

A Case for Smart Clients

While traditional web technologies have revolutionized the way information is delivered to people, they have not revolutionized the general user experience while doing so. Web technologies are now evolving to improve the user experience, with AJAX at the forefront. But when discussing Smart Client technologies, quotes like the following are commonplace: *"Ajax is about a better web, not about making the web better than your desktop."*: *"Don't settle for a browser interface. Give your customers the best possible experience."* This is because client based tools and languages create a richer and more robust user experience simply because they can.

With emerging Smart Client technologies, developers are able to create rich client based applications that can deliver all the ubiquitous internet content that their server based cousins can. The added intrigue is that these applications can continue to operate when not connected to the internet. This is a very important distinction, as anyone who has depended on an internet based application during server unavailability can attest.

With any application that is not 100% standalone, there is a 'Client' side of the application (that interacts with the user) and a 'Server' Side (the database, internet content etc). With traditional web-based applications, both the 'Server' side and the 'Client' side reside remotely away from the users' computer. The 'Client' side (user interface) of the application is delivered to the user one HTML page at a time from the web server. This is both its' allure (from a deployment perspective) and its' limiting factor (both from a usability and accessibility perspective).

So, what is a Smart Client?

Here is Microsoft's definition: Smart clients are easily deployed and managed client applications that provide an adaptive, responsive and rich interactive experience by leveraging local resources and intelligently connecting to distributed data sources.

Here is a definition by required characteristics of Smart Client applications:

1. **Utilizes Local Resources:** A smart client application always has code artifacts on the client that enable local resources to be utilized.
2. **Connected:** Smart client applications are never standalone and always form part of a larger distributed solution. The application can interact with a number of Web services. The application has access to specific services that provide deployment and update services.
3. **Offline Capable:** Smart client applications can be made to work even when the user is not connected. For applications running in occasional or intermittent connectivity situations, such as those used by traveling workers or even those running on laptops, tablets, PDA's, and so on, where connectivity cannot be guaranteed at all times, being able to work while disconnected is essential.
4. **Intelligent Install and Update:** Smart client applications manage their deployment and update in a much more intelligent way than traditional rich client applications. Application artifacts can be deployed using a variety of techniques, including simple file copy or download over HTTP.

These technologies create exciting possibilities for developing applications that satisfy the market for both rich, friendly user applications and seamless connectivity to distributed resources as needed. The implementation of these technologies is not without challenge. But many analysts believe that the pendulum has swung back towards rich clients and away from browser-based applications and that the market will drive these Smart Client technologies to the forefront.

Paul Cafarella
Joe Perzel & Associates