*NXApp*, Summer 1994 (Volume 1, Issue 3). 

# Shrinkwrap Corner

*written by* **Tony Rennier, BLaCKSMITH Inc.,**
**David Spitzler, NeXT Computer, Inc.**

## API: THE NEXT STEP IN SOFTWARE

The myth of the ªkiller applicationº is dead. It died with the notion that any piece of software, out of the box, could completely meet all the needs of any two people, much less millions. Is this surprising? Not in the least. Each individual has a wide variety of skills and tasks, and no two require identical things from their software.

Until recently users had two choices for using software: One, they could change the way they worked to accommodate the limited and inflexible functionality available in commercial software; or two, they could write their own software. While adjusting work habits to accommodate commercial software was often painful, it was usually significantly easier than writing their own apps.

Now, thanks to advances in software, users have a third choice: The software works to accommodate the user. What we're really talking about here are advances in software ªusability.º

Many of the advances in software usability center around the graphical user interface (GUI) and allowing users to customize the look and feelº of software. Yet, this is just the tip of the usability iceberg. The really exciting notion is that software is becoming less like *buildings* and more like *building blocks*. That is, users can purchase shrink-wrapped software and dynamically configure it for specific tasks.

NEXTSTEP is one of a few software environments that provide users with a simple and

intuitive GUI. And it's virtually the only software environment today that provides a solid foundation for the development of software building blocks. The key to putting the whole thing together is providing ways for the building blocks to communicateÐvia interapplication communicationÐand that's what application programming interfaces (APIs) are all about.

## How APIs help the user

Imagine that you own an international trucking firm and you need to track the locations of all your trucks on a map of the world. You have a database that continually tracks the coordinates of all your trucks and an app that draws maps. Without APIs you're forced to query your database manually and then import or retype the information into your mapping app each time you want an updated mapÐso much for real-time displays.

On the other hand, if your database and mapping app have APIs, you can put together a custom app that executes a query against your database every 10 minutes or so, for example, and sends the results to the mapping app for display. If your mapping app has a ªgoodº API you can even keep the mapping app hidden and still have the map image displayed in your custom app. Now you can just sit back and watch!

We've found that APIs are much more than an exciting glimpse into the future of software usability. They're here today and their use is central to the success of organizations that use NEXTSTEP, as well as to those independent software vendors (ISVs) providing solutions. If NEXTSTEP users wanted yesterday's software, they would use Microsoft Windows; the fact that they use NEXTSTEP means they demand more from their solutionsÐand from the companies that provide them.

## Levels of interapplication communication

From the user's perspective there are five different levels of interapplication communication (IAC) to provide expanded functionality to users:

1. **Communication through the file system** Apps can communicate with each other by reading and writing data to a file in a known data format.

2 **The copy and paste facility** Apps can transfer data using a holding bin called a *pasteboard*. This is powerful but labor-intensive for the user.

3 **NEXTSTEP Services** With Services, apps still communicate through a pasteboard, but user interaction is reduced.

4 **NEXTSTEP's Object Links** Sometimes referred to as ªhotlinking,º this is a fully automated copy and paste facility.

5 **APIs** The fifth and highest level of IACÐand our topic for this column.

## What is an API?

Perhaps the best way to think about APIs is to consider a client-server model. Most object-oriented and message-based software today is organized as an engine, or *server*, with a user interface (UI), or *client*. A user sees only the UI portion of the app. For example, in the mapping app we described earlier, the UI allows the user to specify things like the color of the United States or the geographic coordinates that are displayed. When it's time to update the image, the UI sends a message to the engine asking it to render an image with the required specifications. In this way the UI acts as a client to the mapping app's engine.

If the mapping app has an API, the opportunity exists for other apps to act as clients to the engine. This allows the custom app to send messages to the mapping engine specifying the color used to draw the United States and the coordinates of the map to be drawn, and requesting an image of the rendered map. In other words, other apps have the ability to exploit the map app's map-drawing expertise. This kind of communication transforms a piece of software from a simple application into a solution.

An *open API* is simply an API that is published. This means that an app can send and receive messages, and that the messages are documented and readily available to other developers who are trying to use the API. A *closed API* is one that isn't published or that rejects messages from unknown apps.

An example of a closed API is the API that's built into Presentation Builder, the graphing app

that came bundled with Lotus Improv. Improv and Presentation Builder enjoy a rich communication mechanism but invite no one else to join the party; it's impossible for any other app to send data to Presentation Builder to get it graphed.

In contrast, an example of an open API is what's implemented in CHaRTSMITH. With CHaRTSMITH, any app that has numerical data can simply send a message to the CHaRTSMITH API. The server creates a graph and sends it back to the client app if requested. Everyone is invited to the CHaRTSMITH party!

## APIs are good NEXTSTEP business

The NEXTSTEP community today is made up of many groups. For our purposes, we can divide them into two categories: those that develop mission-critical custom applications (MCCAs) and those that do not. In our estimation, the majority of sites today deployed NEXTSTEP largely because they could write and deploy MCCAs five to ten times faster than with other environments. Many of these sites purchase off-the-shelf productivity software as well, though some don't; almost any of them would purchase software that would help their developers produce MCCAs better and faster. Therefore, a well-designed, open API can be the sale/no-sale difference for a siteÐAPIs have a big advantage in the MCCA world.

### Apps vs. solutions

Here's an example from real life: When BLaCKSMITH released CHaRTSMITH™1.0, the product didn't have an API. From the beginning there was interest in the product, but it was evident that some customers were interested in purchasing a charting *solution*. If CHaRTSMITH couldn't pass the ªHow do I integrate great charts with my MCCA?º test, there would be no sale.

When CHaRTSMITH 1.1 was released with a comprehensive, open API, customer interest exploded. It was as if every MCCA site had been waiting for a solution to the same problem. In short, CHaRTSMITH was transformed from a great charting app to an even better charting solution.

### Toward desktop integration

Additionally, APIs make good business sense because they are consistent with NeXT's marketing strategy. The MCCA strategy includes the notion of ªdesktop integrationºÐthe integration of custom and commercial apps. Customers that require apps to be developed five to ten times faster also need them to integrate with productivity apps, rather than have a workstation for custom app development and a PC for running productivity apps.

So what does this mean? It means it's likely NeXT will give your product exposure if the product supports NeXT's marketing message. The theory goes, if it helps sell NEXTSTEP, then NeXT and NeXT's partnersÐVARs, system integrators, and so onÐwill use it. Probably software developers could think of nothing better than to have NeXT and other organizations help market their product.

Perhaps the best summary of the promise and usefulness of APIs is this comment from a developer at a large MCCA site: ªThe best line of code is one that I don't have to design, implement, test, or maintain, yet still meets my needs.º Perhaps we might add, ªThe best marketing dollar for me is the one that came from another company's pocket.º

## A real-life example

Does all this really hold true, or is it just theory? Consider some more recent history.

### That was then

When BLaCKSMITH started working on the CHaRTSMITH product, NeXTÐthen still in the hardware businessÐwas marketing NEXTSTEP as a powerful and intuitive GUI: ªUNIX for mere mortalsº and ªinterpersonal computing.º The new graphing application would be based around a fantastic UI first, an open API second.

It was clear to BLaCKSMITH then, just as it is now, that there was a tremendous advantage to being in synch with NeXT's marketing plan. As development continued it became clear that to meet the 1.0 schedule and to fit in with NeXT's objectives, the API would have to wait until the first interim release. So CHaRTSMITH 1.0 shipped with a compelling UI and full support for the first four levels of IACÐbut no API.

While CHaRTSMITH was still in development, NeXT began to evolve its marketing focus

toward its products' true strength: the NEXTSTEP development environment. And so the mission-critical custom application strategy was born. Based on a 1992 Booze·Allen·Hamilton competitive analysis of development environments, a variety of media, analysts, and industry pundits began praising NEXTSTEP because, ªYou can build custom apps five to ten times faster.º

So, by the time CHaRTSMITH 1.0 shipped, the market decided it needed a charting solution. Fortunately BLaCKSMITH was just around the corner from releasing one: CHaRTSMITH 1.1, a charting solution with a compelling UI and full support for all five levels of IAC, including an open API based on NeXT's Distributed Objects technology.

**This is now**

Today, CHaRTSMITH is a solution rather than an application. Here's why: With its comprehensive API, CHaRTSMITH is a chart server,º the same way the mapping app is a map server in the earlier example.

Another example: An MCCA fetches stock prices periodically from an on-line database. Each time this happens the new data is checked against parameters the user defined. For instance, if the price is lower than the buy price or higher than the sell price, a message is sent to CHaRTSMITH to change the status message. In addition, the new status data is sent to CHaRTSMITH, which draws a chart to illustrate the change. The MCCA then requests a new chart drawn directly into the its view. CHaRTSMITH is hidden, so the MCCA user has no indication that another app is being used to generate the chart.

ff.tiff ¬

Figure 13: *CHaRTSMITH 1.1, the charting solution*

**Moving into new realms**

The possibilities with the CHaRTSMITH API are almost endless. From time to time, customers ask if CHaRTSMITH can be used to produce chart displays that are ªnon-standardºÐthat is, chart types that may be useful to only a small set of users. Before the CHaRTSMITH API came

out, the answer was usually ªNo.º From a revenue point of view, the resources required to modify the application's UI to produce the nonstandard displayÐmainly people's timeÐwere usually significantly larger than could justify a change so few users required.

With the new API, however, this problem becomes substantially easier to solve. If a customer needs a chart display that shows a chart in a chart, for example, the answer now is ªYes!º Simply put together an MCCA that fetches the two sets of data to be charted, create two charts in CHaRTSMITH through the API, request the chart images, and finally draw the second chart directly into the first in the MCCA view.

#### The bottom line

CHaRTSMITH's transition from an application to a solution is having a dramatic impact on sales. Conservatively, more than 80 percent of BlaCKSMITH's current client base can be directly attributed to the CHaRTSMITH API. This means that sales have increased by a factor of at least 5. We expect this trend to continue so long as NEXTSTEP is used by organizations for developing and deploying MCCAs as well as for desktop integration of productivity apps.

One customer, Guy Roberts of the SwissKey Team of Swiss Bank, London, put it this way: Using CHaRTSMITH as a Distributed Object server, we quickly added stable graphing functionality to our application. Working this way was easier, cheaper, and more fun than using equivalent class libraries.º

NEXTSTEP users are very interested in solutions. Can you blame them?

## Conclusion

In the NEXTSTEP market, the futures of software and good business are one and the same. NEXTSTEP users, by their very nature, are looking ahead and expanding the envelope of information technology. Remember: Microsoft Windows exists for ordinary users; NEXTSTEP users are not ordinary.

As in life, the Number One Rule in business is to surviveÐand hopefully to thrive. Perhaps the Number Two Rule is to find customers and make them happy. One could argue that the second rule is nothing more than a blueprint for achieving the first. If developers provide

yesterday's software to customers who need tomorrow's solutions, they'll almost certainly fail. Developers who want to fulfill the needs of NEXTSTEP users must deliver tomorrow's solutions today. Providing software solutions, through well-designed APIs, is a good way to do just that.

Tony Rennier is a founding partner and president of BLaCKSMITH, Inc., and has been developing NEXTSTEP solutions for four years. David Spitzler is an ISV Business Development Manager with NeXT Computer, Inc.

For more information on CHaRTSMITH and the CHaRTSMITH API, please contact:

BLaCKSMITH, Inc.
9401 Mathy Drive, Suite 300
Fairfax, Virginia 22031
Phone: (800) 619-6147
Phone: (703) 250-1741
Fax: (703) 250-1744
E-mail: **info@blacksmith.com**

**OBJECT INTERFACES: OBJECTWARE AT ITS BEST**

Just as interapplication communication has been critical for BLaCKSMITH, so has it been for other developers. For example, Athena Design, Inc., has taken its Mesa™preadsheet and repackaged it as objects. The result is the Mesa Object Library Interface (MOLI)™, a rich API that allows the programmatic insertion of the Mesa spreadsheet engine into custom apps.

MOLI also offers spreadsheet and graph viewsÐMesaViewsÐas objects available in Interface Builder. MesaViews can be dragged from IB and placed in custom apps without writing any code. MesaViews also work as live objects in IB's Test Interface mode just as any Application Kit object does. In other words, MOLI objects integrate with the NEXTSTEP environment even though the designers of NEXTSTEP didn't anticipate these objects when they designed NEXTSTEP.

**Delivering value to customers**
A true test of technology is not whether it provides a good demonstration, but whether it offers value to customers. Using MOLI, AthenaÕs customers have interfaced Mesa to custom apps for front ends to telephone switching systems, monitoring medical instruments, inclusion of spreadsheets in financial apps, and much more. One customer is rolling out an app for users (traders) to monitor an array of financial instruments across a range of markets, so users can trade a series of financial instruments and hedge risk.

As usual, the users needed more functionality than the development team thought they could deliver in a timely manner. The app required real-time data feeds, a grid view of different securities, the ability to rapidly change the calculations that determine trading activity, and the ability to alert the user of important conditions.

The developers used MesaViews to prototype the app and demonstrate the functionality. Users were so pleased with the prototype that the developers are now using Mesa objects as part of the custom application. MOLI allowed the developers to do all those things that the users requestedÐand more.

**Making both programmers and users more effective**
The development team was able to use MOLI to deliver value to its users. The only problem was easily addressed: Users found that when data was closely spaced in the spreadsheet it looked fine, but when the users went to edit the data, the display didnÕt look very good in the editor. So the developers subclassed the Mesa spreadsheet object, wrote a new editor method, and replaced the Mesa editor with the one users liked better. This took one day to complete. The spreadsheet behavior is the same, and now the editor works better.

When Athena delivers technology to customers, it knows itÕs delivering tools to allow customers to be more effective in their work. Athena believes that APIs allow programmers to be more effective in delivering value to end users by offering easier-to-use apps, along with giving itself the ability to enhance functionality by subclassing rather than by rewriting.

**Other objects and APIs**
While Athena has been successful with MOLI, and BLaCKSMITH with the CHaRTSMITH API, they are not the only developers who offer customers object technology that allows the rapid delivery of value to end users. Many NEXTSTEP ISVs offer objectsÐfrom bar code palettes to graph objects to licensing objects, and much more. For information on commercially available ObjectWare for NEXTSTEP, call 1-800-TRY-NeXT and request the *ObjectWare Catalog*, Summer 1994.*ÐDavid Spitzler*

For more information on Mesa and MOLI, please contact:

Athena Design, Inc.
17 Saint Mary's Court
Boston, MA 02146
Phone: (617) 734-6372
Fax: (617) 734-1130
E-mail: **info@athena.com**

---

**Next Article** NeXTanswer #2000 **Core Dump**
**Previous article** NeXTanswer #1997 **appDidInit:**
**Table of contents**
http://www.next.com/HotNews/Journal/NXapp/Summer1994/ContentsSummer1994.html