

northbay news

The monthly newsletter of the NorthBay Chapter of the
Society for Technical Communication

Volume 9, Number 3, July-August 2002

How to Get a Job and Keep It

John Underwood, NorthBay Chapter



As a technical writer in these difficult economic times, you may be unemployed or underemployed. If you are fortunate enough to have a job, you may be concerned about a layoff. Although the employment situation for technical communicators is improving, it is still a long way from the happy days of early 2001. The skills necessary for finding and keeping a job have never been more important, according to Kerry Bourns, who addressed the June meeting of the NorthBay Chapter in her presentation titled “How to Get and Keep a Job.”

Bourns holds a number of departmental positions at Empire College in Santa Rosa, as well as teaching 19 different classes. She also works with the college’s placement department to help more than four hundred students develop their careers and find jobs.

Bourns’s presentation had two parts. The first, getting a job, covered looking for a job and what she called impression management: résumés, cover letters, and interviews. The second, keeping a job, went into the characteristics that employers want to see in their employees, and interestingly enough, the characteristics of a good employer.

Continues on page 2

In This Issue

How to Get a Job & Keep It 1

XML Solutions 1

A Conversation 4

News Briefs 9

*Professional Development
Opportunities 9*

XML Solutions

Elyse Lord, northbay news Editor

At the May chapter meeting, Toby Wraye told a rapt audience how his Niku team produced context-sensitive XML Help files and portable document files (PDF) from a single FrameMaker file.

The presentation followed up on April 2001 and September/October 2001 *northbay news* reports of the Niku team’s innovative work.

Wraye described the general challenges his team faced, which included the following:

- ♦ Writers wanted a WORM (Write Once, Reuse Many) solution that would take input from FrameMaker and produce output in multiple formats. Such a solution seemed to require a prohibitively expensive database.

Continues on page 8

**northbay officers
and committee chairs****president**

chris muntzer
chris_muntzer@agilent.com

co-vice-presidents

rolfe dlugy-hegwer
rdlugy-hegwer@truetime.com
trudie folsom
tfolsom@sonic.net

newsletter

editor: elyse lord
elord@pon.net
layout: ancilla allsman
allsman@attbi.com
copy editor: geneviève duboscq
gduboscq@sonic.net

web

trudie folsom
webmaster@stc-northbay.org

membership

barbara herbert
barbara@sonic.net

hospitality

michele green
m_green01@yahoo.com
ron douglass
rsdouglass@att.net

treasurer

liz kaiser
ehkaiser@sonic.net

submitting articles

We welcome articles and news about meetings, workshops, and courses that pertain to technical communication. Please e-mail simple text to the newsletter editor.

reprints and distribution

If you reprint articles from the *northbay news*, please credit them and forward a copy to the editor. Reprints in non-STC publications are subject to the author's approval.
Copyright © 2002 *northbay news*.

STC Mission Statement

The mission of the Society for Technical Communication is to improve the quality and effectiveness of technical communication for audiences worldwide.

How to Get a Job and Keep It

Continued from page 1

How to Get a Job

Obviously, the first thing you must do is to find out what jobs are available. The traditional methods of looking at the newspaper want ads or sending your résumé to a company that is not advertising for help (called direct application) are not very useful. Companies receive from 70 to 200 résumés per ad; the chances of your application standing out are slim. Direct application rarely works.

More effective methods are using the Internet and networking. If you are looking for work, you should post your résumé on job sites such as Dice.com and make sure the information is current. However, networking is probably the best way to find a job. Employers find 80 percent of their new job candidates from referrals by their employees. You should keep in touch with former coworkers and attend meetings of professional groups such as STC. (See the July/August 2001 *northbay news* article, "Networking in the New, New Economy," for networking ideas.)

Informational interviewing (asking employees or hiring managers who work at desirable companies to describe what they do and what sorts of skills they look for in their employees) can be quite helpful, too. The purpose of this type of interview is not to get a job (although that may happen) but to gather information that will be helpful in your job search—and to make a good impression should a job open up down the road.

Once you have found an open position, you need to get your résumé and cover letter ready and plan for your hoped-for interview. The key to successful résumés, cover letters, and interviews is to tailor them. And the key to tailoring is preparation. Research the company you are applying for. Get information on it from the Web and trade magazines, for example. Write your résumé and letter as if this were the only job you wanted. Sound interested and knowledgeable. Use the words and phrases the company uses to describe the position. This technique also makes your résumé stand out if resumes are going to be screened electronically.

Preparing the résumé and cover letter seems straightforward enough, but how should you prepare for the interview? Be ready for two types of questions. First are the so-called behavioral questions such as "How do you deal with a difficult coworker?" Next, be ready with some responses when the interviewer asks you if you have any questions. You should. Two good ones are "What will the focus of my work be for the first six months?" and "How did this position become available?"

How to Keep a Job

All right, you have a job now—no mean feat these days. What should you do to keep it? Bourns listed six characteristics employers want in their workers.

Continues on page 3

How to Get a Job and Keep It

Continued from page 2

1. Be enthusiastic and diligent. If you get a task at work that seems trivial, do not moan about it and do the minimum. Act, or even better, be enthusiastic about the assignment and do the best you can.
2. Be consistent. Your boss should not have to wonder if this is the right day and hour to give you a job because you might be grouchy.
3. Show respect and loyalty. An important part of this characteristic is to avoid talking behind a coworker's back. Not only is this mean, it can come back to haunt you.
4. Be solution oriented. When there is a problem at work, try to think of a way to solve it.
5. Take initiative. Do not wait to be told to do something you know should be done.
6. Be a team player. Cultivate your communication skills, and be willing to work with all sorts of personalities. This is more important than ever. Employers list it as one of the most desired qualifications.

Work is a two-way street, and Bourns also identified the following characteristics as being desirable in a manager or management team:

- ♦ Supervisors do not give orders, and they always remember that their employees are people.
- ♦ Employers acknowledge the good work of their employees and let them make decisions.
- ♦ Supervisors use the PNP principle. They start out with praise; then they communicate the negative, critical part; finally, they praise the employee again. Also, they criticize only in private.
- ♦ Supervisors set a good example by pitching in.
- ♦ Supervisors do not blow their top in stressful situations. And if they do, they quickly apologize.
- ♦ Supervisors do not take the credit for success; they publicly give credit to the worker who did the job.



Some Final Thoughts

Bourns and audience members touched on some interesting points during the question and answer period. The technical writing profession demands more versatile people now. You can use these rough times to upgrade your job skills and increase your marketability by learning Web design or computer science, for example. On an optimistic note, some in the audience said that the late 1990s trend of outsourcing technical writing appears to be ending. Employers now want full-time writers on-site.

Bourns ended by wishing the audience good luck. If you follow some of her recommendations, perhaps you can create some good luck in your technical writing career.

A Conversation with Macromedia's Barbara Herbert

In February, *northbay news* editor Elyse Lord met with Barbara Herbert, the NorthBay Chapter's membership coordinator, to discuss Herbert's career as a technical writer. After about eighteen years of work as a contract writer—with clients like GE Information Services, Borland, Intuit, and Alcatel—Herbert accepted a permanent position with Macromedia, Inc., in the fall of 2000.

One of Herbert's responsibilities at Macromedia is to document application programming interfaces (APIs). An API document helps software developers understand how two software components communicate with each other. (See the July/August 2001 *northbay news* for more information on documenting APIs.) Below are excerpts from this conversation.

How did you land clients in your work as a contract technical writer?

Most of my work has come from networking at STC meetings or from colleagues. I've also worked with a couple of different agencies. I once found somebody who was maintaining a list of all of the agencies in Northern California. So I got the list, put it in an Excel spreadsheet, and sent everybody my résumé. And so I got some leads from that.

Actually, I got the lead for the Alcatel contract at an STC meeting. When my contract at Alcatel ended after about a year, I got the Intuit and the Macromedia contracts. In both those cases, I was recommended by friends who had been contacted first but who were too busy to take the work. Around that time I also decided I wanted to make more money, so I bid on the entire project, rather than by the hour. I work really fast, so getting paid by the hour kind of penalizes me for being efficient. I heartily recommend bidding on a project basis for anyone who can do it.

I liked the freedom of contracting, but the problem was I'd make a lot of money and then I wouldn't have any work for four months or something. My marketing skills just weren't very good. And after a while that just got old. I happened to hear that Macromedia was looking for a full-time person. And since I had already done some contracts for them off-site, working off-site almost fulltime as an employee wasn't a problem. So I applied, interviewed, and got the job.

And you're not under pressure to spend more time on-site?

Not at all, surprisingly enough. I thought I might be.

I guess if Macromedia's applications have an open architecture, the company is probably not so concerned about intellectual property issues?

Well, it's not that. I mean, I have to connect over a secure network to get through the firewall and everything. I have a shredder at home. I

Continues on page 5

Barbara Herbert

Continued from page 4

think it's more the fact that they're an Internet company to start with. So it would be kind of silly not to try to use the technology. And I was hired at the height of the boom, when there was lots more work than there were people to do it. So I kind of was in a position to say, "I'll only come in one day a week, maybe two days now and then." I guess at that point Macromedia in particular was looking for someone with this highly technical kind of skill set, and they already knew my work, so they accepted my "conditions."

It's interesting that you say "highly technical." In listening to you talk about your background, I'm getting a sense that you're a strong communicator, a good editor, are detail oriented, and work well without supervision. And yet you're focusing on the technical, which may be what hiring managers really want. The other stuff they just expect.

Any good technical writer has to have the other stuff. I mean, it would be a requirement for a senior technical writer to have those qualifications. At the time I was hired, Macromedia needed writers for a kind of a client-server type product, so they were especially interested in someone who had some network or programming experience. Plus they liked me and I liked them.

What was one of the more rewarding projects you worked on as a contractor?

I loved the dBASE work [at Borland], just because I've always loved dBASE and was a dBASE programmer before I became a technical writer. It was also really neat to have a book that came out with Quicken—which won an award [Touchstone]. It was just a little 60-page *Getting Started* guide, but just knowing that so many people were going to read this book was kind of neat.

All the projects are rewarding in their own ways. They're all different. Like the one where I had to take the technical support documents and turn them into customer support documents. That was rewarding because I knew that these poor people (the customers) were turning to these documents because they had a problem—they needed help. So now they could get it.

For over a year, I've been working on the Macromedia Flash Communication Server MX, which lets developers use a new client-server protocol to add communication and collaboration capabilities to their Flash applications. Because I was the first of several writers eventually assigned to this product, I've had the opportunity to help shape this product's document set from its inception, and also to provide input into product features as they were being designed, all of which has been very rewarding.

*All the projects
are rewarding
in their own
ways.*

Continues on page 6

Barbara Herbert

Continued from page 5

What are some of the challenges that are keeping you interested in the API documentation you're doing at Macromedia?

Most of the API documentation tends to be focused on the language reference, because developers have to know what the classes are and what the methods are. But it's also nice to give programmers an overview of "Here's why it works the way it does. Here's a sample application that does this or that. Look how we've put this thing together." So that's kind of along the lines of tutorials and procedures and scenarios which, for me anyway, are much more difficult to write.

Given the enormity of the projects and the fast pace and so on, have you found that developing the processes for documenting APIs is more complicated than developing the processes for developing other types of documentation?

Documenting end-user software is easier in some ways. You run the program. You say, "How do I do this?" "How do I do that?" You have a good idea of how the end user is going to use the product. Well, with an API, you don't even know what to ask. "How do I do" what? The end user (a programmer) can do basically anything with the product. So it is more difficult in that sense.

I'm curious about the timing on a programmer's guide that includes tutorials. What would a typical deadline be like for a project of that scope?

It probably varies depending on whether it's the initial product release or an upgrade. For the initial release, if you count from the time they start to gather the specs (assuming you're around) and the product is being developed and tested, until the time it actually goes out, a year is probably reasonable.



I think when I just did the dictionaries [reference guides], they took a couple of months. But that's just because the other products were already established in the market, so tutorials weren't considered critical. For a new product that's larger in scope—if you don't give [customers] those books, the project will fail. Then the project takes much longer.

That sounds like a lot of pressure.

Yes, but it's nice to know that people are actually reading these documents. Like this product I'm working on now went out to an alpha test. And so I was subscribing to the alpha test list, and people were saying, "I tried to download the dictionary, and I couldn't get it." It turns out the dictionary had been corrupted, and I had to reupload it, and I was so excited. Wow, these people really need my book! They can't use this product without it. So, to actually get real direct feedback from the user. Do you know how rare that is?

Continues on page 7

Barbara Herbert

Continued from page 6

It sounds like you're getting the opportunity to get that feedback in part because of this listserv.

It's an internal list for which all subscribers have signed a non-disclosure agreement, but yes, there's a list. So developers send their questions to the list, and people help each other. And if it's an actual engineering question, one of the engineers or QA people will answer. Sometimes I answer. Very neat.

I'm really struck listening to you talk by your emphasis on the end-user and by the excitement that you have by the whole idea of actually being read.

There are a lot of people who don't read the documentation, but I'm not writing for them. I always try to empathize with the person who's trying to use this product. We all know that most products are notoriously poorly-designed. Somebody's got to help the users figure it all out.

Given that we're now experiencing a slump in the economy, and you probably know a lot of people who are looking for work, would you have advice for the laid-off writers?

Keep learning stuff. Do volunteer work. Build up your portfolio in addition to looking for work. Take some classes if you can afford them. Or just take an elementary programming class just to get a feel for what people are doing out there.

Also, probably the most important ability is the ability to get a job. To know how to interview and to know how to sell yourself and to know how to do your résumé and to follow up. So that's as important as actually being able to *do* a job.

Say you're interviewing, and the hiring manager says, "Well, I'm really looking for somebody with a more technical background." You can say, "Oh. Oh well." Or you can say, "I understand that, and I was hoping that because I have this and this skill and was interested in taking that class, that that would help me to compete with people who may be more technical." And maybe the manager will say, "You know, maybe you're right. We've interviewed a lot of technical people and we're still interviewing, so obviously there's something else we're looking for."

And what's great is that people who don't necessarily have technical backgrounds are getting positions and excelling. I can think of some STC members who have recently got great jobs who didn't have a technical background.

They'd be good to interview, too.



XML Solutions

Continued from page 1

- ♦ Writers wanted to spend their time writing, not putting in XML tags; however, tags would be needed to create the XML Help files.
- ♦ None of the available Help tools, e.g. RoboHelp and Quadralay WebWorks Publisher, gave the team the ability to produce readable context-sensitive XML Help topics in XML and also readable PDF manuals from a single source.

Wraye's team's first big breakthrough came when they discovered DocBook, a standard set of XML markup tags used for publishing computer documents. DocBook gave the Niku team a free and usable document type definition (DTD). (A *DTD* specifies the hierarchy of markup tags and associated values allowed in an XML document.)

*Wraye's team's
first big
breakthrough
came when they
discovered
DocBook*

Another breakthrough came when team member and FrameMaker guru Steve Miller designed a revolutionary FrameMaker 6.0 template. The template used conditional text, paragraph and character definitions, and a modified XML mapping table in the reference pages so that writers could start a new topic with any tag that they designated as a start tag. Without the template, the team would have had to start each new topic with a single specific paragraph tag.

Miller's template also included cross-reference definitions that writers could use to create links to both XML and print outputs from the same cross-reference marker. The definitions prevented errors and saved writers a lot of time.

Before the team was able to produce XML Help files that read well and PDF manuals that read well, they also had to run Perl scripts to ensure that the XML created by FrameMaker had the requisite parent-child elements. (An *element* is a start tag and an end tag. The parent element would be higher than the child element in the hierarchy of data.)

Wraye and Miller hope that the newly-released FrameMaker 7.0 is a more XML-friendly and mature product than previous versions of FrameMaker. If FrameMaker 7.0 lives up to its promise, future single-sourcing teams will not have to jump through as many hoops as the Niku team had to.

News Briefs



Call for Proposals: ASTD's Fall Fiesta

The Eastern Idaho Chapter of the American Society for Training and Development (ASTD) is soliciting presentation proposals for its Fall Fiesta Conference, to be held Friday, September 20, 2002 in Idaho Falls. Proposals are due by July 15, 2002, and should focus on some aspect of training and development. Contact Richard Holman at (208) 526-1571 for more information.

Call for Proposals: STC's 50th Annual Conference

The Society for Technical Communication is soliciting proposals for its 50th Annual Conference, to be held May 18-21, 2003, in Dallas, Texas. Proposals are due by August 1, 2002; forms are available at <http://www.stc.org/conferences.html>. For more information, contact stc@stc.org.

Ideas and Books Sought for Local Tech Writing Library

If you are interested in helping to create a technical writing library, contact Barbara Herbert (Barbara@sonic.net).

Ames Elected to National STC Post

Andrea L. Ames, an associate fellow with the Silicon Valley Chapter, has been elected second vice president for the Society for Technical Communication.

Technical Editing Newsletter Available Online

The June issue of *Corrigo*, the newsletter of the STC Technical Editing Special Interest Group (SIG), is available online at <http://www.stcsig.org/te/>. The issue includes guidelines for estimating editing speed, suggestions for creating a style guide, and instructions for joining the SIG's online discussion group.

Professional Development Opportunities

Java for the Absolute Beginner

July 10, 6-week online Java course

\$59.00

<http://www.javaonline.org>

Beginning Web Design

July 12, 19, 26, and August 2, San Francisco

\$350.00

<http://www.media-alliance.org>

Advanced JavaScript

July 14 and 21, San Francisco

\$200

<http://www.media-alliance.org>

Continues on page 10

Professional Development Opportunities

Continued from page 9

Color PrePress

July 15 and 22 + field trip, San Francisco
\$250

<http://www.media-alliance.org>

Dreamweaver

July 16 and 23, San Francisco
\$300

<http://www.media-alliance.org>

Developing a Single-Sourcing Strategy

July 18-19, San Jose
\$725

<http://www.usabledesign.com/03register/03register.htm>

The eLearning Instructional Design Conference

July 25-July 26, San Francisco
\$1245

<http://www.eLearningGuild.com>

Copyediting Intensive

July 27; August 3, 10, and 17, San Francisco
\$195

<http://www.media-alliance.org>

Advanced Technical Writing

August 19-December 16, Santa Rosa
\$44 plus fees

<http://www.santarosa.edu> (select Schedule of Classes link)

Introductory and Intermediate Adobe Photoshop

August 19-December 16, Santa Rosa
\$33 plus fees

<http://www.santarosa.edu> (select Schedule of Classes link)

XML—An Introduction for Web Developers

August 19-December 16, Santa Rosa or Petaluma
\$33 plus fees

<http://www.santarosa.edu> (select Schedule of Classes link)

Beginning Web Design

August 24 and 31; September 7 and 14, San Francisco
\$350.00

<http://www.media-alliance.org>

