NTEN
Nonprofit Technology Network

Let’s Talk:

How Open APIs Can Change How Nonprofits Manage Data

An NTEN Report
January 30, 2007

By Michelle Murrain
with Katrin Verclas, NTEN

Special thanks to Eugene Kim of Blue Oxen Associates for review of this paper.

About NTEN

Where the Nonprofit Technology Community Meets

NTEN aspires to a world where all nonprofit organizations skillfully and confidently use technology to meet community needs and fulfill their missions.

We are a membership organization of nonprofit technology and program staff and technology providers. Our members share a common goal of helping nonprofits use all aspects of technology more effectively.

We believe that technology allows nonprofits to work with greater social impact. Our goal is to enable our members to do their jobs better, and to help their organizations strategically use technology so that they, in turn, will make the world a better, just, and equitable place.

NTEN facilitates the exchange of knowledge and information within our community. We connect our members to each other, provide professional development opportunities, educate our constituency on issues of technology use in nonprofits, and spearhead research, advocacy, and education on technology issues affecting our entire community.

NTEN
870 Market Street, Suite 1264
San Francisco, CA 94102

Phone: 415.397.9000
Fax: 415.397.1833
http://nten.org

Designed by Quilted.org

Executive Summary

Nonprofits manage lots of data: research, files, data on constituents, donors, and clients, etc. Choosing software to manage this organizational data is difficult enough. Getting applications to talk to one another in ways that are efficient and in support of the organization's mission is even harder. Application Programming Interfaces or APIs provide a way to simplify data management tasks. Software vendors and open source software projects are increasingly using APIs to help their customers/users increase the extensibility and integration possibilities of their software. This paper examines the problem that APIs try to solve, technologies behind APIs, vendor strategies, and examples of things you can do with APIs.

Contents

1. Introduction: The Problem 4

2. Definitions 5

3. Examples: Vendor APIs Today, Nonprofits Providing APIs, and Other Commercial Open APIs 9

4. Key Issues To Consider 11

5. Conclusion 14

6. Annotated Resources 15

7. Appendix: Self-reported survey of APIs of select applications 16

Introduction – The Problem

A recent study by dotOrganize[1] showed that many nonprofits struggle to manage their data. Nonprofit organizations have data stored in different applications, ranging from fundraising applications to event management programs, to client tracking databases, and grants management applications. Moving critical data from one database to another is difficult if not outright impossible. Perhaps the event management package won't talk with a fundraising database. Or the client management package and billing software aren't on speaking terms.

Nonprofits are challenged by managing their data effectively on a daily basis. This is hampering their work, making processes redundant and cumbersome, and adding precious time and costs to their operations – money and time better spent on services and programs.

Typically, organizations struggle with integrating data in three different ways:

1. By finding manual, clunky, or expensive custom methods of data integration. These include manually reconciling numbers from different packages (like accounting and event registration or grants management,) or manually exporting, and then importing data from one application to another. This kind of manual import/export (by, for example, using the export tools of one application to create a file that can be read by the import tool of a second application) is a very common method of data integration used.

2. By bringing in third parties to write custom bridging software to connect one application to another (often with little familiarity with either application).

3. By choosing a unified software solution that tries to bring together many different individual software functions (like fundraising and event management as well as bookkeeping). These 'one-size-fits-all' solutions are often not as good as individual packages in terms of sophisticated functionality, but because they are unified, and data flows easily from one function to another, they can be a compelling choice. However, these integrated packages are available only in some areas of the sector; in others, they simply don't exist.

APIs provide ways to more easily share data between applications. Open APIs also have the potential to bring data together in ways that create new ways of displaying information – such as bringing fundraising data to a content management system or blog, or by coupling geographic data with maps from Google.

In this paper, we'll:

- Explain what open APIs are conceptually,
- Describe what the technology behind APIs looks like,
- Give a few concrete examples of how APIs are used, and
- Delve into some issues to think about as you consider using these technologies in your organization.

[1] **Online Technology for Social Change: From Struggle to Strategy**: http://dotorganize.net/article.php?list=type&type=3 – a survey by dotOrganize on issues that grassroots nonprofits face in harnessing online technologies.

For the purposes of this discussion and to illustrate the potential of open APIs, we will focus on a relatively small subset of software applications and web services available to the nonprofit sector: those that focus primarily on constituency relationship management.

We are focusing on these applications for two reasons:

1. By definition, this class of CRM applications straddles several others (such as content management systems for websites and other databases for more specific functions such as events, donors, etc).

2. APIs are best developed for these applications. They are not only hotly debated for this class of applications, but are also part of the business model of at least two of the major vendors in this area.

API - Application Program Interface

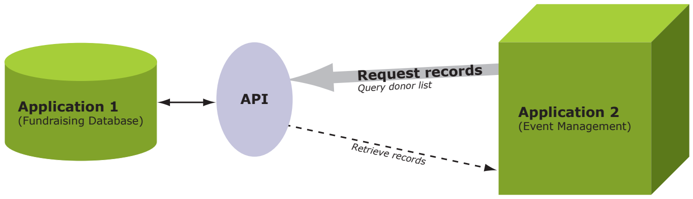
Wikipedia defines an API as:

"An application programming interface (API) is the interface that a computer system, library, or application provides in order to allow requests for services to be made of it by other computer programs, and/or to allow data to be exchanged between them."

Definitions

Exchange of data between applications, particularly database applications, is not new. One method that is commonly known, but inefficient, is exporting data from one application and importing it into another. There have been other database interface methods as well. But APIs, which are relative newcomers, have matured to the point where most applications now have some sort of APIs, and most vendors are at least planning to implement APIs if they haven't already.

Here is a basic diagram of how an API works.



One application (in this example, Application 2, an event management application) sends a request for data to another application (Application 1, which in this example is a donor database) in such a way that Application 1 can understand exactly what data is being requested. It could be, as in this example, a specific request for either a list of donors with particular characteristics or a query about a specific donor. The application would receive this request using one of a range of protocols, then answer it and return data if the conditions are met (conditions that pertain to authentication or a specific query, for example). Then, the requesting application (application 2) can do with the data as it pleases, which could include creating new records, checking against the data, or some other action.

Definition of an "Open API"

So what does it mean when an API is "open?" There is quite a bit of controversy and misunderstanding about this term. Often, it is used to talk about an API that is openly accessible or openly published to the public – such as Google's API, or the API for the social bookmarking tool, del.icio.us. For the purposes of this paper, an API that is open is defined ***as an API that is documented, is accessible to the audience it is intended for, and has no additional cost to access it.*** This means that if the API for a commercial software product is free and documented for the organization that uses the product, it's an open API. It does not mean that the API or the data it accesses is necessarily open to the public. In many cases, APIs might be implemented in an authenticated and secure manner to access organizational data that is not meant for public consumption, but for internal data integration needs.

In software that is written for the nonprofit sector, the vast majority of those APIs are open. Commercial vendors have, for the most part, responded to the nonprofits' need for data integration with new APIs for their applications that are accessible to their clients, and also typically reasonably well documented and supported. Most open source software projects also provide APIs, and, of course, because they are open source, these APIs can be improved upon or new APIs written as people need them.

Technology of APIs

APIs can generally be accessible via two means:

1. Internally (same-machine) APIs, or
2. Externally, via a number of technologies, including REST and SOAP.

Many APIs are platform dependent and server based – thus they depend on applications that reside on the same server (or at least servers of the same platform).

Examples of internal (same machine) APIs:

Blackbaud's APIs:

- dependent at this point upon the .NET 2.0 framework (a Windows framework, however in this instance, the APIs are not web-based); hence not cross-platform
- working on new APIs that are cross-platform. The release date for these was not obtainable by us, despite several inquiries.

CiviCRM's APIs:

- mostly php-based, and meant to be used on the same server as other applications
- provide rich possibilities for interfacing with open source content management systems, like Drupal and Joomla.
- There are plans for externally accessible APIs, though no concrete release date.

API Protocols:

SOAP (**Simple Object Access Protocol**) is the older of the technologies, and is a protocol used for interacting with APIs with XML (**eXtensible Markup Language**) messages (usually over HTTP, sometimes over SMTP). SOAP evolved from an even older protocol, called XML-RPC (XML Remote Procedure Call) that was quite limited in scope.

REST (**Representational State Transfer**) is a simpler, lighter protocol, used over HTTP, similar to HTTP GET. Messages are exchanged in any number of formats, including plain text, XML, and JSON (**JavaScript Object Notation**).

APIs accessible via SOAP and REST can (and often do) include authentication tokens, as well as keys, so that the source of the API request is vetted. Thus, the data accessible behind the API is secure.

Most of the more recent open APIs in nonprofit fundraising and advocacy applications such as GetActive and Salesforce, are based on REST and/or SOAP. Many commercial sites with open APIs such as Flickr, Google, and Amazon are generally implementations of REST.

Vendor APIs Today

All nonprofits have basic needs around keeping track of constituents and donors, and fundraising. And nonprofits often need this data to be accessible to other applications within their organization – such as event management, project management, and e-advocacy tools for example.

There are primarily two kinds of CRM and fundraising applications that nonprofits use:

1. In-house applications like Blackbaud's Raiser's Edge, Sage Fundraising, or Target's Team Approach or

2. Web-based, hosted services where the data is held offsite, such as Kintera, Salesforce.com, GetActive and Convio (which, in January 2007, announced a merger, combining the two companies)

What is CRM ?

CRM — originally **C**ustomer **R**elationship **M**anagement — it is a sales term that has been morphed by the nonprofit sector into Community or Constituent Relationship Management.

Most vendors of CRM and fundraising applications have included various methods for data integration. Some offer simple import/export, others have jumped fully into leveraging API technology to help nonprofits integrate their data.

Salesforce is a commercial vendor that is fully leveraging API technology. For the company, "APIs are the business model." Their web-based application is designed to be a platform. Salesforce, focused on the commercial sales market but with a nonprofit give-away program, has chosen to focus on providing the platform as a service and to give people tools such as APIs and Apex (a programming language that runs on their servers allowing for detailed customization) to help build on that platform.

GetActive, which is an application that is used for online outreach and advocacy as well as fundraising and CRM, has taken a similar approach as Salesforce in having open APIs, making them an important part of their business model. In addition, GetActive started a new effort, called "GetActive Extensions" that include new API libraries and tools and new ways for GetActive users to collaborate and exchange best practices and code. At the time of this writing it is unclear how the merger between Convio and GetActive will affect the future of the GetActive Extensions.

GetActive has had a number of APIs available to its clients, including ones that allow connections to client data. These enable capabilities such as adding records and syndicating content.

Blackbaud is the only major player that does not have open APIs (they charge for access to them), and their APIs are, for the most part, platform-dependent, which makes integrating this application with others harder.

There are increasing efforts to be able to integrate CiviCRM, an open source CRM/Fundraising application, with other applications. CiviCRM has a number of APIs that allow for access to the data that is housed within it. These APIs are – except for one – accessible only internally (on the same machine). That is, users can only access the APIs using PHP applications on the same server as the installation of CiviCRM. This can be useful for integrating with a CMS that is on the same server, but it becomes difficult to integrate with applications (such as accounting packages) that are on a different server or platform.

Example vendor APIs

Vendor	Open	Technology
Blackbaud	No	Server / Platform based (.NET)
CiviCRM	Yes	PHP, server-based
Convio	Yes	SOAP / REST
GetActive	Yes	SOAP / REST
Kintera	Yes	SOAP
Members-only	Yes	SOAP
Results Plus	Yes	Server / Platform based (.NET)
Salesforce	Yes	SOAP

* We contacted a number of vendors with specific questions about their APIs; we provide their responses in full in the Appendix of this report.

Nonprofits Providing APIs for Using Their Data

Vendors are not the only ones who take advantage of the potential of open APIs. In some instances, nonprofits hold a wealth of data that is made publicly available in the form of online databases. For example, APIs published by the Institute for Money in State Politics[2] for its campaign contributions databases have been used by Project Vote Smart to publish detailed candidate contribution data on their site. There are many similar online databases on census data (The DataCenter[3] , for example), climate change data, or air quality data that could be used more extensively by other organizations with the availability of open APIs that would allow advocacy organizations to talk to these databases in real time.

Other Commercial APIs

A popular use of an API for a large-scale service is mapping data important to nonprofits and their constituents. Using the Google maps API to map specific addresses is fairly straightforward, and a number of nonprofits have used it.

Full Stop[4] , a campaign of a UK-based organization dedicated to stop cruelty to children, has used to map people who join the campaign. Ghostcycle[5] uses it to create an interactive map of bicycle accidents in Seattle.

In short, APIs allow nonprofits to use their data more effectively, share it between applications more easily, and leverage their data for better outcomes in advocacy and fundraising. All nonprofits should be thinking about the software they use, and whether those products have APIs.

Examples of possible uses of public resource APIs

Resource	API Technology	Direction	Possible Uses
Google Map API	HTTP/JSON	One way	Mapping constituents, events or incidents
Flickr API	REST/SOAP/XML-RPC	Two-way	Collaborative image sharing, image posting about particular issues, fundraising using available images
del.icio.us API	REST	Two-way	Collaborative research, sharing links of constituents
You Tube API	REST	One-way	Collections of videos in particular categories

[2] See http://www.followthemoney.org/services/index.phtml
[3] See http://www.dataplace.org
[4] See http://www.bethefullstop.com/map
[5] See http://www.ghostcycle.org/

Key Issues to Consider

Integration

Complete data integration – that is, the seamless movement of data from application to application within an organization so that it can be accessed however and from wherever necessary – is the holy grail of data management within organizations. Because many organizations now use CRM, fundraising, advocacy, and client management packages that are hosted online with web service providers such as Salesforce, Kintera, Convio, or GetActive, for example, making the data fully accessible for internal organizational use is even more critical and, with open APIs, more possible.

Security

Security should be a priority for any nonprofit organization that has any sort of sensitive data on the Internet (membership lists accessible only to members, for example). Obviously, implementing an API is also something where security should be considered, especially when opening an API up to a larger public.

That said, there is no conflict between security of data and the openness of APIs. Our definition of an "open" API is simply that it is available at no additional cost to those for whom it is appropriate (e.g. clients of a vendor) and that is documented. It is important to understand the circumstances under which data can be accessed via the API. Is authentication necessary, for instance? The API should be designed with security in mind, and any organizational data, although accessible in a way that makes it easier to integrate with other applications or use in new kinds of ways, should be inaccessible to unauthorized users unless the data is public. This might mean that the server the data is on needs to be appropriately firewalled, for instance, or might involve talking with vendors about their security features if data is hosted offsite.

Open data for public-facing applications or for collaboration

Many nonprofits engage in activities that generate data that is either useful for the general public or useful for small groups of other nonprofit organizations, either locally or nationally. The options are endless. From projects like Project Vote Smart to creating interesting custom greeting cards for charities, from mapping services to sharing resources, the possibilities inherent in combining data from different sources into new applications and creative ways to display information (the term for this is "mashup") are endless. And, furthermore, there is a rapidly increasing amount of data that is freely available. Open APIs are an essential pre-requisite for these innovations for nonprofits and their causes.

Open source and open APIs

Open source applications have generally embraced open standards and open APIs. Open standards, such as HTML, HTTP, and XML are a standard part of the design of open source software. There is also an implicit or explicit understanding among open source developers that users are using multiple tools and integration is an important thing to build in from the beginning. Open source tools are, by nature, open – their code is open and modifiable, their APIs are by definition open (if not always well documented), and, if not present, the applications can be modified to include APIs. For example, there is an effort underway now to integrate Joomla, an open source content management system, with CiviCRM, a nonprofit-focused CRM tool. There are an increasing number of projects to connect CRM services such as Salesforce and Democracy in Action to open source CMS tools such as Joomla, Plone and Drupal used by nonprofits.

Vetting software with APIs in mind

Because of the overarching need for data integration, most nonprofits have looked to unified, one-stop packages that offer a broad array of functionality. Many vendors have been happy to provide these packages, and there are a number of them in the nonprofit market. However, as nonprofit needs vary, these unified packages may require significant compromises in functionality, or they simply do not exist. If an organization decides that, in order to get more complete functionality or because it is constrained by legacy applications, it needs to find an array of services, the next step is to use other methods to integrate data.

When making decisions about software, an essential question to ask is whether or not the vendor can or will provide APIs to access the data in both directions.

What kinds of questions are important to ask vendors, and what kind of information would you want to place in an RFP?

- Ask whether an application or online service has APIs, and ask for links to documentation, so that you can investigate what those APIs can do.
- Ask for examples of how other customers have used the APIs.
- Ask about the technology of the APIs (such as REST or SOAP), and the requirements for accessing the APIs (are they platform dependent?)
- Finally, ask whether the APIs free, or whether there are support or other costs associated with them.
- Do not be afraid to make a vendor explain what can be highly technical information in a way that you truly understand.

Discussion lists such as the NTEN lists at www.groups.nten.org are a good way to ask questions and hear from other non-profit staff about their experiences with specific vendors about their APIs. They are also great for asking technical questions about specific APIs.

Conclusion

Although APIs have been around, and many large-scale products for the for-profit sector have been using them for a few years, it is the rise of Web 2.0 tools that has brought APIs to the forefront, not only for interesting “mashups” but also for much-needed internal organizational data integration.

While many vendors in a variety of vertical sub-sectors, many software products for the nonprofit sector do not, at this point, have APIs. An increasing number of vendors are beginning to understand that the organizations they work with need flexible ways to access and integrate their data, and they are beginning to implement APIs.

However, with the exception of a few notable examples, we have not yet reached the holy grail of applications that seamlessly talk to each other and share data easily at a cost that most nonprofits can afford. Certainly, the more knowledgeable and sophisticated customers organizations become about APIs and their use in the growing nonprofit software sector, the more they can demand from their specific vendors, and advocate on a broader scale for their inclusion in products that the vendors offer the sector.

Resources

Open API Debate

http://nten.typepad.com/newsletter/2006/10/the_great_open_.html – the Open API debate with multiple perspectives on APIs.

Open API Debate panelists

http://nten.typepad.com/newsletter/2006/10/feature_what_ve.html – what the panelists said about APIs – and a link to the podcast.

Online Technology for Social Change

From Struggle to Strategy: http://dotorganize.net/article.php?list=type&type=3 – a study done by dotOrganize on issues that nonprofits face in harnessing online technologies

Programmable web

http://blog.programmableweb.com/. Website with information about public-facing APIs of all sorts. You'll find examples, code, and resources to help implement mashups.

How to roll out an open API from O'Reilly

http://radar.oreilly.com/archives/2005/05/web_services_es.html. This might be a useful resource if you are in the position of actually writing APIs (for open source projects for instance, or if you are a vendor)

Information for Money in State Politics API

http://www.followthemoney.org/services/ - the detailed information about access to their API, which was used by Project Vote Smart (http://vote-smart.org/index.htm)

CiviCRM APIs

http://wiki.civicrm.org/confluence/display/CRM/CiviCRM+Public+APIs. These are the current APIs (accessible mostly via php) that CiviCRM provides.

Salesforce APIs

http://developer.salesforce.com – lots of resources and details on APIs, code to access the SOAP layers, and information on Apex.

Flickr API

http://www.flickr.com/services/api/ The documentation on the variety of APIs the Flickr offers.

Google APIs

http://code.google.com/

del.icio.us API

http://del.icio.us/help/api/

ma.gnolia API

http://ma.gnolia.com/support/api

Appendix

We asked a number of vendors for their answers to a set of nine questions about their products in regard to their APIs. The questions are as follows:

1. Does your product have APIs that allow other applications to access data from your application?
2. Do you have features that call APIs of other applications?
3. If your application has APIs, what are the technologies used by those APIs? Please give a detailed technical overview.
4. We define an 'open API' as one that is documented, accessible to the audience it is intended for, and there is no additional cost to access it. Are your APIs open, as we've defined it?
5. If they are not open according to this definition, please explain.
6. Is there documentation for your APIs online? (If yes, please include the link)
7. If you don't have APIs in your applications yet, do you plan to implement them sometime in the next year?
8. If so, when will they be released?
9. Will they be open, according to above definition?

We asked the following vendors (those that replied are indicated in blue)

- Blackbaud
- Civicrm
- Citysoft
- Compass Technology
- Convio
- Democracy in Action
- Donor 2
- Donor Express
- Etapestry
- Fundtrack
- Getactive (acquired by Convio)
- Kintera
- Mission Research
- ResultsPlus! Software
- Sage Software
- Salesforce
- SofTrek
- Target Software (acquired by Blackbaud)
- Telosa

The detailed answers from the vendors that responded are reprinted here in full.

CiviCRM

1. Does your product have APIs that allow other applications to access data from your application?

Yes. CiviCRM includes a substantial set of API's which are being used to exchange data and integrate CiviCRM with other applications. We also use SOAP API's internally to integrate the CiviMail high-capacity broadcast email component with the CiviCRM Core.

2. Do you have features that call APIs of other applications?

Yes. We use this fairly extensively to add more functionality to CiviCRM. A few ones that come to mind are:

- Integration with two popular open source Content Management Systems, Drupal and Joomla.
- Integration with Yahoo! and Google API's for their geocoding and mapping services
- Integration with online payment services from Paypal (WebPayment Pro (Soap/XML) and IPN) and Moneris. The abstraction is flexible enough to plugin other payment processors.
- Integration with USPS web services

3. If your application has APIs, what are the technologies used by those APIs? Please give a detailed technical overview.

All API's are implemented in PHP. CiviCRM is primarily a PHP application (we integrate with one Perl component, amavisd). The API's can be accessed internally from within the CMS (Joomla and/or Drupal) if CiviCRM is enabled. Our mass mailer component, CiviMail, accesses some of the API's via SOAP. We have only exposed a few API's via SOAP since that is not the main focus of our audience. CiviCRM v1.7 (scheduled for March 2007) will also expose a REST interface along with a simpler API as discussed in our blog. This enhancement will greatly expand the number of applications which can "consume" CiviCRM API's. The CiviCRM ecosystem is at a stage where folks want to integrate CiviCRM with their favorite applications.

4. We define an 'open API' as one that is documented, accessible to the audience it is intended for, and there is no additional cost to access it. Are your APIs open, as we've defined it?

Yes. The APIs are included in our open-source distributions. They are supported by our team and our community. Documentation, code samples and unit tests are available online (see next response below).

5. Is there documentation online? (if yes, please include the link)

- CiviCRM Public API's (http://wiki.civicrm.org/confluence/display/CRM/CiviCRM+Public+APIs)
- Using CiviCRM APIs - Code Snippets (http://wiki.civicrm.org/confluence/display/CRM/Using+CiviCRM+APIs+-+Code+Snippets)
- SVN repository of unit tests for CiviCRM API (http://svn.civicrm.org/trunk/test/CRM/api/)

6. If they are not open according to this definition, please explain.
   (not applicable)

7. If you don't have APIs in your applications yet, do you plan to implement them sometime in the next year?
   (not applicable)

8. If so, when will they be released?
   (not applicable)

9. Will they be open, according to above definition?
   (not applicable)

Compass Technology

1. Does your product have APIs that allow other applications to access data from your application?

   Yes. The Compass CRMTM for Fundraisers business rules and processes layer is completely accessible via SOAP Web services and directly accessible via the public classes and methods in the business rules assembly.

2. Do you have features that call APIs of other applications?

   Yes. Compass CRM for Fundraisers is built on the Microsoft® DynamicsTM CRM 3.0 platform and calls the Microsoft CRM APIs.

3. If your application has APIs, what are the technologies used by those APIs? Please give a detailed technical overview.

   Compass CRM for Fundraisers is built on the Microsoft CRM platform. The APIs are written in .NET 1.1 and accessible by any .NET-compatible language.

4. We define an 'open API' as one that is documented, accessible to the audience it is intended for, and there is no additional cost to access it. Are your APIs open, as we've defined it?
   Yes.

5. If they are not open according to this definition, please explain.
   N/A

6. Is there documentation online? (if yes, please include the link)
   Not Yet.

7. If you don't have APIs in your applications yet, do you plan to implement them sometime in the next year?
   N/A

8. If so, when will they be released?
   N/A

9. Will they be open, according to above definition?
   N/A

Convio

From the CTO of Convio, David Crooke:

"Convio has recently acquired GetActive Software, and the GA team brings with them a rich expertise and heritage in providing Open APIs, and a customer base who are active users of them. As a result, Convio's expertise and product offering in this area will accelerate rapidly in 2007 and beyond."

1. Does your product have APIs that allow other applications to access data from your application?
   We have a number of APIs, primarily designed around real-time integration to specific third party applications such as donor databases and internet software that our customers would like to use with Convio and deliver a seamless experience with co-registration, single sign-on, etc. Some of these have been developed with partners such as Target Software and Heller Consulting, and others with specific customers. Our product also has rich support for less formal iterations based on non-webservice HTTP calls (e.g. "REST" style) which can be constructed by customers or our Services team using dynamic capabilities within our Content modules.

2. Do you have features that call APIs of other applications?
   Yes, we take advantage of many APIs in third party products as well ... a key part of the value that companies like Convio deliver to clients is the fact that, as a SaaS vendor, we delver a complete, working system with all of the third party moving parts. We include, use, resell or integrate with products and services from over 30 third-party vendors within the Convio application suite, and also offer a generic data integration engine that allows us to synchronize data between Convio and any offline constituent data repository.

3. If your application has APIs, what are the technologies used by those APIs? Please give a detailed technical overview.

Our formal APIs are done using SOAP Web Services, with the classic Apache Foundation Java stack on the server side - Apache Axis, Tomcat, etc. By careful selection of datatypes and formats, we design our WSDL specifications to be compatible with and easily used by all SOAP-capable platforms, including not only Java but Perl, Microsoft.NET, Ruby on Rails, etc.

4. We define an 'open API' as one that is documented, accessible to the audience it is intended for, and there is no additional cost to access it. Are your APIs open, as we've defined it?

The majority of our APIs at present are intended for integration to specific applications, and as such they don't need to be open by this definition - rather, our software engineers work closely with their peers at the third party vendors to ensure successful integrations. For certain facilities such as Single Sign-On integration which appeal to a wider range of customers, we provide standardised documentation and they are open by this definition.

5. If they are not open according to this definition, please explain.

Per our observations on the NTEN-hosted teleconference in the fall, we believe this delivers useful services to a much wider range of clients than providing APIs which require them to engage in custom software development, hence this is the area where we have focused our attention.

6. Is there documentation online? (if yes, please include the link)

The APIs and corresponding documentation that we offer are for use by our clients only, and not the general public, and as such are not publically available on the web.

7. If you don't have APIs in your applications yet, do you plan to implement them sometime in the next year?
8. If so, when will they be released?
9. Will they be open, according to above definition?

Response for 7-9: In the coming year, Convio anticipates that we will be considerably expanding the range both of (a) specific integrations between Convio and partner products, in the form of complete, supported solutions, engineered by Convio and its partners, and (b) generic Open APIs for use by customers, their third-party consultants and our own in-house Services team in developing custom solutions. However, we have not finalized our plans and cannot yet offer specific details.

eTapestry

1. Does your product have APIs that allow other applications to access data from your application?

   Yes

2. Do you have features that call APIs of other applications?

   Yes – IATS Ticketmaster, Network One, Delivra, Crescendo, Guidestar, PGCalc, WealthEngine, Walker Information.

3. If your application has APIs, what are the technologies used by those APIs? Please give a detailed technical overview.

   Detailed technical overview is probably excessive for a survey. Suffice it to say that we either use HTML to make requests or post data, or SOAP protocol to establish a session, query the database, and return data.

4. We define an 'open API' as one that is documented, accessible to the audience it is intended for, and there is no additional cost to access it. Are your APIs open, as we've defined it?

   No.

5. If they are not open according to this definition, please explain.

   We design the online giving pages using our proprietary API. For more complex operations we have a more sophisticated API for which we charge $5000 + ecommerce license (10% of base fees) to access it.

6. Is there documentation online? (if yes, please include the link)

   (We provide online documentation for customers who pay for access to our API.)

7. If you don't have APIs in your applications yet, do you plan to implement them sometime in the next year?

   (no answer given)

8. If so, when will they be released?

   (no answer given)

9. Will they be open, according to above definition?

   (no answer given)

Kintera

Kintera has prepared a library of existing code that provides instructions to use SOAP Web service calls in order to utilize the proven Web technology of HTTP with the flexibility and extensibility of XML. These interfaces take the inherent advantage of the HTTP authentication mechanisms as well as SSL for secure channel communications in providing a secure, real-time method to synchronize an external database or application with Kintera's software as a service (SaaS) technology platform, Kintera SphereTM.

In addition to the dozens of custom-built APIs that have been built and maintained for specific customers, Kintera maintains a public library of APIs utilizing SOAP Web services, found at https://www.kintera.com/services/sphere.asmx. Kintera's Web services are based on Kintera Web Service Framework, which provides infrastructure support for Web service security, error handling and usage logging. Kintera's current Web services are primarily used for integrations and custom projects for clients.

Kintera actively provides services including project management, development and consulting to support any project requiring Web services. Kintera maintains a dedicated staff of database professionals who are experts in the .NET environment who can help an organization interface with these Web services, or create entirely new APIs.

Currently, Kintera does not charge for use of its Web services, but the Web services are only made available to current Kintera clients. Kintera plans to develop a broader set of Web services in 2007 that will provide additional extensibility of Kintera Sphere.

Results Plus! Software

1. Does your product have APIs that allow other applications to access data from your application?.
   Yes

2. Do you have features that call APIs of other applications?
   Yes. We make use of many. The most well known ones being Word, WordPerfect, Outlook and QuickBooks.

3. If your application has APIs, what are the technologies used by those APIs? Please give a detailed technical overview.
   .Net

4. We define an 'open API' as one that is documented, accessible to the audience it is intended for, and there is no additional cost to access it. Are your APIs open, as we've defined it?
   Yes

5. If they are not open according to this definition, please explain.
   NA

6. Is there documentation online? (if yes, please include the link)
   http://support.metafile.com/fileareas/rpAPI/APIDocumentation.pdf

7. If you don't have APIs in your applications yet, do you plan to implement them sometime in the next year?
   NA

8. If so, when will they be released?
   NA

9. Will they be open, according to above definition?
   NA

Sage Software

1. Does your product have APIs that allow other applications to access data from your application?

   Not currently

2. Do you have features that call APIs of other applications?

   We call Sage MIP Fund Accounting API for our integration between our fundraising solutions and our fund accounting solution. We also call various APIs in integrating with 3rd party solutions such Microsoft Office, Outlook, and Sage Payment Solutions.

3. If your application has APIs, what are the technologies used by those APIs? Please give a detailed technical overview.

   NA

4. We define an 'open API' as one that is documented, accessible to the audience it is intended for, and there is no additional cost to access it. Are your APIs open, as we've defined it?

   NA

5. If they are not open according to this definition, please explain.

   NA

6. Is there documentation online? (if yes, please include the link)

   NA

7. If you don't have APIs in your applications yet, do you plan to implement them sometime in the next year?

   TBD

8. If so, when will they be released?

   TBD

9. Will they be open, according to above definition?

   TBD

Salesforce

1. Does your product have APIs that allow other applications to access data from your application?
   Absolutely.

2. Do you have features that call APIs of other applications?
   Absolutely.

3. If your application has APIs, what are the technologies used by those APIs? Please give a detailed technical overview.

Our API has been reconceptualized as a programming language in its own right. This language is called Apex. Here is a white paper that describes our model: https://wiki.apexdevnet.com/index.php/Apex_Code:_The_World%E2%80%99s_First_On-Demand_Programming_Language

One essential point to note is that salesforce.com is BOTH 1) On-Demand, meaning that the users need no more technology than a computer with an Internet connection to use it AND 2) Multi-Tenant, meaning that every organization lives on the same exact same service. This means that any upgrade or product advancement benefits EVERY organization and that we can achieve an economy of scale that is impossible for applications that require a separate "environment" for every organization.

Because our API can be understood as a programming language and our services are On-Demand and Multi-Tenant, users can build wholly custom data management applications leveraging our world-class infrastructure (hosting, bandwidth, database) or they can integrate any other web-services in to the salesforce.com platform.

The following languages or toolkits one can be used to write to our API: Java, Javascript, Microsoft (.NET, C#, Visual Basic, Office Toolkit) PHP, Perl, ATAPI (Telephony), Ruby, AJAX, Eclipse integration.

4. We define an 'open API' as one that is documented, accessible to the audience it is intended for, and there is no additional cost to access it. Are your APIs open, as we've defined it?
   Yes

5. If they are not open according to this definition, please explain.

6. Is there documentation for your APIs online? (If yes, please include the link)
   http://developer.salesforce.com

SofTrek

1. Does your product have APIs that allow other applications to access data from your application?

Yes, PledgeMaker has an add-on feature called InfoLink with an extensive set of inquiry and update functions.

2. Do you have features that call APIs of other applications?

Yes, PledgeMaker's InfoLink interacts with a number of third-party applications and web services for desktop integration, address standardization, credit card processing, email delivery, reporting, and other features.

3. If your application has APIs, what are the technologies used by those APIs? Please give a detailed technical overview.

The PledgeMaker Web API allows external applications to retrieve and update PledgeMaker data using standard web protocols. The API is designed for use by third-party developers and does not require knowledge of the internal structure of PledgeMaker. The API accepts two types of requests: inquiries (gets) and inserts/updates (puts). Requests and responses pass data as XML. Get type requests accept a set of parameters as attached XML and return an XML file containing the data requested. The list of get transactions is extensive, and can be customized to meet a client's specific need. Put type requests add or update information the PledgeMaker database. Hundreds of fields can be updated via the API, with duplicate checking, code translations, validations, and calculations performed automatically just as in batch file imports.

4. We define an 'open API' as one that is documented, accessible to the audience it is intended for, and there is no additional cost to access it. Are your APIs open, as we've defined it?

No, they are not.

5. If they are not open according to this definition, please explain.

The PledgeMaker API is documented and accessible via the Internet. There is an additional cost for its use.

6. Is there documentation online? (if yes, please include the link)

Yes. www.pledgemaker.com/infolink

7. If you don't have APIs in your applications yet, do you plan to implement them sometime in the next year?

N/A

8. If so, when will they be released?

   N/A

9. Will they be open, according to above definition?

   N/A