streamline the editing process. We started paying so much attention to standards and consistency that many projects (including the letter management project) almost died in the editing process. I started the project during the last week of February 2003. Systems Architecture integrated the completed documentation into the online help system at the end of March 2004, so the project lasted 13 months, not including the planning which was completed by my manager before I began working on the project.

The project began to slow down with the first department edit, for which I handed the modules in on May 22, and did not see them again until July 15. For the second edit, I turned the modules in on September 11 and I did not receive them back until November 11. Each edit took nearly 2 months. In my opinion, probably only one edit was necessary, and two months for an edit is too long. If it took one month to write the documentation, it should not take one to edit it. A number of factors allowed this to happen- not only on this project but on a number of department projects during this time period. One factor is the emphasis on preliminary publication. Since writers were only trying to get to the preliminary deadline, they no longer followed up with editors to ensure the editing process went efficiently, and since editors knew the focus was on preliminary publication, they not only took their time in edits, but they began to request optional edits on nearly every document for a large percentage of projects. Another factor may have been the workload of the editors, but this is hard for me to judge- since I didn’t know my editors workload at the time I worked on the letter management project.

The Technical Edit also took a very long time. I sent the modules to the SMEs for Technical Edit on December 19, 2003. I didn’t receive them all back until February 11, 2004. The edit took nearly 2 months, and for almost all of the documents the technical editors didn’t even request a change.

Since I worked on the letter management process, the department has streamlined the editing process. For example, the department no longer requires a peer edit, and for senior employees, the department does not always require a supervisor edit. We also advised editors that they don’t have time to perform a second department edit on most projects. Almost all projects were undergoing two department edits when I worked on the letter management project. Now almost all projects undergo only one. The department also eliminated the technical edit. The Technical
Edit often took a very long time, and almost never resulted in a significant change, and the SMEs had already reviewed the document during the committee review. If these changes had been made before I worked on the letter management process, I could have completed the project in half the time it took. It is easy to imagine how the department’s production is now greatly improved— with each technical communicator’s project similarly more efficient.

**Implement the Solution**

After the letter management documents completed the editing process, I implemented the solution by integrating the documentation into the Artiva Architect online help system and posting a final PDF version of the documentation to the internal intranet. Once I integrated the help into the software, I didn’t have to worry about packaging and delivering it for clients, because it is packaged and delivered along with the rest of the software by other departments in the company.

The implementation process is fairly straightforward for the Technical Communications department; the only time it hinders the production of documentation is when other departments have to perform the integration tasks. For example, the Systems Architecture department integrated the letter management documentation. The table in chapter 3 shows that it took over six weeks for Systems Architecture to integrate the completed help. The tasks included in the integration only take a few minutes.

Since I worked on the letter management project, we have worked to address the integration lag with systems architecture. Systems Architecture now allows Technical Communications to perform some of the integration procedures, which decreases the time to implementation by at least a week.

**Evaluate the Solution**

Since I used the documentation process used by all online help projects at Ontario, I did not conduct a formal evaluation. We do have some methods to evaluate the solution, but nothing systematic we apply to every project. We incorporate improvements to our products and processes by taking suggestions from within the department, from other departments in the company, and from clients. For example, we recently conducted a survey of clients to determine how much they use our documentation and how helpful it is to them. We constantly change and tweak the way we approach documentation and documentation processes, but we could probably improve our documentation by incorporating a more systematic approach toward evaluation. We could solicit feedback from users and internal clients in regular intervals and by asking more specific questions (for example, questions that pertain to a particular project rather than the online help system as a whole).

**Final Thoughts**

A few months after the internship period ended, I became a Supervisor in the Technical Communications department, with the responsibility of managing six technical communicators. My success in the technical communicator role helped show that I could handle the Supervisor position. Since that time (and even before), I’ve been able to have a significant influence on how the department works and the documentation we produce. I’ve been able to research and recommend tools and processes for the department to use, I’ve talked
with clients about our documentation, and I’ve been able to help each of the individuals that report to me be successful in their jobs. I was successful in the technical communicator role from the moment I started at Ontario Systems due to:

- Learning and working quickly.
- My experience with software, programming, and technology in general.
- Managing projects well.
- Working well with others inside and outside of my department.
- Thinking critically about my work
- Ontario’s positive work environment.

The MTSC program greatly improved my ability to learn and work quickly. In the MTSC program we had a tremendous workload and our professors had very high expectations for us, both of which increased my capacity to learn and work quickly (under stress). Because the MTSC program required that we work on many projects at once, and because we were explicitly taught about organization communication and project management, I was able to efficiently manage simultaneous projects with intersecting deadlines the day I walked in the door. Additionally, by taking programming and systems analysis classes to fulfill my MTSC electives, I was able to prepare myself for work in the software industry. I’ve been able to pick up software very easily, and I’ve been able to speak in the same language as many of the SMEs I’ve had to work with. In the MTSC program, I did not like working on so many group projects. It always seemed that someone in the group wouldn’t do their fair share of the work; and sometimes group members didn’t even get along. Now, I understand how the group projects prepared us for the real world, because we have to work with other in the real world as well. The MTSC focus on rhetorical writing has helped me think critically about our solutions and processes, instead of getting stuck in a routine or a template. In the Organizational Communication class, Professor Gary Shulman emphasized finding an organization where you fit, and I think my success at Ontario is also due to the work environment here. I’ve been able to thrive in an environment where I’m respected by the company and coworkers and where I can self-direct my work.

I’ve been able to apply much of what I learned in the MTSC program to my job in the software development industry at Ontario Systems. To be successful and make an immediate impact wherever you choose to go, I give the following advice to MTSC students:

- Take advantage of your electives.
- Be prepared to work hard in the program and in the real world. The MTSC program teaches you to handle a workload bigger than most people you’ll work with can handle.
- Learn from ambiguous MTSC projects (projects are even more ambiguous in the real world).
Appendix A: Technical Communicator Job Description
Ontario Systems, LLC
Job Description

Job Title: Technical Communicator
Department: Technical Communications, Education Services, Client Services Division
Reports To: Supervisor, Technical Communications
Prepared By: Jan Hunt
Prepared Date: January 17, 2003     Revised: January 9, 2004
Approved By: Cindy Ammerman
Approved Date: January 24, 2003

PURPOSE OF THIS POSITION
To research, design, produce, and maintain printed and online technical documentation for all product divisions in a timely manner, and to assist the company with other documentation needs.

ESSENTIAL DUTIES AND RESPONSIBILITIES include the following. Other duties may be assigned. The percentage specified for each essential duty and responsibility is an indicator of the approximate amount of time spent on that particular duty and/or responsibility. It is not however, an indicator of the importance of the duty. The importance of each duty and/or responsibility may change daily dependent upon related circumstances.

1. Create printed and online documentation. (Approximately 60% of job)
   • Research and analyze subject matter, analyze the target audience, and plan content and organization of the final product
   • Coordinate with the supervisor and the writing committee to create and approve a documentation plan
   • Interview subject matter experts from development and other departments, experiment with the product, and read other documentation to research product functionality
   • Develop a strong understanding of document structure, the subject matter, formal grammar rules, and departmental and industry procedural and stylistic standards to write correct, complete, and consistent documents for print or online delivery
   • Communicate extremely technical information in basic, understandable language, striving for a reading level no greater than 8th grade
   • Create screenshots, flowcharts, and other graphics
   • Lay out print documents such as quick reference documents
   • Coordinate with Marketing Communications to create document covers and other graphics
   • Follow the procedures for editing which include: self-editing, peer editing, and ensuring accuracy by having projects reviewed by a subject matter expert
   • Coordinate and lead review meetings comprised of subject matter experts from Application Development, Application Support, Project Management, Training Services, and Technical Communications
   • Publish documents to multiple delivery systems
• Revise and update current and subsequent versions of documentation

2. Manage assigned projects. (Approximately 20% of job)
• Complete multiple projects on intersecting timelines
• Schedule, track, and meet project timeline milestones
• Delegate tasks to other team members and to subject matter experts
• Work cooperatively with team members and other internal and external contacts
• Ensure accuracy, thoroughness, and neatness of final products
• Communicate progress, constraints, and various issues with Manager, Technical Communications, design and development teams, and directors - product management
• Learn software and hardware pertinent to assigned projects
• Solicit feedback from internal and external customers to help improve content and processes

3. Communicate information to team members accurately and professionally. (Approximately 10% of job)
• Communicate information about department and company tools to team members
• Provide examples of information presentation techniques for committee approval and for team members to use as guides
• Assist team members in performing procedures while researching the system
• Assist team members in performing procedures with the single sourcing tools
• Prepare training presentations and/or one-on-one sessions for new team members
• Assist team members whenever needed
• Research and gather updated technical communications information by attending internal and external workshops, classes, and training sessions and extensively using Internet and email resources
• Present new information to the standards committee or other team members as appropriate
• Research and present questions about standards to the standards committee
• Create documents to communicate committee decisions to the department

4. Respond to company requests in a timely manner. (Approximately 10% of job)
• Update and create documentation CD-ROMs
• Send manuals, CD-ROMs, and other documentation as requested
• Review development design documents for opportunities for consistency and adherence to standards
• Answer standards and grammar questions

SUPERVISORY RESPONSIBILITIES
This job has no supervisory responsibilities.

QUALIFICATIONS To perform this job successfully, an individual must be able to perform each essential duty satisfactorily. The requirements listed below are representative of the
knowledge, skill, and/or ability required. Reasonable accommodations may be made to enable individuals with disabilities to perform the essential functions.

**EDUCATION and/or EXPERIENCE**

Bachelor's degree (B.A. or B.S.) from four-year college or university and over one year related experience preferred, and/or equivalent combination of education and experience. Experience and degree in English, technical communications, MIS, Computer Science, Information and Communication Sciences, or related area are preferred. In addition, experience in software documentation for multiple outputs (i.e. training materials, CD-ROM, help files, user guides, quick references) for complex software is beneficial. Knowledge of computer-related terminology, desktop publishing, editing, online help, single sourcing, layout programs, and PC operating systems, software and hardware is required. Knowledge of XML or SGML, XSL, DTDs, and basic programming is preferred.

**LANGUAGE SKILLS**

Ability to read, analyze, and interpret general business periodicals, professional journals, technical procedures, or governmental regulations. Ability to write reports, business correspondence, and procedure manuals. Ability to effectively present information and respond to questions from groups of managers, clients, customers, and the general public.

**MATHEMATICAL SKILLS**

Ability to calculate figures and amounts such as discounts, interest, commissions, proportions, percentages, area, circumference, and volume. Ability to apply concepts of basic algebra and geometry. Ability to apply concepts such as fractions, percentages, ratios, and proportions to practical situations.

**REASONING ABILITY**

Ability to define problems, collect data, establish facts, and draw valid conclusions. Ability to interpret an extensive variety of technical instructions in mathematical or diagram form and deal with several abstract and concrete variables.

**CERTIFICATES, LICENSES, REGISTRATIONS**

Attain Ontario Systems classes and/or certifications as required
Attain external classes and/or certifications as required

**PHYSICAL AND EMOTIONAL DEMANDS**

While performing the duties of this job, the employee is regularly required to sit; use hands and fingers to operate a computer keyboard, mouse, and telephone keypad; and talk or hear. The employee is occasionally required to stand; walk; reach with hands and arms; and stoop, kneel, crouch, or crawl. The employee must occasionally lift and/or move up to 25 pounds. Specific vision abilities required by this job include close vision, distance vision, and color vision. The employee must be able to see and use a color computer monitor and keyboard for detailed written and graphics work. The employee must be able to lift boxes of manuals for storage and distribution as needed.
The employee is required to have emotional stability to deal with stress from the need to meet tight deadlines and the ability to deal with confidentiality of passwords and financial information related to Ontario Systems’ business. This position involves constant constructive criticism of one's work; therefore, the employee must be able to accept this criticism in an emotionally calm manner.

WORK ENVIRONMENT
The work environment characteristics described here are representative of those an employee encounters while performing the essential functions of this job. Reasonable accommodations may be made to enable individuals with disabilities to perform the essential functions.

While performing the duties of this job, the employee is occasionally exposed to risk of electrical shock.

The noise level in the work environment is usually moderate.

Nonstandard work hours are required occasionally. Early shift, late shift, and weekend hours may be required.

Travel time varies based on assigned projects.

CONTACTS The following is a list of internal and external contacts with which this position is required to communicate. The frequency of the contacts is also noted.

Internal

• Education Services Group Management & Staff (Daily or As Needed)
  ➢ Technical Communications Department
  ➢ Training Services Department
  ➢ Knowledge Engineers
  ➢ Administrative Assistants
• Subject matter experts including Systems Analysts, Project Consultants, and Application Consultants (Daily or As Needed)
• Marketing Communications Department Staff (Weekly or As Needed)
• Information & Technology Group Staff (Occasionally or As Needed)
• Directors - Product Management (Occasionally or As Needed)

ADDITIONAL SKILLS The skills listed below include those that are needed to be successful in this position.

• Ability to analyze information
• Ability to gain a basic, functional understanding of Ontario Systems’ programming (Caché) and operating systems (UNIX) levels
- Ability to handle multiple projects simultaneously
- Ability to learn basic programming concepts
- Ability to work efficiently without direct supervision
- Ability to work with all personality types
- Excellent verbal and written communication skills
- Familiarity with online help and online help tools
- Familiarity with online publishing tools such as Acrobat Distiller, Acrobat Reader, and Microsoft FrontPage
- Familiarity with single sourcing
- Familiarity with use of desktop publishing applications such as Adobe FrameMaker, Microsoft Word, and Adobe PageMaker
- Familiarity with XML or SGML concepts and tools
Appendix B: Letter Management Documentation Plan
Letter Management Plan (part of CT Vision Production Plan)
Development SME – (Developer SME)

Letter Management

Letter Management Overview
What Is Letter Management?

*Discuss how you can use events, scripts, letter form types, and shared variables to control letter processing. Discuss letter batches. Discuss using MS Word as your letter editor.*

How to Plan a Letter
Overview: How to Create a Letter
*Includes letter form types, letter definition, letter body*

Microsoft Word Letters
Why Use Word as Your Letter Editor

*Mention that you can use the SYWORD (GUI only) and SYLETTER service. Mention that you can merge fields and the naming conventions for using a field, shared variable, letter phrase, or script as a merge field. Cross-reference the term letter variables as merge fields (FACS terminology).*

How to Use Microsoft Word as Your Letter Editor

- Install plug-in
- CT Term version 4.0 or higher
- (Optional) Set the default directory for templates for all users to access

Letter Form Types
What Is a Letter Form Type?
Why Use a Letter Forms Type
How to Create a Letter Form Type

Letter Form Types Report
What Is the Letter Form Types Report?
Why Run the Letter Form Types Report
How to Run the Letter Form Types Report
**Letter Definitions**

What Is a Letter Definition?
You can use a script (L or G type) to include records from a child table in a letter. For example, you can include a payment history. You can use a script to format the date in the letter. You can use letter services.

You can use Microsoft Word as your letter editor (link).

Mention that you can run the Letter Definitions Report.

Why Use a Letter Definition
How to Create a Letter Definition
What Is the Letter Body?
How to Create a Letter Body
How to Create a Letter Body for a Microsoft Word Letter
How to Set Margins and Page Size in a Letter Definition
How to Set Margins and Page Size for a Microsoft Word Letter
How to Add a Field to a Letter Body
Include naming convention.

How to Add a Script to a Letter Body
Include naming convention.
How to Add a Script to a Letter Body for a Microsoft Word Letter
How to Add a System Variable to a Letter Body
How to Add a System Variable to a Letter Body for a Microsoft Word Letter
How to Add a Shared Variable to a Letter Body
Include naming convention.

How to Add a Shared Variable to a Letter Body for a Microsoft Word Letter
How to Add a Letter Phrase to a Letter Body
Include naming convention.

How to Add a Letter Phrase to a Letter Body for a Microsoft Word Letters

**Letter Definitions Report**

What Is the Letter Definitions Report?
Why Run the Letter Definitions Report
How to Run the Letter Definitions Report

**Letter Events**

What Is a Letter Event?
You can set a letter event on the table and the letter. The letter overrides the setting on the table.

Why Set a Letter Event
How to Set a Letter Event on a Table
How to Set a Letter Event on a Letter

**Letter Phrases**

What is a Letter Phrase?
Why Use a Letter Phrase
How to Create a Letter Phrase
How to Add a Field to a Letter Phrase
How to Add a Script to a Letter Phrase
How to Add a Printer Control Code to a Letter Phrase
How to Add a Shared Variable to a Letter Phrase
How to Add a System Variable to a Letter Phrase
How to Add a Letter Phrase to a Letter Phrase
How to Add a Letter Phrase to a Letter Body

**Letter Phrases Report**
What Is the Letter Phrases Report?
Why Run the Letter Phrases Report
How to Run the Letter Phrases Report

**Letter Batch Process**
What Is the Letter Batch Process?
When you create a batch, the system gathers letter requests based on selection criteria for the batch and archives batch information so that you can rebuild the batch if needed.
Why Process a Letter Batch
Overview: How to Process Letters in a Batch
How to Create a Letter Batch
How to Print a Letter Batch
How to Delete a Letter Batch
(1,5,4)
How to Delete Letter Batches by Date

**Letter Batches Summary Report**
What Is the Letter Batches Summary Report?
Why Run the Letter Batches Summary Report
How to Run the Letter Batches Summary Report

**Letter Batches Detail Report**
What Is the Letter Batches Detail Report?
Why Run the Letter Batches Detail Report
How to Run the Letter Batches Detail Report
Requesting Letters
How Requesting a Letter Affects the System
   Explain all the ways you can request a letter. Requesting a letter from the account screen or from a script can bypass the batch process.
Where to Request a Letter

Menu or account screen
How to Request a Letter
How to Request a Letter from the Account Screen
This procedure bypasses the batch process.
How to Preview Letter Requests
Menu (1,5,8)
How to Delete Unprinted Letter Requests

Updating Letter Batches
How Updating a Letter Batch Affects the System
Why Update a Letter Batch
When to Update a Letter Batch
How to Update a Letter Batch

Rebuilding Letter Batches
How Rebuilding a Letter Batch Affects the System
Why Rebuild a Letter Batch
   Cannot reprint letters after update
When to Rebuild a Letter Batch
After update and before you delete the batch.
How to Rebuild a Letter Batch

Letter Addresses
What Is a Letter Address?
Why Use a Letter Address
How to Create a Letter Address

Letter Statistics
What Is the Letter Statistics Report?
Why Run the Letter Statistics Report
How to Run the Letter Statistics Report

Glossary Terms
Mail merge
Appendix C: SME Email Conversation about Letter Addresses
-----Original Message-----
From: (Developer SME)
Sent: Thursday, April 10, 2003 7:22 AM
To: Matt Troy
Subject: RE: Letter Addresses

In most cases it probably is just as easy to insert a letter phrase into the letter. I think that by using the SYADDRESS script and the Letter address for the letter is changed, then the user doesn't have to also change the Letter Phrase in the Letter body since the SYADDRESS script will return whatever Letter Phrase is on the Letter Address. I am certain there is definitely other reasons that this is used rather than inserting the Letter Phrase into the letter. You might talk to whomever you were talking to in Orange to see why they used the SYADDRESS script. I hope this helps.

Thanks,
(Developer SME)
Application Developer
Ontario Systems

-----Original Message-----
From: Matt Troy
Sent: Wednesday, April 09, 2003 5:55 PM
To: (Developer SME)
Subject: RE: Letter Addresses

About the SYADDRESS script... is there really any advantage to using it... because it seems like it would be easy enough just to insert the letter phrase directly into the letter body.
Thanks!
Matt

-----Original Message-----
From: (Developer SME)
Sent: Friday, March 28, 2003 4:30 PM
To: Matt Troy
Subject: RE: Letter Addresses

In order to print the letter phrase that is on the letter address, you need to use the SYADDRESS script. This script will use the Letter address set for the Letter and return the letter phrase specified on the Letter Address. If you don't use the SYADDRESS script to retrieve the letter phrase from the Letter Address, than if you want an address on your letter you would either have to add the fields directly to the letter or insert the letter phrase in yourself. Here is an example of a letter...

<<$$SYADDRESS>>
Dear <<DTFIRST>> <<DTLAST>>,
Please pay your bills.

Sincerely,
<<SYUSERID>>

Where the $$SYADDRESS is inserted is where the letter phrase will be returned. In
the case above, the letter address would be inserted at the top of the letter. If you do
not use the $$SYADDRESS in the body of the letter, you would either (a). Not insert a
letter address in the letter or (b). Insert the Letter phrase directly into the letter body.
Let me know if you have any questions.

Thanks,
(Developer SME)
Application Developer
Ontario Systems

-----Original Message-----
From: Matt Troy
Sent: Friday, March 28, 2003 3:06 PM
To: (Developer SME)
Subject: RE: Letter Addresses

Hey (Developer SME):
I was wondering if you could give me an example of how the letter phrase prints a
letter address.

I guess you have to put the letter phrase you set in the letter address in the letter
definition. The phrase has to have the same fields in it that you define in the Address
Field Information window on the Letter Address. The phrase prints just like a normal
phrase, except that it's pulling the fields from the address- b/c the address might have
a different table?

Thanks!
Matt

-----Original Message-----
From: (Developer SME)
Sent: Thursday, March 20, 2003 2:22 PM
To: Matt Troy
Subject: RE: Letter Addresses

Matt,
I was able to research and track down the following answers for you. Unfortunately, I
wasn't able to get gather much information on the questions in red. The problem for
me is giving good examples since I rarely get to see how the client actually uses
something. What I hope you can do in this instance is find someone who has been able to use this functionality and can give you good reasons and examples about using the Letter Addresses. I would suggest talking to someone in Orange. I believe they asked for this functionality so I am sure that they can give you some good examples.

1. This is definitely a question for Orange. I have been told they use this functionality.
2. You need to specify a letter phrase because if the Letter Address is printed on a letter whatever is in the Letter Phrase will be used to print the address.
3. The sort on the Letter Address will sort the letters in a batch with that letter address according to whatever field is specified for the sort. For example, in Letter Batch 1, you have two letters with Letter Address A and Letter Address B. On Letter address A, your sort field is the first name. So this is how the letters will be sorted...
   -Letters with Letter Address A
     First Name: (Developer SME)
     Cindy
     Lisa
     Matt
   -Letters with Letter Address B

4. The prompt field will only show if you request the letter from the window on the account screen. The address prompt doesn't display in any other instance.
5. Yes, if you prompt the user for information it will override the address you specify in the Default Address field in the Address Information window.
6. If you specify a Shared Variable window on the letter definition, you can basically use it to prompt for information you want to put in a letter. For instance if you have the shared variable %BAL as a field in your window and that field also in the body of your letter whatever value the person enters for %BAL would be inserted in the letter body. You could also prompt for a value to be used in a script.
7. Unfortunately, I don't have a good reason to use Letter Addresses. This will probably a question for Orange.

Please let me know if you have any questions. Sorry I couldn't come up with anything on the other questions.

Thanks,
(Developer SME)
Application Developer
Ontario Systems

-----Original Message-----
From: Matt Troy
Sent: Friday, March 14, 2003 10:38 AM
To: (Developer SME)
Subject: Letter Addresses
Hi (Developer SME):
Happy Friday! Here are my final Letter Mgt questions for the week. We're getting through these pretty quickly. After Letter Address, we only have Letter Batches and Word letters to go. Thanks for all your help! Take your time...

1. What are address Set Fields for? Why do we need them?

2. Why do you need to specify a letter phrase (General Information on Letter Addresses) in which to put the address? Does the address always actually print in the phrase? Does the address not appear on a letter, unless you put it in a phrase (with all address fields present in the phrase)?

3. What is an example of how you would use the sort field (General Information on Letter Addresses) on a letter address?

4. When you set the Prompt for Address field in the Address Information window on a letter definition- does this cause the prompt to appear for every way you can print a letter (batch, account screen, script)?

5. If you prompt the user to enter an address when a letter prints, does the address they enter override the address you specify in the Default Address field in the Address Information window on the letter definition?

6. What is the Shared Variable Window field for in the Address Information window on the letter definition? What does the window do, and why do you need it?

7. Is the main reason we need letter address to print the address on a letter definition for envelops? Do users make letter definitions for envelops?

Thanks!

Matt Troy
Technical Communicator
Ontario Systems
Tel: (765) 751-7000
Fax: (765) 751-7499
Appendix D: Department Edit Form
Department Edit Form

Use the Department Edit Form for all projects. Editors must initial and date the form for each edit.

Project Manager: ____________________________ Host Name: ____________________________
Special Considerations: ____________________________ Host Port: ____________________________
Task Number: ____________________________ Namespace: ____________________________

Audience: ____________________________

Supervisor Preliminary Edit

Submit the draft and this form to your supervisor for a preliminary edit. Also include the task request and supporting documents, such as the content plan, investigation notes, and emails.

Product Document Type

<table>
<thead>
<tr>
<th>Title</th>
<th>Pages</th>
<th>Assign Date</th>
<th>Due Date</th>
<th>Initials</th>
<th>Return Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Committee Review

Schedule a committee review. You should invite your supervisors, subject matter expert, application support representative, training representative, and editor to this meeting.

☐ Complete   Date ____

First Edit

Submit the draft and this form to the first editor for a first edit. Also include the task request and supporting documents, such as the content plan, investigation notes, and emails.

Product Document Type

<table>
<thead>
<tr>
<th>Title</th>
<th>Pages</th>
<th>Assign Date</th>
<th>Due Date</th>
<th>Initials</th>
<th>Return Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Task:**

**Final Edit**
Your supervisor performs the final edit for your project. Submit the total project, this completed form, and all previous edits.

<table>
<thead>
<tr>
<th><strong>Product Document</strong> Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Title</strong></td>
</tr>
<tr>
<td>--------------</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

**Final Edit Comments:**

**Optional Edit**
If the project requires extensive changes after the first edit, submit the draft and this form to the first editor. Also include the task request and supporting documents, such as the content plan, investigation notes, and emails.

<table>
<thead>
<tr>
<th><strong>Product Document</strong> Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Title</strong></td>
</tr>
<tr>
<td>--------------</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Appendix E: Technical Editor Job Description
Ontario Systems
Job Description

Job Title: Technical Editor
Department: Technical Communications, Education Services, Administration Division
Reports To: Supervisor, Technical Communications
Prepared By: Jan Hunt
Prepared Date: January 8, 2003
Approved By: Cynthia Ammerman
Approved Date: January 24, 2003

PURPOSE OF THIS POSITION
To edit technical and non-technical documentation for all product divisions.

ESSENTIAL DUTIES AND RESPONSIBILITIES include the following. Other duties may be assigned. The percentage specified for each essential duty and responsibility is an indicator of the approximate amount of time spent on that particular duty and/or responsibility. It is not however, an indicator of the importance of the duty. The importance of each duty and/or responsibility may change daily dependent upon related circumstances.

1. Edit printed and online documentation. (Approximately 65% of job)
   - Edit content for grammar, spelling, word usage, completeness, voice, clarity, and readability
   - Ensure technical accuracy of the content by learning and testing the subject matter
   - Determine suitability of content for target audience
   - Determine suitability of content for delivery media (print, web, help, etc.)
   - Rewrite or modify content when appropriate and discuss changes with author
   - Provide tactful, constructive, and appropriate feedback to writers by various methods of communication

2. Maintain writing standards. (Approximately 15% of job)
   - Create, maintain, and adhere to writing and editing procedures, standards, and templates
   - Maintain an editing project workflow process
   - Obtain feedback from authors regarding current and proposed standards
   - Ensure company-wide compliance with application of standards
   - Consider exceptions to standards on a limited basis and discuss/determine whether changes are needed
   - Ensure usability for single sourcing where applicable
   - Work cooperatively with division representatives to create and communicate standards

3. Manage assigned projects. (Approximately 10% of job)
   - Schedule, track, and meet project milestones and deadlines
• Delegate tasks to other team members and to subject matter experts when appropriate
• Work cooperatively with team members and other internal and external contacts
• Work cooperatively with management to create and adhere to timelines
• Ensure accuracy, thoroughness, and neatness of final products
• Communicate progress, constraints, and issues with the manager, design and development teams, and product coordinators
• Solicit feedback from internal and external customers to ensure satisfaction
• Submit status and time reports to supervisor in the requested format

4. Train and mentor fellow team members. (Approximately 5% of job)
   • Prepare complete, effective training presentations and one-on-one training sessions for new team members
   • Provide ongoing tips and training on writing, editing, and compliance to standards
   • Assist team members whenever needed

5. Complete writing projects. (Approximately 5% of job)
   • Prepare documentation for short turn-around writing projects as needed
   • Prepare departmental documentation for writing and editing staff as needed (for example, writing guidelines)

SUPERVISORY RESPONSIBILITIES
This position has no supervisory responsibilities.

QUALIFICATIONS To perform this job successfully, an individual must be able to perform each essential duty satisfactorily. The requirements listed below are representative of the knowledge, skill, and/or ability required. Reasonable accommodations may be made to enable individuals with disabilities to perform the essential functions.

EDUCATION and/or EXPERIENCE
Bachelor's degree (B.A. or B.S.) from four-year college or university; or one to two years related experience and/or training; or equivalent combination of education and experience. Experience and degree in English, Technical Communications, Computers, or related area are preferred. Knowledge of computer-related terminology, desktop publishing, editing, online help, layout programs, PC operating systems, and software and hardware is required. Experience in a field related to editing may apply as a substitute for the previous requirements.

LANGUAGE SKILLS
Ability to read, analyze, and interpret general business periodicals, professional journals, technical procedures, or governmental regulations. Ability to write reports, business correspondence, and procedure manuals. Ability to effectively present information and respond to questions from groups of managers, clients, customers, and the general public.

MATHEMATICAL SKILLS
Ability to calculate figures and amounts such as discounts, interest, commissions, proportions, percentages, area, circumference, and volume. Ability to apply concepts of basic algebra and geometry. Ability to apply concepts such as fractions, percentages, ratios, and proportions to practical situations.

REASONING ABILITY
Ability to define problems, collect data, establish facts, and draw valid conclusions. Ability to interpret an extensive variety of technical instructions in mathematical or diagram form and deal with several abstract and concrete variables.

CERTIFICATES, LICENSES, REGISTRATIONS
Attain Ontario Systems certifications and/or classes as required
Attain external certifications and/or classes as required

PHYSICAL AND EMOTIONAL DEMANDS
While performing the duties of this job, the employee is regularly required to sit; use hands and fingers to operate a computer keyboard, mouse, and telephone keypad; and talk and hear. The employee is occasionally required to stand; walk; reach with hands and arms; and stoop, kneel, crouch, or crawl. The employee must occasionally lift and/or move up to 25 pounds. Specific vision abilities required by this job include close vision, distance vision, and color vision. The employee must be able to see and use a color computer monitor and keyboard for detailed written and graphics work.

The employee is required to have emotional stability to deal with stress from the need to meet tight deadlines, as well as the ability to deal with confidentiality of passwords and financial information related to Ontario Systems’ business. This position involves constant constructive criticism of one's work; therefore, the employee must be able to accept this criticism in an emotionally calm manner.

WORK ENVIRONMENT
The work environment characteristics described here are representative of those an employee encounters while performing the essential functions of this job. Reasonable accommodations may be made to enable individuals with disabilities to perform the essential functions.

While performing the duties of the job, the employee is occasionally exposed to risk of electrical shock.

The noise level in the work environment is usually moderate.

Nonstandard work hours are required occasionally. Early shift, late shift, and weekend hours may be required.

Travel time is required occasionally.
CONTACTS  The following is a list of internal and external contacts with which this position is required to communicate. The frequency of the contacts is also noted.

Internal
• Marketing Staff (Daily)
• Education Services Group Management & Staff (Daily or As Needed)
  ➢ Technical Communications Department
  ➢ Training Services Department
  ➢ Knowledge Engineers
  ➢ Administrative Assistants
• Subject matter experts including Systems Analysts and Application Consultants (Weekly or As Needed)
• Other Ontario Systems’ employees (As Needed)

External
• Clients (As Needed)

ADDITIONAL SKILLS  The skills listed below include those that are needed to be successful in this position.

• Ability to analyze information and solve problems
• Ability to apply standards consistently
• Ability to apply subject matter from the reader's perspective
• Ability to handle multiple projects simultaneously
• Ability to learn new tools and concepts quickly
• Ability to learn software and test procedures
• Ability to vary work style based on the criteria for each situation
• Ability to work efficiently without direct supervision
• Ability to work with all personality types
• Excellent negotiating skills
• Excellent written and verbal communication skills
• Familiarity with online help
• Familiarity with single sourcing, desktop publishing tools, and XML concepts
• Strong attention to detail
Appendix F: Technical Edit Checklist
## Technical Edit Checklist

The technical editor uses this checklist for the technical review of all projects. The technical editor must sign and date this checklist.

<table>
<thead>
<tr>
<th>Type of Project:</th>
<th>Date Submitted:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Title:</td>
<td></td>
</tr>
<tr>
<td>Tech Editor:</td>
<td></td>
</tr>
<tr>
<td>Due Date:</td>
<td></td>
</tr>
<tr>
<td>Return to:</td>
<td>Proj. #</td>
</tr>
</tbody>
</table>

**Important Reminder**: Legally, Ontario Systems must support what it documents. Ontario Systems acquires the obligation to make the system comply with the documentation. Unless specifically documenting an add-on product, we document only standard features. To ensure correct documentation, you need to verify by testing, conducting the procedures, and reading code. Please list errors, corrections, and suggestions on the pages within the project.

Some of these sections might not apply:

### Introduction:
- [ ] Appropriate title
- [ ] Correct introductory information
- [ ] Correct prerequisites for the topic
- [ ] Correct project or investigation number

### Concepts:
- [ ] Technically correct conceptual information
- [ ] No custom features shown or documented
- [ ] Thoroughly and sufficiently answer questions users might ask
- [ ] Realistic examples and screen captures
- [ ] Enough examples (if more are needed, please provide them)

### Graphics:
- [ ] Correct screen captures
- [ ] Correct illustrations that relate to realistic scenarios
- [ ] Acceptable graphics for user
- [ ] Sufficient number of screen captures or other graphics

### Charts:
- [ ] Technically correct (especially variable names, definitions, emulations)
- [ ] Flow charts have a logical beginning and end so clients understand flow
- [ ] Flow charts enhance understanding

### Procedures: I have performed all the steps in all procedures. __________ (Please initial)
- [ ] Appropriate procedure for a knowledgeable or novice user
- [ ] Accurate and technically correct procedures
- [ ] All steps included
☐ APPROVED
☐ APPROVED, Pending Changes
☐ NOT APPROVED

Sign and date below.

Sign and date below.

Please make changes indicated and return for final approval.

Date:

(Technical Editor’s Signature)

(QCA Reviewer’s Signature) Date: