WHITEPAPER:

XMPie and Open XM

A new approach when bridging print and digital media

XM PIE

one to one in one™

A xerox Company

XMPie and Open XM™

Contents

PREFACE

A New Era in Cross Media Communications

Integrating digital and print media for effective individualized multichannel communication

Cross Media Communication is all about engaging customers through individually relevant dialogues that extend over time and multiple touch points. When done well — creatively compelling and individually relevant touch points — such communications deliver excellent business results. Moreover, there is evidence that communications with a mix of digital and print touch points are more impactful than those that are purely digital.

A key challenge in creating multichannel communications (we use Cross Media and Multichannel interchangeably) is to make sure that these dialogues will appear as *one conversation*. Without that, multichannel becomes an ineffective babel of dialects, one per media-channel. Another key challenge is to make sure that for each touch point, *the interaction experience will not only be relevant but that it will reflect the state-of-the-art for the touch point's media*. This is because outdated experiences are never engaging.

Creating individualized multichannel communications that will deliver both "one conversation" and "State-of-the-Art experiences for each media" is exactly what we aimed to enable, practically from the founding of XMPie. And this paper is all about a new technology — Open XM™ — that we are introducing, which makes "State-of-the-art" a much more attainable characteristic for multichannel communications. Open XM embraces the state-of-the-art capabilities and workflows that are used today in the digital media world, while ensuring cross-channel consistency and synergy across the print and digital touch points of the dialogue.

Jacob Aizikowitz

March 2016

XM PIE
one to one in one
A xerox Company

The Challenge

Bridging digital-media and print-media for consistent and meaningful customer communications

Bridging digital-media and print-media personalization technologies in order to create a unified personalization framework for cross-media applications, was the vision behind founding XMPie. There are a few key principles that we established early on, and they continue serving us perfectly well, even today:

1. The commonality between the various media touch points — say the print piece, the email piece, and the landing site — will be an encapsulation of the data and the rules and not just the data alone, and certainly not the design.
2. Design for each media channel should be media-specific and use the state-of-the-art design / development technology for that media type.

The two together mean that while there will be consistency across media touch points in terms of content, including personalized dynamic content, the user experience for each media type can be the most advanced possible for that particular media, and will therefore be creatively compelling and engaging.

An important observation is that trying to provide cross media consistency just by having a common database for all media-types is a mistake. This is because with personalization and dynamic content there are always scripting rules that use the data to compute the variability; restating such rules separately for each media is a recipe for disaster.

This paper is about Open XM, a new approach that XMPie developed for handling the digital media touch points. What's novel about Open XM is that rather than forcing the use of an XMPie proprietary design tool for digital media, we are open to all of today's modern design and development environments. We have to allow the professionals — in our customer's shop, of our customer's customers, or at an agency providing services — to use the tools of their choice and, in addition, to easily collaborate, review, change, and deploy.

The Open XM approach to digital media is very different from what we introduced for creating variable data documents in print. But, there are good reasons for that.

Below, we will explain our motivation and the need for the two different approaches, but we will show how elegantly they are linked and integrated, sharing the XMPie Personalization Engine as a common foundation.

Designing for Print vs. Designing for Digital

Designing for print is about creating one document, which will be printed once or in a series of identical copies; we use “static” to refer to such documents.

Designing for digital is about creating content to be delivered through websites (or other digital media touch points) that change based on data and rules applied through a scripting language; we use “dynamic” to characterize such presentations (e.g., dynamic website).

Also, designing for print is almost exclusively done using the Adobe Creative suite of tools (InDesign, Illustrator, and Photoshop), whereas designing for digital has no single exclusive design tool that everybody uses universally. Moreover, the direct use of HTML and even scripting, such as JavaScript, is common practice in designing for digital.

It should be clear from the above that while there is certainly a place for a technology that will introduce personalization and dynamic content to print, such technology is not needed for digital. Any attempt to bring to the digital space proprietary design / development tools is doomed to fail. This is because not only are people in digital questioning whether print is needed, they are certainly not going to accept being restricted to use some proprietary design tool just in order to have consistency between digital and print. The reality is that they have not embraced it and most likely will not embrace it.

Moreover, with the light-speed advancements that the likes of Google, Facebook, Amazon, Adobe, or Microsoft are driving in the digital media worlds, the likelihood for a cross-media solutions provider to be able to provide a proprietary design environment for digital that will be as current as what’s the State-of-the-art in digital is rather slim.

All of these, together, led us to re-think the approach to digital, and this is where Open XM was born.

Next Generation Cross Media – Open XM

The key message of Open XM is that for digital-media the solution does not require the use of a specific, proprietary, design tool. *Instead it provides a technology stack for accessing XMPie personalization which can be used in multiple ways and by any web design tool*.

With Open XM, one can use a design tool of choice — even editing the HTML directly — and create state-of-the-art digital media experiences, where the technique of adding personalization is no different than the technique used for, say, adding Google analytics or connecting to Facebook (See Figure 2: Google Analytics, in the Appendix section below; see Figure 3: Facebook, in the Appendix section below).

HTML pages — websites but also emails — will include code snippets that call upon XMPie personalization in an industry-standard, well-understood way. A way that is supported by any digital design tool.

And this makes Open XM a solution that is open, enables the use of digital's state-of-the-art technology, and supports collaboration.

The Open XM Technology Stack

The Open XM™ technology stack consists of XMPL (an XMPie dialect of HTML), XMPL AngularJS Lib (a JavaScript library for invoking the Data / Logic layer), XMPL Rest APIs Server (the APIs that implement the functionality in the XMPL Lib and XMPL HTML), and including also the classic uProduce SOAP APIs that are the true access point for the Data / Rules / and computation capabilities of our foundation (See Figure 1: Open XM Technology Stack).

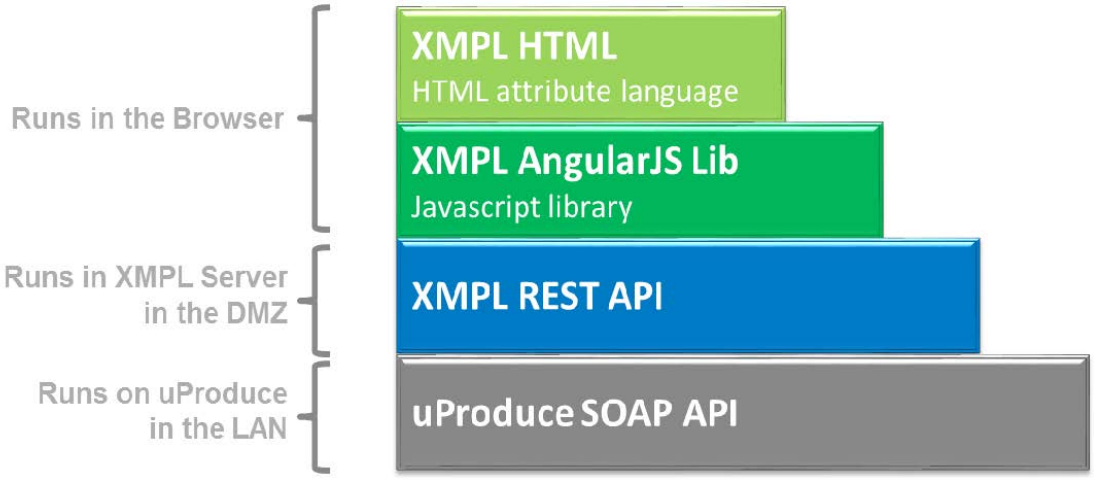


Figure 1: Open XM Technology Stack

The layered and indented structure of Open XM, as shown in Figure 1: Open XM Technology Stack above, reflects its ability to serve personalization through different scenarios or workflows:

- Pure HTML based personalization through tagging an HTML file with the XMPL HTML tags;
- Web-pages that use JavaScript and invoke the XMPL Angular JS Lib in order to incorporate personalization; or
- An App that introduces personalization by making API calls directly to the XMPL REST API layer

These layers are all different faces of ADOR® Technology — the personalization technology foundation of XMPie. For an example of invoking Open XM, please see Figure 4: Using XMPie Cross Media Technology Stack, below.

Supporting Diverse Design Environments

For users of Adobe Dreamweaver, we updated our uCreate plug-in to Dreamweaver to use Open XM and to maintain the resulting HTML file and website to be open and fully compatible with any other design environment. This enables collaboration across the value chain, where templates and websites created with Dreamweaver and uCreate can be used by any other design environment and vice versa. With Open XM collaborating with the digital team of a brand or an agency is a breeze.

Supporting Programming

With Open XM, one can develop JavaScript solutions to be executed within a browser and bring personalization to these programs by using the needed elements from the XMPL AngularJS Lib of the Open XM stack.

Also, one can develop Apps or other programs and use directly the XMPL REST API element of the Open XM stack in order to introduce personalization to such programs.

In summary, through the Open XM stack and our uCreate digital Plug-in to Adobe Dreamweaver, we are providing a breadth of possibilities to implement cross-media — from novice developers to extremely experienced ones — but with these common characteristics:

- It is State-of-the-art for digital
- It is open and enables collaboration
- Personalization is consistent across all media touch points

Summary

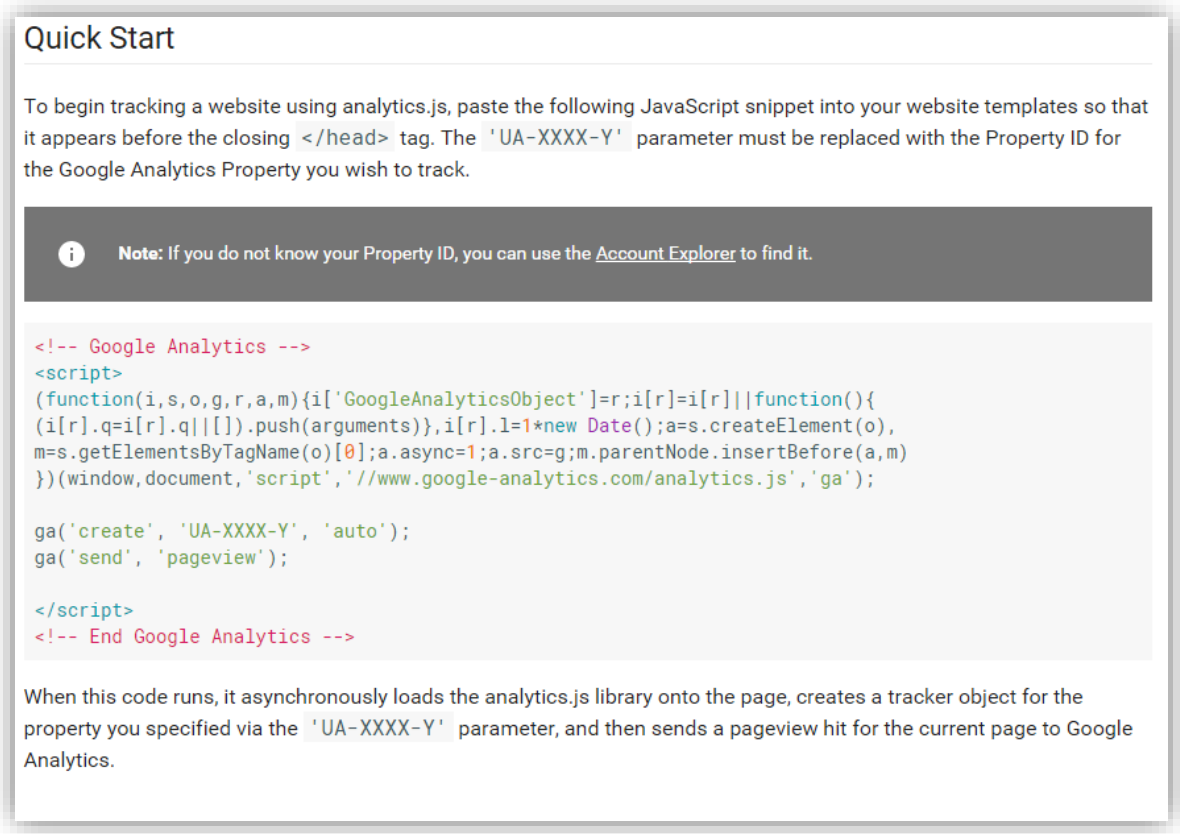
We have presented a new and innovative approach for the design-for-digital workflow, where we open the process to the most advanced techniques available for digital media today. It is based on Open XM -- a technology stack that allows calling upon XMPie personalization from any digital environment.

With Open XM, XMPie has opened the door to the digital world with all its richness, yet, at the same time, has not changed its hallmark solution for print. Now, indeed, print state-of-the-art and digital state-of-the-art live together under the same roof, delivering the full promise of cross-media.

Appendix — Samples

Google Analytics

Adding Google Analytics to a website is achieved by placing Google code snippets into the site's HTML. The following is from Google Developer site.



Quick Start

To begin tracking a website using analytics.js, paste the following JavaScript snippet into your website templates so that it appears before the closing `</head>` tag. The `'UA-XXXX-Y'` parameter must be replaced with the Property ID for the Google Analytics Property you wish to track.

Note: If you do not know your Property ID, you can use the Account Explorer to find it.

```
<!-- Google Analytics -->
<script>
(function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
(i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
})(window,document,'script','//www.google-analytics.com/analytics.js','ga');

ga('create', 'UA-XXXX-Y', 'auto');
ga('send', 'pageview');

</script>
<!-- End Google Analytics -->
```

When this code runs, it asynchronously loads the analytics.js library onto the page, creates a tracker object for the property you specified via the `'UA-XXXX-Y'` parameter, and then sends a pageview hit for the current page to Google Analytics.

Figure 2

Facebook

Introducing Facebook technology into a website is done by placing Facebook-provided code snippets into the site's HTML. The following example is from Facebook's developer site – a cookbook to add the "Like" button to a website.

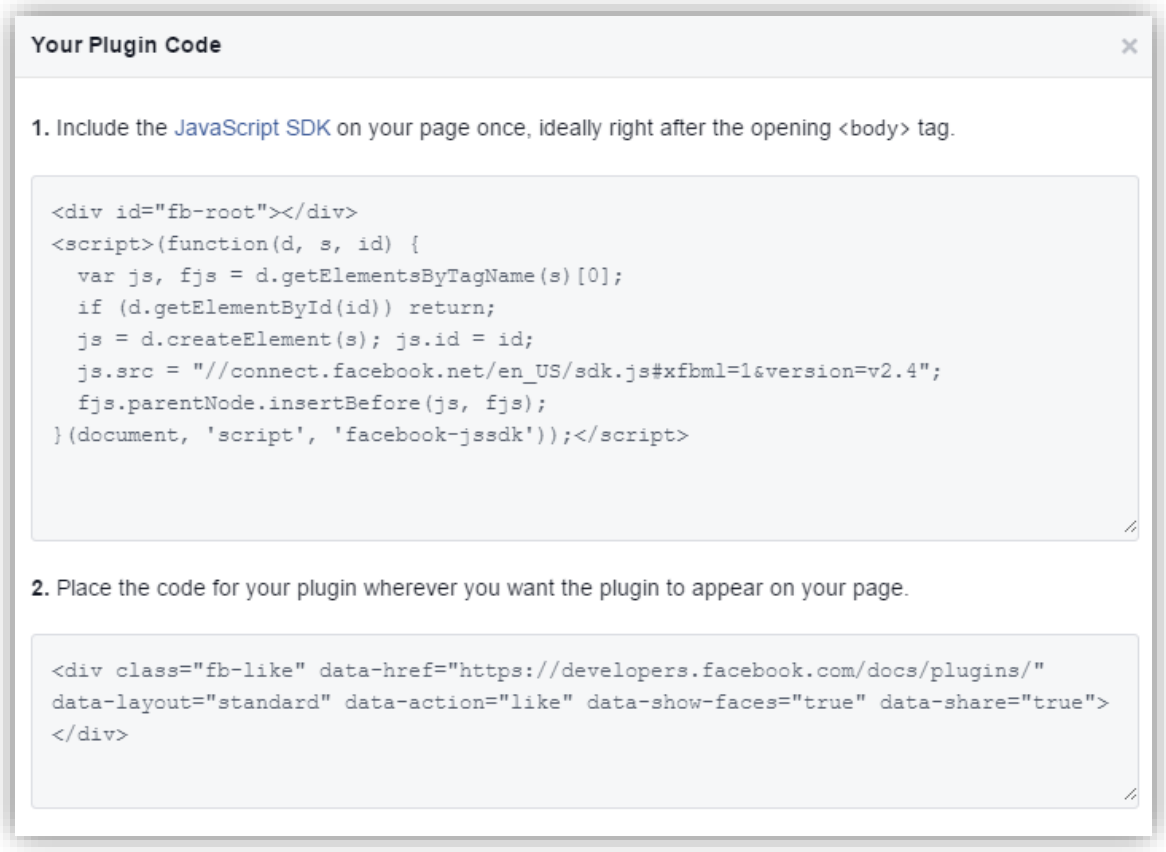
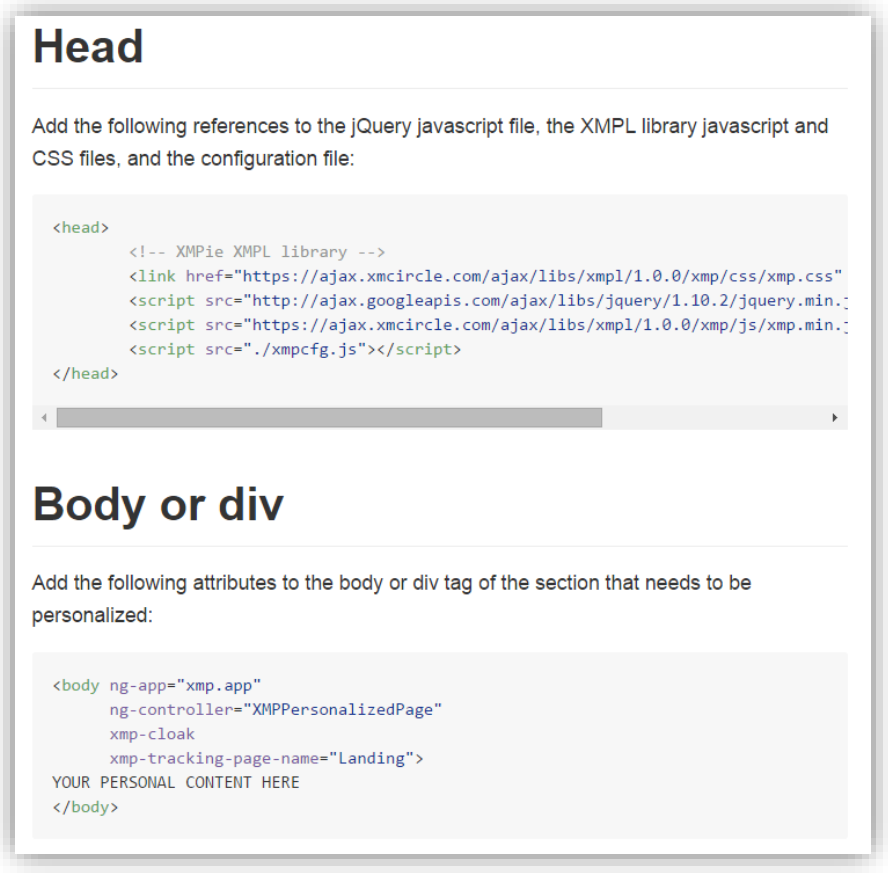


Figure 3

XMPie

Adding XMPie personalization into a website is done by placing XMPie's XMPL code snippets into the site. Here is the code snippet for creating a personalized page:



Head

Add the following references to the jQuery javascript file, the XMPL library javascript and CSS files, and the configuration file:

```
<head>
        <!-- XMPie XMPL library -->
        <link href="https://ajax.xmcircle.com/ajax/libs/xmpl/1.0.0/xmp/css/xmp.css"
        <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.
        <script src="https://ajax.xmcircle.com/ajax/libs/xmpl/1.0.0/xmp/js/xmp.min.
        <script src="./xmpcfg.js"></script>
</head>
```

Body or div

Add the following attributes to the body or div tag of the section that needs to be personalized:

```
<body ng-app="xmp.app"
      ng-controller="XMPPersonalizedPage"
      xmp-cloak
      xmp-tracking-page-name="Landing">
YOUR PERSONAL CONTENT HERE
</body>
```

Figure 4

Appendix — Use Cases Leveraging Open XM

Modern Web Design using HTML5 & CCS3

XMPL was built to work smoothly with HTML5, CCS3, jQuery, and AngularJS and other modern technologies. To personalize a website that wasn't built especially to work with personalization, one just needs to add the XMPL personalization tag attributes.

Powerful Control over Page Rendering and Behavior

You can use ADOR logic to control text, HTML content, image source HTML classes, div visibility, page controller logic, animation, form submission and much more. This is because one can use any logic and control mechanism available in the digital platform that will take actions based on values of variability as provided through the use of the XMPL technology stack.

Responsive Design

Modern websites are expected to provide a great experience both on desktop websites, tablets, and smartphones. Responsiveness is typically achieved by means of CSS3, and because XMPL and CSS3 work smoothly together, the website responsiveness is unaffected by the addition of XMPL, so the website looks great on all devices.

Single Page Application (SPA)

To provide the user with a "Wow" user experience modern websites make use of transitions and other animations. These are not limited to animations within the page but also when moving from one page to another. Instead of a post back that has the user waiting, and then renders the screen entirely with the new page, resulting in a "flicker", with an SPA the new page floats into the screen smoothly using any animated transition. A common way to implement this technique is using AngularJS SPA. XMPL was built to add personalization to an AngularJS SPA website.

Use any Server Technology or None at all

In previous generations of cross-media technology the website had to be implemented using ASP.NET. There is no such restriction in XMPL. This allows users to use any server technology they see fit; for example, to run PHP code on Apache servers. The reason for this is that the XMPL relies on the client side only, which then connects to the XMPL services. It doesn't care where the page originated from – form a PHP Server or and ASP.NET server. This also allows the hosting of an independent HTML page server without any server technology, such as on Amazon AWS S3.

Embed Individualized Content within a Hosting Web Page

A common marketing practice is to ask a user who browses a corporate website for some personal details before letting them download PDF with additional information. XMPie opens new opportunities there because the downloaded PDF can be personalized. However, with previous generation of cross-media this wasn't really feasible, because it required the entire corporate website to run on the XMPie servers with its proprietary technology. A large corporate entity would rarely agree to this. With XMPL though, the corporate website remains

intact. Just like the corporate website developer adds Google Analytics snippets, Facebook Like button snippets to the webpage to get the respective functionality, to add personalization he will need to add the XMPL snippets. This way the above downloaded PDF can be personalized. Furthermore, other sections of the sites can be personalized. For example, the site can display a special offer using uImage personalization.

Use Available Templates

Another gain from the fact that XMPL is based on standard technology, is that one can pick from the numerous HTML website templates that are available on the Internet. Most of these templates are of a very professional standard - employing responsive design, animation, and more. One simply needs to add the snippets to personalize them, and voila.

Round Trip Development

The development of a website, and also a cross-media campaign, is an iterative process. Often the website is created by a 3rd party web development company, and the personalization is then carried out by the service provider. The customer then asks for modification of the content and styling of the page, which requires another iteration. With XMPL the website is easily transportable and can then be sent back to the 3rd party web developer to do the work. The personalization will keep working even when running on the developer machine.

Quick Development Cycles

The need to continuously upload pages to the web server during the development cycle is a hassle. But with XMPL, the development can be carried out on the developer machine file system. Make a change in the webpage text file, save it locally and refresh the browser to see the outcome – that's all it takes – a few seconds work.

Mobile Apps on Apple App Store and Google Play

XMPL has a role to play in Mobile Apps. One can create mobile apps that play a part in the cross-media campaign. One way is to create a responsive or mobile website. Another option is to take that website, convert it to an App using PhoneGap and publish it on the App Store. A third option is to write a native App, and connect to XMPie using REST APIs.

Emails

Email pages are essentially HTML documents with a reduced set of tags and options. XMPL for Email uses a reduced set of the same tagging language, hence one can leverage this knowledge to email channels too. Similarly, one can make use of existing HTML email templates found on the Internet.