

An Introduction to  
Open Source & Open Standards

**TABLE OF CONTENTS<sup>1</sup>**

**Introduction & Definitions ..... 3**  
**Open Computing and Open Standards..... 5**  
**The Role of Open Source Software ..... 7**  
**Driving Success ..... 12**  
**Conclusion..... 14**  
**About Us ..... 15**  
**Contact ..... 16**

---

<sup>1</sup> This document represents excerpts from a recent article by Douglas Heintzman

## INTRODUCTION & DEFINITIONS

The IT industry is going through major changes. New concepts in technology, such as Web services and grid computing, are opening the door to tremendous opportunities for taking eBusiness to the next level of profitability. The potential of these technologies to transform business is truly remarkable, and open standards and open source software will play increasingly critical roles in this new world.

In the first generation of eBusiness, standards allowed heterogeneous systems to communicate with each other and exchange data. This was critical to the development of the World Wide Web, eMarkets, eCommerce, and inter-company integration. These capabilities drove cost down and productivity up, while increasing both speed to market and business agility. During the next ten years, business agility will continue to be the critical business differentiator for businesses and governments, and those that can shift their business strategies quickly in response to market dynamics, emerging opportunities, and competitive threats will prosper as on demand organizations. In this next generation of eBusiness on demand — where computing resources become virtualized with corresponding flexibility and cost variability, where application function is discovered and bound to in a remote and just-in-time way, and where IT systems and business processes become integrated horizontally — open computing and standards will become more important than ever.

While the roles of standards and open source software do overlap in that many companies who are using open source in part as a means of implementing open standards, the cores of their respective value propositions are distinct and should be discussed as such.

### Definitions

One of the great challenges in the industry dialogue regarding "open" concepts is a clear definition for each of the various terms. Clarity of definition should contribute to clarity of discussion. For the purpose of this paper, the following definitions will be used:

**Open computing** — a general and inclusive term that is used to describe a philosophy of building IT systems. In hardware, open computing manifests itself in the standardization of plug and card interfaces; and in software, through communication and programming interfaces. Open computing allows for considerable flexibility in modular integration of function and vendor independence.

**Open standards** — interfaces or formats that are openly documented and have been accepted in the industry through either formal or *de facto* processes, and which are freely available for adoption by the industry. In the context of this article the term will be used to specifically refer to software interfaces. Examples that many people are familiar with include HTTP, HTML, WAP, TCP/IP, VoiceXML, XML, and SQL. They are typically built by software engineers from various IT/software companies who collaborate under the auspices of organizations such as W3C, OASIS, OMA, and IETF.

**Proprietary** — describes interfaces that are developed by and controlled by a given company and have not been made freely available for adoption by the industry. Proprietary software uses non-public interfaces or formats. When an interface is non-public, the owner of the proprietary interface controls it, including when and how the interface changes, who can adopt it, and how it is to be adopted.

**Open source software** — software whose source code is published and made available to the public, enabling anyone to copy, modify and redistribute the source code without paying royalties or fees. Open source code evolves through community cooperation. These communities are composed of individual programmers as well as very large companies. Some examples of open source initiatives are Linux, Eclipse, Apache, Mozilla, and various projects hosted on SourceForge.

**Free software and software libre** — terms that are roughly equivalent to *Open Source*. The term "free" is meant to describe the fact that the process is open and accessible and anyone can contribute to it. "Free" is not meant to imply that there is no charge. "Free software" may be packaged with various features and services and distributed for a fee by a private company. The term "public domain" software is often erroneously used interchangeably with the term "free software" and "open source" software. In fact, "public domain" is a legal term that refers to software whose copyright is not owned by anyone, either because it has expired, or because it was donated without restriction to the public. Unlike open source software, public domain software has no copyright restrictions at all. Any party may use or modify public domain software.

**Commercial software** — software that is distributed under commercial license agreements, usually for a fee. The main difference between the commercial software license and the open source license is that the recipient does not normally receive the right to copy, modify, or redistribute the software without fees or royalty obligations. Many people use the term "proprietary software" synonymously with "commercial software." Because of the potential confusion with the term "proprietary" in the context of standards and interfaces, and because commercial software may very well implement open, non-proprietary interfaces, this article will use the term "commercial software" to refer to non-open source software.

**MaxEXP** – San Diego Media's proprietary open source eBusiness software solution, designed to provide the maximum customer shopping experience. Comprised of easily configured integrated modules, MaxEXP provides scalability, flexibility, ease of use, and maximum security to most any size eBusiness endeavor. MaxEXP also includes a robust set of tools and features tailored to providing the highest possible customer care and service.

**Web services** – describes a standardized way of integrating Web-based applications using the XML, SOAP, WSDL and UDDI open standards over an Internet protocol backbone. XML is used to tag the data, SOAP is used to transfer the data, WSDL is used for describing the services available and UDDI is used for listing what services are available. Used primarily as a means for businesses to communicate with each other and with clients, Web services allow organizations to communicate data without intimate knowledge of each other's IT systems behind the firewall.

### The Open Computing Environment

Most major companies and governments have embraced the concept of open computing. They purchase IT goods and services from a variety of vendors and expect the technologies to work together. They wish to have the flexibility to deploy hardware and software in a specific way in order to address specific business problems. They do not wish to be subjected to the priorities and schedules of any particular vendor. Open computing provides them with a way to treat technology components as discrete modules that can be mixed and matched.

There are a number of common beliefs associated with this trend. In the first place, IT organizations investing in open computing believe it will maximize their flexibility and, consequently, the amount of business agility they have. They believe that open computing will allow them to rapidly adopt technology innovations and to exploit technology cost reductions. They believe open computing will provide them some degree of vendor independence. Well thought-out architecture and open standards are critical elements of open computing. Increasingly, companies are using open source software as a means of accelerating the adoption of open standards which subsequently allows them to implement open computing.

### A Brief History of Open Standards

Interfaces are the means by which the many elements of an application talk to each other and to other applications. Historically, interfaces were built by companies to allow internal and external programmers to add to the function of an application and to allow the value of an application to be enhanced with the capabilities of other applications. Companies that owned these interfaces wielded considerable control, which in turn enabled them to sell vertical software "stacks" to customers, or to assemble software stacks<sup>1</sup> that drove value into hardware platforms. This control model was very successful and the source of considerable profits for many of the major founders of the IT industry. When a company said they were an IBM shop, an HP shop, or a DEC shop, it really meant that, for the most part, they ran their business on a particular manufacturer's hardware and software.

In the 1980s computing technology started to become more stratified with much more distinct horizontal structures. Computer hardware became more commoditized and architectures more open. This led to greater degrees of modularization, interoperability, and the development of a marketplace for peripherals. The net effect was an increase in the rate of innovation, greater value for dollar, and a certain degree of loss of account control by hardware vendors. The software side of the equation also saw horizontal stratification. Operating systems became more generic and independent of hardware platforms. The middleware layer evolved, allowing for greater cost effectiveness and greater innovation at the client layer since application vendors were freed from having to worry about the plumbing.

These developments started to force interface standardization, which became vital in the effort to exploit networking technology and the growing usage of the Internet. The potential for computers to communicate with each other and for great stores of information to be virtualized was predicated on simple and standardized communications and interfaces. Therefore, while it may have been possible for a business to be an IBM, HP, or DEC shop in the past, it had become impossible for any one company to control the interfaces that ran the world's networks.

During the 1990's a number of major companies made strategic decisions to embrace this evolution toward open standards. These decisions were based on simple pragmatism: If we are going to live forever more in a networked world, then that networked world must run on open standards. This development has been good news for customers of IT and the IT industry in general. The skill and resources of these industry players have been critical in the development of robust, functional, and highly practical interfaces that are critical enablers of eBusiness.

The battle of "openness" is still being waged. For the most part businesses have embraced open standards as a means of ensuring degrees of flexibility and vendor independence. Many vendors have also embraced open standards, either because their role in the ecosystem as a provider of horizontal infrastructure or networking capability necessitates it, or because of their desire to participate in markets dominated by other players who use their market position to promote their proprietary interfaces. Some vendors have been successful in exploiting what economists call the "network effect" — the tendency towards adoption of a common platform owing to the intersecting interests and interdependencies of ecosystem participants, including consumers. In turn, these companies have been able to exert control over programming interfaces and document formats to protect their market positions. However, with the increasing momentum towards open standards and development of powerful alternative approaches such as XML, Web services, and J2EE (which isn't so much a standard as a widely adopted programming model), the ability to exploit proprietary interfaces for competitive advantage will likely diminish. At the same time, companies developing open standard eBusiness platforms, such as San Diego Media's MaxEXP product and IBM's Websphere, will continue to provide value by providing solid and proven platforms that are virtually impervious to obsolescence.

## THE ROLE OF OPEN SOURCE SOFTWARE

It has become clear that open source software (OSS) has an important role to play in the IT industry and business in general. Yet there is considerable confusion about the strengths and weaknesses of OSS. Some believe it will eventually replace the commercial software model, even that OSS is a critical element of a modern democracy. Others decry OSS as the single greatest threat to capitalism and the principles of intellectual property — the ruin of Western society. Neither of these extremes is accurate. OSS, for the most part, represents a software development process. It can be leveraged to provide considerable value and complement commercial software products. At the same time, commercial software products will continue to play a critical role for the foreseeable future. (The rationale for this conclusion will be discussed below.)

Open source software (OSS) isn't developed by any one company; it is developed by a community, and it comes in many flavors. For example, the Linux movement was started by an individual who was quickly joined by many others who used the Internet to collaborate on what is now a major open source platform project. San Diego Media's MaxEXP platform was developed through a collaboration of programmers and business team managers, including current customers of the solution. Others such as Apache are offshoots of academic work. Some, such as Mozilla and Eclipse, were seeded by substantial code donations from major software companies.

### Open Source Projects

OSS describes many different kinds of projects, each with different characteristics. These projects are long term and evolutionary in nature and don't usually have a defined end point. It is useful to break these groups down into four major subcategories, as described below.

*"Academic" projects:* University students and professors, and researchers from academic, public, and private research facilities use OSS as a means of peer review. OSS provides a mechanism for others to comment on their work, review its merit, and improve the code. Some researchers and companies publish their work to the public in order to promote discussion and innovation. Some companies use the publication of academic open source as a means of improving relations with other developers and researchers and even to attract talent. The emphasis is on innovative function. The code base typically isn't structured in such a way — and isn't stable enough — to deploy in a commercial setting.

*"Foundation" projects:* These projects focus on some of the lower layers of the software stack. The major examples are Linux (operating system), Eclipse (development tools framework), Apache (HTTP server), Globus (grid computing), and Mozilla (Web browser). These types of projects represent the bulk of open source community development. They are delivering the bulk of the value of the Open Source movement and have by far the highest level of commercial deployments and support infrastructure. These projects have large and vibrant communities including large commercial software companies supporting them.

*"Middleware" projects:* These projects are focused on areas such as application servers, databases, and portals. This group includes Web application servers such as Tomcat and JBoss, Databases such as MySQL, and Portal software such as Jetspeed. In most cases these communities have small private companies at their core who have service business models. These communities have not attracted a critical mass of programmers and are only likely to have a marginal impact on the software industry in the near term. The dynamics of this category will be discussed below.

*"Niche" projects:* This group represents the bulk of the 56,000 registered projects at SourceForge.org (a Web-based repository of OSS projects). These projects are typically very small, and have very niche focus. Many are experiments or toolkits developed by industry organizations. They collectively represent a significant influence on the market but individually have negligible impact. Code from these projects is used to test against and as a source of ideas and techniques.

There is one other category — *"Walled Gardens" projects* — that uses some of the same community development processes, but which is not really an open source project: i.e., the source code is not available to be modified by the public at large. Walled garden projects are composed of small groups of companies that put in place a mechanism to collaborate on the development of some common technology that all participants may access. The source code from this kind of project is only available to members of the walled community, whose membership is typically by invitation only and supported by dues. These types of projects are not common. Commercial software companies may also use a walled garden approach as an internal development process.

## **The Role of Foundation and Middleware Projects**

Of the open source project groups described above, the two most frequently discussed are the "foundation" projects and the "middleware" projects. These are the areas with potentially disruptive influence in the market. It is quite normal that mature, common, "foundation" layers of technology will become commoditized over time. Value will migrate to higher and more innovative layers. This has certainly been true of the hardware industry and will also be true of the software industry. We have seen many examples of this already. TCP/IP stacks, compressions tools, Web servers, and browsers come to mind. As a rule, the adoption of open standards by an industry will accelerate this migration of value as the cost of entry to a market falls and customers have greater choice. There have been some exceptions to this phenomenon — notably desktop operating systems and basic office productivity function. These markets have been protected in large part by the "network effect" discussed earlier. The combination of network effect and control over proprietary interfaces has slowed the rate of commoditization and allowed the realization of considerable control and profitability from those product areas.

Open source software has the potential to disrupt the status quo. Already it has presented a serious challenge to Microsoft's strategy to move into the Unix server business with Windows/Intel economics. Versions of OSS have become very capable and scalable server operating systems. Many companies and governments have implemented such servers because it offers them some key advantages. On the low end it can provide attractive economics. In the mid range it can run on high performance RISC servers and has great affinity with Unix. On the very high end it can run on mainframe business computers and even runs some of the world's most powerful super computers. This kind of scalability and multiplatform support has some very obvious advantages, including: 1) the ability to scale IT systems to match shifting business requirements, 2) the ability to optimize programmer productivity and minimize support requirements, and 3) the ability to leverage new innovations and take advantage of new cost structures regardless of platform.

To a lesser extent, OSS is also making inroads on the desktop. As companies and governments seek more cost effective ways to deliver IT services to their user populations, the combination of the efficiencies of open source, the runtime support of Java, the Web application model supported by a Web browser, and the aggregation and personalization services of a portal — combined with high function open source office alternatives such as "Open Office" — are providing a viable and financially attractive alternative to the traditional Windows environment.

## **Health and Sustainability for Foundation OSS Projects**

The growing popularity of open standards leads to an interesting question: How far up the software stack will open standards and open source accelerate commoditization, and at what rate? There are many factors to consider, the two most important being 1) the health and sustainability of a community and 2) the involvement of major industry players.

Regarding the size and relative health of the open source communities, we can distinguish the work of software development communities focused on the foundation/platform projects and the work of major industry players focused on projects at higher levels in the software stack.

## Large and Healthy Developer Communities

As noted earlier, projects such as Linux, Eclipse, Mozilla, Globus, and Apache have relative large and healthy developer communities. There are a number of reasons for this success, from both a business perspective as well as a developer perspective.

### Business perspective

- 1) These open source projects provide business value to end user customers. Because the community development process enforces modularity and standards compliance, these projects yield high dividends and provide significant value for businesses and governments.
- 2) These open source projects represent areas where one or more significant commercial software vendor has taken an active sponsorship role and where there is solid overall support. And the rewards are increasingly clear. Commercial software vendors recognize the opportunity to amortize collective investment into commoditized layers in the software stack. Moreover, commercial software vendors believe they can disrupt competitive control points such as critical interfaces or software stack components, eliminate vendor lock-in, and support multiple platforms. This is especially true of Linux.
- 3) Critical skills to support ongoing foundation projects are readily available in the software development community. These skills come from the hacker community, from the consumers of the technology, from companies with complimentary business models, from commercial software vendors, and from academia. All of these skill sources are fed by academic institutions that use OSS code to teach computer science concepts. Consequently, there is an educated population of programmers emerging out of colleges and universities who are familiar with the open source environment and with the technologies being built in open source projects.
- 4) Long-term plans for improvements (design, development, test, documentation, etc). The increasing number of conferences, commercial interest, and worldwide collaboration regarding the future of open source computing is a strong indication of its maturity and reveals the degree of importance associated with it.

### Developer perspective

- 1) Passionate interest in developing and enhancing code. These projects make for a highly visible "artistic canvas" for programmers to demonstrate their skills and technique. Some developers participate, at least in part, to establish a reputation and build credentials to interest potential employers.
- 2) Significant overlap between sets of users and developers. The direct participation of users and developers in the use of open standards and open source software leads to rapid incorporation of domain requirements into the code base and a significant degree of project focus on business oriented as well as technical problems. This focus supports the relevancy and value of the project's output and subsequently the health of community participation.
- 3) Diverse, interactive community. The sheer volume and the many diverse perspectives of project participants leads to rapid rate of innovation, optimization, and bug fixing.

- 4) Strong overall project/code leadership. A strong leadership style of the open source "maintainer" ( the person in charge of deciding what goes in and what stays out of any particular release) is essential. Linus Torvalds, the creator of the Linux operating kernel, is a great example of this leadership. Strong leadership promotes focus and momentum and generally helps keep the project moving in the right direction. This success factor, like many others becomes circular and self-promoting. The important successful projects attract strong leadership and the strong leadership in turn promotes the health of the project.

The foundation projects mentioned above are healthy and appear to be sustainable.

### **Involvement of Major Industry Players**

The second factor, involvement of major industry players toward viable OSS solutions, is also significant. Linux, Eclipse, Mozilla, Globus, and Apache all have active participation and support from some of the largest and most influential software companies in the industry. The Linux community counts HP, IBM, Sun, and Oracle among its active contributors. Members of the Eclipse community include Borland, IBM, Oracle, Sybase, Montavista, and Red Hat, and many others. Mozilla has participation by Netscape, AOL, IBM, Red Hat, and Sun. Globus is sponsored by IBM, Microsoft, and CISCO. Apache lists Apple, IBM, Sun, CollabNet, and Red Hat among its active contributors. These lists are, of course, small samples of the large number of companies that are involved in these initiatives. Their members also include contributors from Brown University, Massachusetts Institute of Technology, Stanford University, and many other academic institutions.

Why are major software companies increasingly interested in open standards and the open source movement? Primarily because investing in these communities makes good business sense, for three major reasons:

- 1) Drive rapid adoption of open standards. Those companies that have made a strategic decisions to back open standards and whose business model is based on widespread adoption of open standards support open source projects to help business and governments get ready access to high-quality, open source implementations of open standards and speed industry adoption.
- 2) Use OSS as a strategic business tool. OSS can eliminate competitor "lock-in," create a competitive playing field, and open up new business opportunities. Vendors who support OSS make money on OSS services, as well as through the sale of components based on open source platforms. And their customer base is satisfied by the ability to extend their systems via additional components and services from alternative vendors, as needed. OSS can effectively "level the playing field" by removing many of the structural advantages that a company controlling proprietary control points may have.
- 3) Extend mindshare. Active participation in an open source community can enhance partnership relationships and help build relationships with a broad spectrum of developers.

The importance of the participation of these commercial software vendors in the advance of OSS adoption cannot be understated. These companies provide a critical mass of highly skilled programming expertise and core competence. The involvement of these companies provides stability to open source communities and reassurance to consumers of the technology as to its long-term viability. The technical resources these companies provide can kick start an open source community and provide critical technical and domain expertise as well. For example, the donations of the Netscape code base to Mozilla by AOL and the Eclipse framework to the Eclipse Project by IBM have enabled two new OSS movements over the past three years. In addition, the financial resources of these companies can fund critical government standards compliance testing.

## **Middleware OSS Projects**

Middleware OSS represents a potential opportunity to many entrepreneurs — typically small companies seeking to develop alternatives to commercial middleware technology. And as we know, opportunity invites competition, and competition drives innovation. Some middleware OSS projects may migrate into the foundation efforts and be picked up by those communities. We have already seen basic application server functions migrating into the Apache project. This function will be very useful to many business and governments and will likely be used widely in the near future.

The commercial application server vendors are rapidly innovating and creating function that allows them to competitively differentiate their products and resist commoditization due to high-value capabilities such as portals, business processes integration, automation, nationalization, and other strengths. The middleware market will continue to evolve and very likely be characterized by high rates of innovation.

What is driving open computing, open standards and open source success?

The reasons for the success of open computing and open standards are obvious. They are a necessary feature of a networked world, and they are essential to the critical business factors of flexibility and business agility.

### General Benefits

In general, businesses and governments see value in the following OSS features:

- 1) Flexibility to modify. Some businesses or governments require specialized modifications to a code base to accommodate specific business or technical requirements. OSS offers this flexibility. The National Security Agency (NSA) has done just this and created a secure version of OSS. San Diego Media is continually expanding its MaxEXP platform, and provides new features and upgrades to its entire customer base as part of its ongoing service offering.
- 2) Cost effectiveness. OSS often has some attractive up-front cost advantages, although there is debate as to the total cost of ownership (TCO). There is anecdotal evidence that some companies have realized considerable license savings. On the other hand, it is argued that scarcity of skills translates to higher support and maintenance costs that nullify the up-front cost advantage. The economic case will vary from geography to geography as the availability of skill and labor rates vary. Unfortunately, there is no clear data on the total cost of ownership of OSS vs. commercial software yet.
- 3) Web Services readiness. Web Services is a new paradigm and probably one of the most discussed subjects amongst the IT solution providers. It is a simple concept that enables grand integration of various blocks of existing web infrastructure. It uses XML based messaging standards like SOAP, WSDL and UDDI, which will provide solutions to the industry for many more years. Open Source products like Linux, Apache, Perl and PHP are popular robust parts of web infrastructure. The rising need of SOAP based web services has prompted the development of libraries for various open source languages in line with some industry efforts around .NET and J2EE framework.

## Specific Benefits

In the specific case of the foundation projects, businesses and governments see value in:

- 1) **Multi-platform support.** Some businesses and governments have realized advantages in deploying a common operating environment across multiple hardware platform architectures. They also see some advantage in being able to scale their applications beyond the confines of one particular hardware architecture. An enterprise might deploy an application on an Intel platform, and then need to scale the application to midrange systems. There may be other reasons to move platforms such as reliability, manageability, corporate or departmental mergers, security requirements, or the exploitation of some specialized capability. This kind of flexibility allows businesses and governments to match their IT support to their business requirements and have that IT support change as the business requirements change.
- 2) **Standards enforcement/promotion.** Standards compliance is a natural and inevitable characteristic of community developed software. So some businesses and governments have decided to use OSS as a means of promoting or enforcing open standards and open computing. Implementing Linux or Apache for example implies the implementation of many of the most important Internet standards. Strict standards adherence at the lower foundation layers allows permits considerable flexibility of configuration and of choice of application and vendor.
- 3) **Auditability/Security.** Some businesses and governments believe that being able to see the underlying mechanics of code enhances their confidence in the reliability and security of that code. While this transparency argument does make aesthetic sense and many cite it as a rationale for selecting OSS alternatives, there isn't any hard data to suggest that OSS is inherently any more secure or stable than commercial software. Without question, security needs to be designed into a program, and a program's security needs to be assessed on a case-by-case basis. In any case, the platform foundation layer of the software stack is where most of this scrutiny is focused.

## CONCLUSION

Open computing, open standards, open source software, and commercial software which implements open standards are all succeeding because they are enablers of technological evolution and because businesses and governments recognize value in them. Businesses and governments will strive to attain the flexibility and business agility of the on demand world. Open computing platforms — both hardware and software — are essential underpinnings of the journey towards on demand computing. The role that standards have played in the evolution of eBusiness has been well established. The role that open standards *will* play as part of the open commercial and open source projects that embrace those standards is central to the further evolution of eBusiness toward more responsive, focused, and resilient eBusiness capability.

## ABOUT US

San Diego Media, Inc. is a Southern-California based company focused on designing and developing eBusiness solutions for small to mid-sized companies primarily in the retail, distribution and manufacturing industries. San Diego Media leverages open standards, modular-based architecture to develop software components that enable a stable, secure and state-of-the-art solution approach for it's customers, allowing the right balance of content and commerce for each eBusiness solution produced.

Our commitment to developing exceptional business-optimizing software is further reinforced by a comprehensive suite of professional services designed to facilitate the design, development, and support of each customer application.

### **Our Mission**

San Diego Media seeks to develop partnerships with companies in need of flexible, scalable, secure and profitable eBusiness solutions. Since launching our own proprietary eBusiness platform MaxEXP in 2001, we have been successfully proving to our clients that a network-based, open-standards software solution can provide everything necessary to fulfill their Web services and eCommerce needs. The open standards and modular architecture of MaxEXP provide the additional assurance of knowing that tomorrow's inevitable technology shifts will not impact its ability to conduct business. In short, obsolescence is not an issue.

As San Diego Media continues to market its powerful MaxEXP platform, it also continues to develop powerful new eBusiness tools designed to further enable productivity and efficiency.

Behind all of these power business tools is our complete professional services offering- a tremendous asset and powerful differentiator that other platform and software providers cannot easily offer. San Diego Media's professional services include strategic consulting, visual process modeling, graphic user interface (GUI) and design, content development, marketing, branding support and channel conflict management.



## CONTACT

Whether you're looking for significant online revenue, increased traffic and brand recognition, business process streamlining, or all of the above, we can help. Our open source technology and award-winning designers and engineers will give your eBusiness the power of content, commerce, and control, with a best-of-breed solution that will give your customers a reason to come in- and come back.

Give us a call today. We'd like to make you our next success story.

San Diego Media  
9820 Willow Creek Road, Suite 450  
San Diego, California 92131

Toll Free: 800.808.8696  
Local: 858.536.5505  
Fax: 858.536.5504

Email: [sales@sandiegomedia.com](mailto:sales@sandiegomedia.com)

