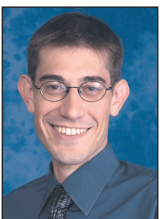


# Compressing Apps for Zippy Network Performance

**N**etworks grow ever more complicated. From [Ethernet](#) to [WiFi](#), consumers are connecting more devices to the Internet worldwide and in more complicated ways. Each technology differentiates itself with subtle advantages, but every networking method has limits. A network can only transmit so much information per second, which means each network has a limited bandwidth. A state-of-the-art fiber optic connection transmits an HD movie in seconds while it might take a cell phone hours to download a song over GPRS.

The high cost of networking makes efficient use of bandwidth important. Economic costs are relatively small and getting smaller, but a user paying for a cell phone data plan might disagree. For example, Amazon charges around 12 cents per transmitted gigabyte, so using those bytes wisely will save money. Cell phone providers often charge for data sent to a phone, so compression will save money. Beyond cost savings, minimizing the number of bytes transmitted saves something far more valuable - time. Users have no interest in watching screens load. The more narrow the connection, the more critical it is to be efficient. Cell phone users want their data immediately.



**By Jason Foutz**  
Software Programmer

Your network provides access to your applications. Anyone connected to your network could launch your application with Java Web Start, if desired. If you make your application available on the Internet, people could connect with almost any device – smartphone, tablet, laptop, or desktop – to your application. Perhaps they are using a fast connection on their desktop at work. They could be relying on WiFi at a customer's job site. They might even be using a cellular connection built into a tablet from the beach. It's impossible to predict all the different types of network clients that your application must deal with.

From version 12 onward, BBJ® makes use of Java's [Pack200](#) technology, which cuts the bandwidth required for initially launching a thin client on first install or after an upgrade by over 60%. Jetty, the built-in Web Server that ships with BBJ, automatically compresses jars with Pack200 reducing, for example, BBJThin client .jar from over 6 MB to less than 3 MB. On a local network, this compression saves users a few seconds; over the Internet, it saves minutes.

BBJ's Web Server also uses [Gzip](#) compression for many resources. Gzip speeds up both BUI and static resources served from the htdocs directory. Gzip provides the biggest benefit when used on text files. All standard web files such as HTML, CSS, and Javascript are readily compressible by Gzip. Gzip does a really fantastic job with core BUI files, reducing 1.3 MB files down to less than 400 KB. The payoff of this huge saving is most valuable and most visible on smartphones. Decreasing the amount of required bandwidth saves customers significant time and money resulting in a greatly improved user experience.

Compression in BBJ happens behind the scenes. The Jetty Web Server examines your resources and compresses them as needed. Jetty compresses any jar required by the Web Start thin client before sending it over the network. Compression reduces the size of the JavaScript files that support a BUI application by 60%. Jetty also compresses additional resources such as custom CSS or a customized HTML index page. All of this happens behind the scenes without any extra effort. Jetty provides all of the speed without any administration overhead.

## Summary

BASIS takes advantage of several industry standard techniques to make your business application as small and nimble as possible on any network –

- Pack200 drastically reduces network resources for downloading Web Start clients
- Gzip speeds up BUI
- Jetty compresses virtually every resource served with state-of-the-art tools to save your customers time, and money.

Compress coal and you get diamonds. Compress web resources and you get happy customers. Priceless. ■