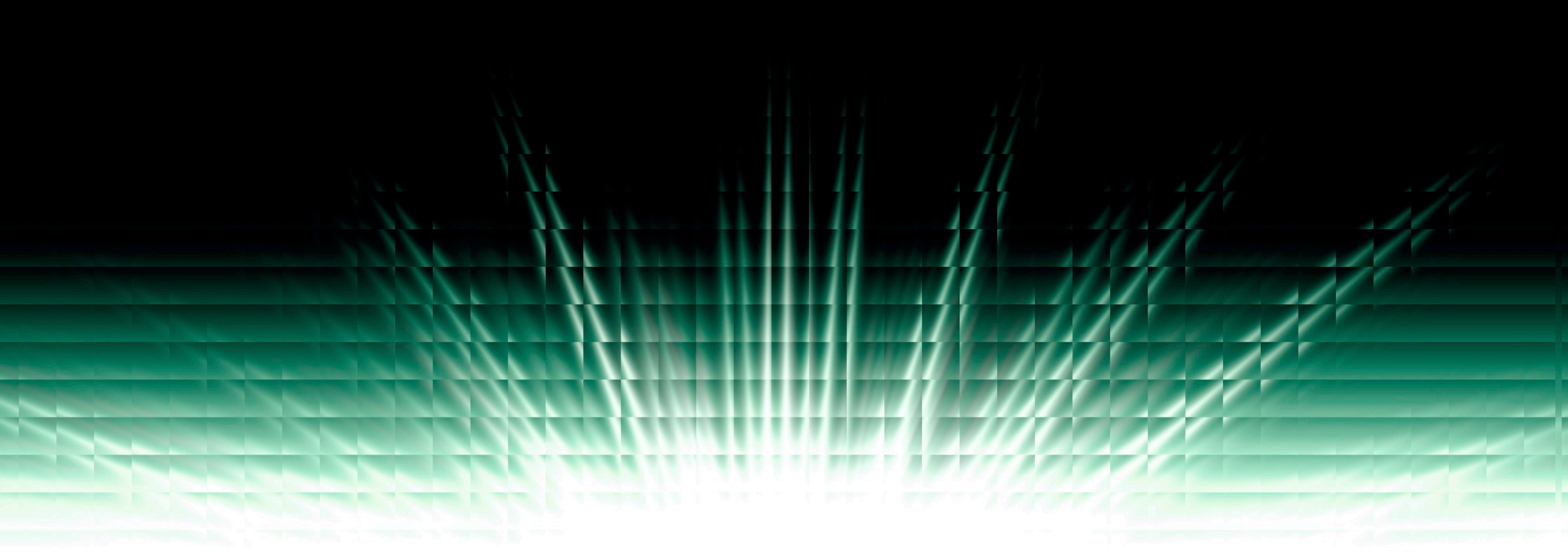

GROWING THE GREEN APPS ECOSYSTEM: BEST PRACTICES FOR DATA PROVIDERS

Open data strategies • Engagement tactics • Case studies • Developer interviews

BRIGHTER
PLANET

> *“The power of digital information to catalyze progress is limited only by the power of the human mind. Data are not consumed by the ideas and innovations they spark but are an endless fuel for creativity. A few bits, well found, can drive a giant leap of creativity. The power of a data set is amplified by ingenuity through applications unimagined by the authors and distant from the original field.”*
>
> *–Interagency Working Group on Digital Data, 2009*

About Brighter Planet

Brighter Planet is a sustainability tech company that helps organizations build brand, efficiency, and revenue by integrating carbon and energy analytics into custom applications for managers, employees, and customers. Since getting its start in a college classroom in 2005, Brighter Planet has partnered with dozens of leading organizations, attracted hundreds of thousands of customers, performed tens of millions of cloud-based carbon calculations, and prevented hundreds of millions of pounds of carbon dioxide emissions. This work has been recognized with a Financial Times Social Innovation Award, a TreeHugger Small Business of the Year award, and an EPA Best Overall Green Apps award, among others.

Brighter Planet’s CM1 web service powers sustainability calculations inside client applications, helping developers and organizations efficiently build green programs that harness hard scientific data to inform decision making. Used by hobbyist programmers and Fortune 500 companies, the CM1 API provides real-time, standards-certified impact calculations for activities ranging from travel to facilities to product lifecycles.

For more information visit http://brighterplanet.com.

This paper was edited and reviewed for factual accuracy by Ethan McMahon, Office of Environmental Information, U.S. EPA, in his capacity as an EPA employee.

The green apps movement

Green apps have enormous potential to mitigate increasing threats to the environment by using new data to power more sustainable decisions. Data providers, such as government agencies, can play a pivotal role in fostering a situation that helps developers make green apps.

Two aspects of the ongoing global information revolution are setting the conditions for innovative software applications to create real environmental good. First, the everyday decisions that govern the behavior of individuals and businesses are migrating to cyberspace. Shopping, communication, navigation, social interaction, banking, entertainment, utility billing, and any number of other activities are increasingly conducted or managed on computers and smartphones. These activities comprise the majority of human resource use and environmental impact. Software architects have more opportunity to influence them than ever before.

Second, data available to application developers is growing ever more abundant, more accessible, and of higher quality. The world's data volume is doubling every 18 months[1], and open data movements are seeing more and more data made available online from governments and enterprises. This is true of environmental data and of data related to many other activities that can be cross-referenced to create new environmental insights.

The real sustainability potential of these two trends lies in using data to power informed decisions. The app revolution is already providing significant environmental benefits, since many apps help users to be more efficient by improving navigation, shopping for second-hand goods, arranging carpools, or simply eliminating the need for books, papers, and gadgets by merging their functionality into computers and phones. But the focus of this paper is specifically on applications that connect growing environmental data resources to the digital platforms where everyday decisions are now made.

> ***"We think we can have a substantial impact on making the world a better place by enabling personal impact tracking."***
>
> *–developer,* oroeco TURNING GREEN TO GOLD

Large organizations with control over concentrated pools of resources are discovering this potential, and are increasingly using advanced software programs to manage their actions for efficiency and environmental impact. But the dispersed, decentralized actions of individuals and small organizations are only just beginning to be addressed by sustainability applications. There is an abundance of qualitative green claims and conservation tips in mobile and web apps, but quantitative tools that leverage hard data to drive environmentally-informed decisions have yet to realize their potential.

A community of green software developers focused on this latter opportunity is beginning to take root in the US, and data providers in government and business have an important role to play in fostering growth in this area. This paper outlines best practices for the two major roles these organizations can play in the green apps ecosystem: ensuring their data provision is as developer-friendly as possible, and engaging the developer community to encourage transformation of data into useful apps.

[1] PricewaterhouseCoopers 2011

It is a common myth that the greatest data challenge developers face is findability. In fact data location is usually the easy part; nevertheless data providers often—and with good intentions—spend the bulk of their effort publicizing their data. This is an understandable pattern. Collecting, preparing, and publishing data is an expensive and time-consuming process that is most immediately rewarded (and justified) by clicks, views, and downloads.

But long-term value depends on how engaged developers become with the data, whether or not they actually use it in an application, and ultimately on their applications' value to society and the environment.

This more comprehensive success relies on a number of factors that pick up where publicity leaves off. In the following pages we present strategies for organizations to channel data resources through green developers to create real public good.

These open data- and developer engagement best practices are informed by the results of surveys and interviews we've conducted with green developers, and our own experience with environmental data and green apps at Brighter Planet.

> ***“If you want the green apps community to innovate and continue grow in a positive way, we need access to resources, mentors, and experts.”***
>
> *–developer,* airSTAT.US

Open data: a reassessment

The open data movement has been around for years, and has brought about major changes in data practices at many key organizations. While the basic tenets of open data have been extensively documented elsewhere and aren't the focus of this paper, they do bear repeating here. Open data is traditionally defined as:

- **Free** and not licensed in any way that restricts use or redistribution.
- **Accessible** online, easily discoverable through the organization's website, and available for download in bulk.
- **Properly formatted** in standard, non-proprietary formats like CSV, XML , and JSON that are readable by machines and easily integrated into developers' databases, rather than in formats like PDF that aren't machine-readable, or closed formats requiring licensed software.
- **Primary** raw source data that doesn't assume a particular use.
- **Documented** to ensure appropriate and efficient use by as broad an audience as possible.
- **Current** and timely to ensure relevance and accuracy.

Amazing progress has been made in recent years by government agencies and other institutions in meeting the basic requirements of open data. The next step in fostering innovative data-driven applications is to reassess data provision policies from two new perspectives: *orientation* and *lifecycle*.

> “ ***The biggest challenge to date has been extracting meaningful, structured data, and turning it into valuable information or analytics.*** ”
>
> *–developer,* airSTAT.US

DATA ORIENTATION

Key takeaway: orient data files around topics, not publications.

Current situation:

Even before the advent of electronic data technologies, government agencies collected and published data diverse topics. Data was released through a scheduled (annual, quarterly, monthly, etc.) series of print publications available from the Government Printing Office and other agencies.

As data has been moved online, this "publication orientation" has continued. To use one popular dataset as an example, the EPA's Fuel Economy Guide is released annually, along with its print version. There is undoubtedly value in this orientation for certain audiences; the news media, for example, could analyze a newly-released annual guide to identify that model year's most efficient vehicles. But it doesn't help developers.

Developer reality:

Developers can be most efficient with topic-oriented data. To developers it's not relevant if the data was historically delivered in regular publications. Rather, the developer needs the data to drive her application.

With the fuel economy example, imagine a developer building an app intended to help used car buyers choose vehicles by looking at the total cost of ownership (purchase price plus ongoing fuel costs) rather than just upfront purchase price. To inform this application, the developer needs to feed it fuel economy data that covers all used cars on the market. Much to her chagrin, the current process involves downloading a separate file for each model year, resolving conflicting data formats and calculation methodologies, and assembling her own unified database. This hurdle is present throughout the open data ecosystem and consistently thwarts the efforts of green developers.

Solution:

Data providers should orient their data files around topics, not publications. To continue the fuel economy example, the Fuel Economy Guide website should provide a single data file listing all vehicle models since 1985 (adding a "model year" column). When an annual Guide is released, additional records are simply added to this file. The URL to this file should never change, so that developers and their apps know where to refer to it.

DATA LIFECYCLE

Key takeaway: automate and facilitate two-way data error correction and updates.

Current situation:

Data publications will always contain errors of some sort. Traditionally, print publications will be followed by one or more "errata" documents that enumerate these errors and provide corrections.

In many cases this tradition has persisted in the context of electronic data: an initial file is published, followed by one or more errata files that make corrections to the original data. In other cases, data providers integrate corrections into the master data and publish new, corrected data files, often incrementing a version number or adding a suffix like "corrected" to the filename.

Data updates—new publications of data—work similarly. A data provider will post a new file to its website and link to it from a "news" page or list of data files.

In all of these cases, the data provider requires that the developer to regularly check a website for revised data publications.

The reverse process—developers reporting errors in and suggesting corrections to a data provider's publication—is often equally cumbersome. When an invitation for submission is made, it often directs developers to a general email inbox with little or no guidance about reporting format or convention.

Developer reality:

Because the challenges outlined above, many green developers choose to retrieve data onve and never return for errata, corrected versions, or subsequent publications. The result is that new corrected data are not used, even if data providers made the effort to revise and release it, and derivative apps and decisions are incorrect.

Enthusiastic developers will monitor data provider websites for corrections and publications. Some use automated monitoring tools to heuristically recognize notifications of new data, but most will make personal visits to determine availability. Either way, one can describe this effort as "polling"—frequent trips, nearly all fruitless, with the occasional "Glad I checked!" moment.

Upon finding posted corrections, the developer's response will depend on the provider's method of delivery:

- When errata are posted, they are often in narrative (prose) form, requiring interpretation and manual application by the developer. Machine-readable errata are rare and, when present, still require manual examination to determine how to apply them.
- New versions of complete files, while nominally requiring less manual effort (i.e., the developer can simply re-run her import procedure), present their own challenges. Consider again the hypothetical fuel economy app described in the previous

section and imagine that the developer has included a "my options" feature, allowing the user to build a list of different automobiles she is considering for easy access. In software, this list would be persisted by recording the "ID" of each automobile record in its fuel economy database. But if an updated data file is imported that changes the order or number of records, these IDs will no longer match what the user chose, breaking the application. Recognizing this danger, the developer is forced to look for differences between the old and new files, identify the changes, and apply them one-by-one to the original data.

A developer who identifies an error in a provider's data is similarly vexed. The easiest route is just to fix the error in her local import of the data—this is known as "forking" because from that point forward, the developer's copy of the data has diverged from the provider's. An altruistic developer might choose instead to report the error to the data provider and await a data update—a process that could take weeks or months.

Solution:

Solving the lifecycle challenge involves three key practices—correction, notification, and error submission.

First is the process of correcting published data. In either case—errata or new file versions—the key practice is “keying” the data. Just as every U.S. citizen has a social security number, and every house an address, every record of data should have a unique key that never changes.

To see how this solves this problem, consider the errata method: the data provider releases an errata file in the same format as the original data file, with one row for each corrected record. The record's "key" is always provided—this allows the developer's automation toolset to identify the record in her local database that is being corrected. Then, elements of that record that have changed are provided in the errata file. This way, the toolset knows which details to correct.

> ***“The most difficult part was programmatically reading data from [the provider’s] website. It's a group of very old/slow web applications with no API and off the wall conventions for sending form data.”***
>
> *–developer,* SmarterMeter

A consistent key also enables automation of the "updated file" method. Returning to the fuel economy app example, the app uses the data provider's established key as the ID stored in the "my options" list. That way, when the data is re-imported, the list is still valid, because the unique ID of the automobile never changes.

In some cases, keying systems already exist—the International Air Transport Association defines airport codes, the United States Postal Service provides zip codes, etc. Where a universal keying system does not exist, the data provider should define one— they are the best equipped to handle this task. Data providers that do create their own keying systems should take care to provide "conversion" dictionaries to match up their own universal keying system with any partial systems already in the wild.

Second is the means by which developers become notified of new data. Having solved the keying challenge, notifications of errata publication vs. update file release can be made in the same way; in either case, the developer's toolset can be configured to retrieve the new data and incorporate it automatically. If a provider abides by the practice described above in Orientation, the process of publishing new data is equivalent to the "new version" correction process, since brand new data is simply added to a single master file.

The notification process itself depends on the provider's method of publishing corrections:

- If a data provider adopts the single-file-per-topic practice described in the Data Orientation section, and issues corrections by updating that file, machine-readable notification is unnecessary. The developer will simply set her toolset's automated import process to run at regular intervals (hourly, daily, etc.)—this process will import new and correct data soon after it is published.
- If the provider prefers to release errata, the developer's toolset will not automatically know where to look for these files. The provider can either provide an RSS/Atom feed linking to new errata files (which the developer's toolset will poll regularly) or establish a PubSubHubBub subscription point for each master data file they publish (the developer will subscribe her toolset to this endpoint which will thereafter receive notifications whenever new errata are available).

Third is facilitating the submission of errors from developers. The key concept here is to enable the developer to apply the correction to her own local copy of the data, while also easily submitting it for inclusion into the provider's file. The most straightforward way to do this is to accept submission of errata files, just as they would be issued by the provider itself. The developer can apply the file to her own data, and the provider can either issue it publicly (after review) or apply it to the master file for update release.

CASE STUDY: DATA.GOV

Data providers sometimes try to help developers but don’t always solve the right problems. In early 2009, the U.S. federal government's first Chief Information Officer announced the creation of Data.gov, a single-source repository for government-produced data. But as many green developers will tell you, the lack of a single repository for all data has never been their greatest challenge. (Developers have a much harder time understanding data than finding it.) Developers are accustomed to identifying authorities and assessing their publications for desired data. What's more, Data.gov currently weighs in at over 250,000 datasets, making search the only practical navigation tool—a process that, conducted with Google, would often just as easily turn up the data at its original location.

Data.gov does do an excellent job delivering data in consistent formats. But this standardization comes at a cost: when data is distanced from its original compilers, subsequent challenges in Orientation and Lifecycle will only intensify.

With budget cuts looming on the horizon, Data.gov's place within the broader electronic government effort may be evolving. Leaders could consider a decidedly advisory role for the organization, assisting agencies with applying the best practices discussed here and elsewhere to their own data publications. As an "internal consulting firm," Data.gov could increase value throughout the government data ecosystem without distancing the data from its original sources.

Engaging green developers

Data availability and openness is critical, but innovation in this space also depends on a community of engaged software developers with the motivation, knowledge, resources, and audience to create compelling new apps. Data providers, as well as players in other segments of the app ecosystem, can help to propel the success of sustainability applications by working to engage developers and users through a variety of initiatives.

Contests

Contests are a proven way to drive creative use of data resources and development of green apps. While prize money is an obvious motivator, publicity, endorsements, and reputation can also inspire competition. The scope of contests can be tailored to the goals of the organizer (e.g. open to the public, to students in a particular class, or to clients of a certain company), as can the deliverables (e.g. apps developed, new users won, ideas generated, votes garnered).

Action: Don't treat participating developers like your own private tech team—establish significant prizes or incentives.

Case study: *EPA Apps for the Environment Challenge*. In one of the top contests ever run on the Challenge.gov platform, the EPA engaged scores of developers across the country who generated many innovative apps that use environmental data. They organized a contest that included both public voting and expert judging, and motivated entrants with recognition rather than prize money.

Hackathons

Hackathons bring groups of coders together for short periods of time to collaborate on software projects. The concept has been around for years, and it's a tried-and-true tactic for engaging developers and inspiring innovative work. The goals and parameters set by hackathon organizers can range from focused to open-ended depending on the objectives, but the tradition is to leave coders as much leeway as possible to encourage creativity.

Action: Partner with local programming organizations, develop participation incentives (great food, sponsored activities, etc.), include a strong theme, provider access to data experts, and publicize heavily.

Case study: *EcoHack NYC*. Nearly 100 developers, designers, and scientists came together for a day in New York this month to collaborate on environmentally-oriented apps, helping to deepen network connections across organizations and produce the seeds for numerous innovative green projects.

> ***"Communal Hackathons [get] people of different backgrounds to create something together—not just computer hackers, but designers, planners, city employees, students, writers, and others converging on a single idea."***
>
> *–developer,* San Francisco **Street Cleaning**

Funding

One challenge facing the creation of data-driven apps for individuals and small organizations is economic. Many of the potential apps that could provide major sustainability benefits for users may not be viable as business plans, so organizations with an interest in green app development can help overcome this hurdle by sponsoring the creation of apps. Grants, fellowships, and one-time commissions can all be viable funding models for software developers. The funds could come from governments, non-profits, donations, or by aggregating the demands of multiple users.

Action: Recognize that third-party apps using your data could potentially advance your organization's purpose better than any in-house program could. Offer grants/fellowships that can be rapidly awarded with little to no red tape—developers gain and lose interest quickly.

Case study: *Summer of Smart*. Summer of Smart, an initiative that brings together citizens and government to address urban issues in San Francisco, has supported coding efforts by funding residencies and development fees associated with social apps.

> ***“It will be challenging to build partnerships that generate revenue, since we are building a new product without much precedent.”***
>
> *–developer,* oroeco TURNING GREEN TO GOLD

Promoting existing apps

The success of the green apps movement depends not just on the creation of innovative environmental apps, but also on their market penetration. Publicity is key in a media environment flooded with hundreds of thousands of software applications, and every organization rooting for the success of green apps has an important role to play in helping to promote the apps it supports. Giving coverage to other developers' green apps through blogs, social media accounts, and other channels can help an organization engage the developer community and support sustainability while also building its own brand.

Action: Don't ask developers for an app that's already available—closely monitor the green apps space to identify outstanding works and publicize them generously.

Developer hubs

Another tactic for engaging developers is to bring the conversation to them on their own turf. Sites like GitHub that host developer coding and communication are a great place for data providers to build awareness and networks by commenting on projects and following key developers. Encourage developers to collaborate with ideas, code, functionalities, and user interfaces. It's often better for teams to combine their skills than to compete with each other.

Action: Create an account on GitHub for your organization. Get in-house developers or volunteers to follow related repositories, comment on commits, and offer patches via "pull requests" when your data changes and a corresponding modification is necessary.

Support

Data providers should be readily accessible to developers. Simply having clearly stated points of contact for each specific data set, so that the human side of the organization is visible and questions can be easily asked, is a great start. A more engaging option is to open an Internet Relay Chat (IRC) channel to provide data support by allowing developers and data providers to chat in real time via online text.

Action: Publish the email address of the maintainer with every data set you provide, along with a phone number if possible. Create a #yourorganization channel on Freenode's IRC network that's passively monitored by at least one employee during business hours.

> ***“[My biggest challenge was] access to subject matter experts when I had questions about the meaning and impact of the data.”***
>
> *–developer,* airSTAT.US

APIs

While not a requirement for open data, web services built to easily serve data directly to applications are a great way to engage and empower developers. Application programming interfaces (APIs) are a commonly used tool that let developers integrate with data sources to make real-time, automated data requests as their apps are in use. They can save developers an enormous amount of time importing, curating, and processing data, or can enable apps that would't otherwise be possible.

Action: Especially if your organization's data changes frequently, consider exposing it via a well-documented API that uses open standards.

Conclusions

The emerging green developer community has the potential to produce a new class of environmental innovators. The passion and creativity of a single developer can be magnified exponentially as her app gains users and begins to influence the billions of small decisions we all make every day. These small steps can add up to big change.

Data owners, such federal government agencies, have an important and unique role in making this a reality. They can create and maintain relationships with developers that help them truly understand developer needs. They can provide data experts to ensure that data developers use data properly. And they can also provide incentives that encourage developers to make apps that help the public.

To summarize the advice to data providers in this paper: watch developers carefully, and see what you can do to help. Take note of the things that every individual developer must do to your data before it can be used. Follow your error reporting process from end to end and identify failures. Align your organization's goals with clear developer incentives. Above all, provide specific contacts for each dataset, and make them available via channels developers already comfortably use.