

NOVEL ADVANCEMENTS IN INTERNET-BASED REAL-TIME DATA TECHNOLOGIES

Gerry Myers
AZ Technology, Inc.

Draft 2

Title: Novel Advancements In Internet-Based Real-Time Data Technologies
Author: Gerry Myers (AZ Technology, Inc.)
For: SpaceOps2002 Conference (October 9-12, 2002)
Track 2: Ground Segment Engineering and Architectures Session

Abstract

AZ Technology has been working with NASA MSFC (Marshall Space Flight Center) to find ways to make it easier for remote experimenters (RPI's) to monitor their International Space Station (ISS) payloads in real-time from anywhere using standard/familiar devices. That effort resulted in a product called "EZStream" which is in use on several ISS-related projects. Although the initial implementation is geared toward ISS, the architecture and lessons learned are applicable to other space-related programs.

This paper begins with a brief history on why Internet-based real-time data is important and where EZStream or products like it fit in the flow of data from orbit to experimenter/researcher. A high-level architecture is then presented along with explanations of the components used. A combination of commercial-off-the-shelf (COTS), Open Source, and custom components are discussed. The use of standard protocols is shown along with some details on how data flows between server and client.

Some examples are presented to illustrate how a system like EZStream can be used in real world applications and how care was taken to make the end-user experience as painless as possible. A system such as EZStream has potential in the commercial (non-ISS) arena and some possibilities are presented. During the development and fielding of EZStream, a lot was learned. Good and not so good decisions were made. Some of the major lessons learned will be shared. The development of EZStream is continuing and the future of EZStream will be discussed to shed some light over the technological horizon.

Background

AZ Technology is familiar with the data needs of experimenters and researchers. AZTek has built and flown experiments/instruments on-orbit and has had to deal with live on-orbit data. AZTek found that during a mission, personnel were confined to the "data room". All data came to one or two computers and if data was to be monitored, team members had to stand in front of one of them. AZTek also wanted to show the world what great things they were doing. Of course, static charts and graphs could be presented with old data but more powerful presentations would use live data direct from orbit.

AZTek decided a new system was needed and came up with a few simple requirements. First, live data should be viewable from anywhere (or nearly anywhere) in the world. Experimenters shouldn't be stuck in the data room. What if essential personnel were at home or on business travel when something went wrong? Team members should be able to check first-hand what was happening to the payload wherever they happened to be. Second, the system should be easy to use and setup. Such a system would serve Education/Public Outreach requirements if it was easy enough for schools and museums to access. Third, the system should control access to the data displays. When companies spend millions of dollars to develop and fly experiments they don't want proprietary data immediately available to all their competitors. Therefore, the system had to control who could see the data but still allow easy access for Education/Public Outreach purposes. Finally, since the first requirement of "anywhere in the world" pretty much dictated use of the Internet, the system had to be firewall friendly to be widely accessible. Most corporate,

NOVEL ADVANCEMENTS IN INTERNET-BASED REAL-TIME DATA TECHNOLOGIES

Gerry Myers
AZ Technology, Inc.

government, or campus networks are protected by a firewall. Many great schemes of data distribution fall short when they hit the firewall. Firewalls are setup to keep people out and system administrators are often hard to convince to open non-standard ports through their firewall.

AZTek presented the ideas to NASA MSFC and received a Small Business Innovation Research (SBIR) grant to research the technologies required for such a system. With the knowledge gained from that grant, EZStream was developed to answer the above requirements. Although this paper isn't a sales pitch on EZStream, it does use EZStream as a case study on how AZTek developed such a system, the technologies used, and lessons learned along the way. This information may help others who look into developing similar systems. With that aside, AZTek is marketing EZStream and can be contacted for sales or customization.

EZStream's Place in ISS Data

Data from ISS passes through a lot of systems before it reaches the destination desktop. Figure 1 shows a simplified view of the data path and where EZStream fits.

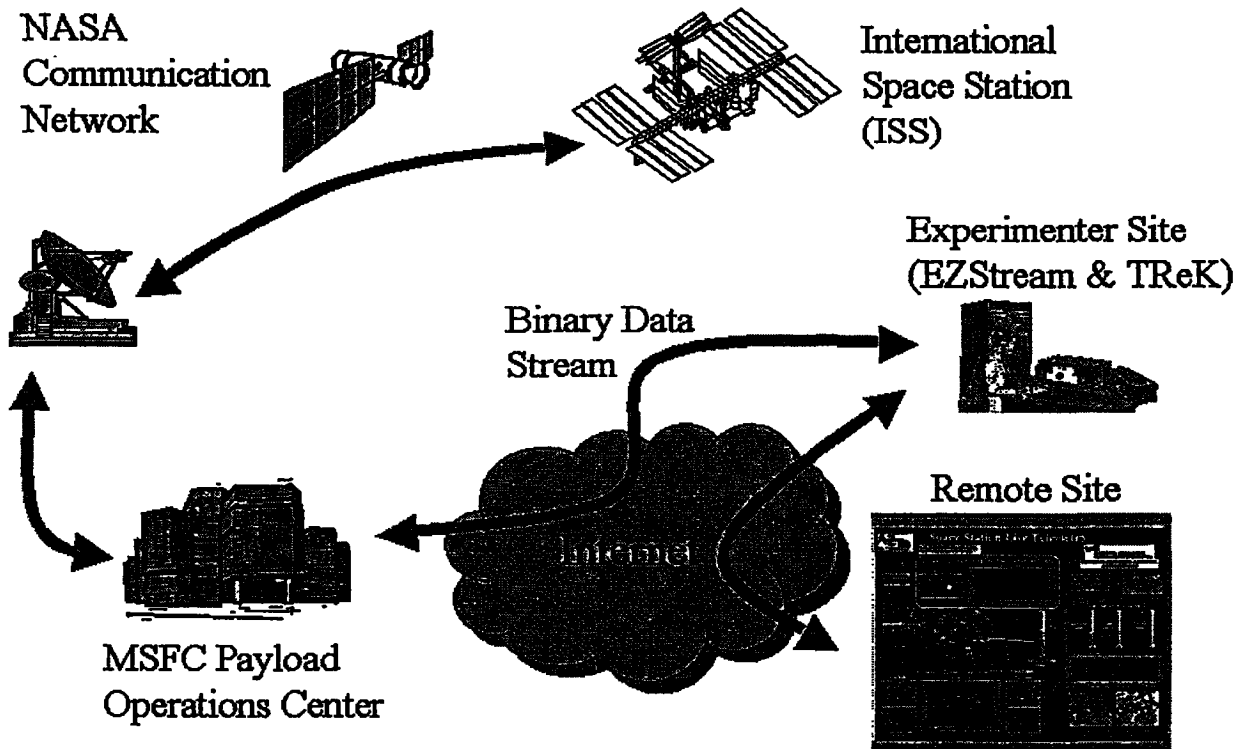


Figure 1 Where EZStream Fits in the ISS Data Flow

EZStream is typically used at the experimenter site. MSFC Payload Operations Center sends the appropriate binary data stream from their data processing servers to an Internet Protocol (IP) address at the experimenter site. That IP address is associated with a Telescience Resource Kit (TReK) workstation. TReK was developed by MSFC to parse, process, and convert the data from the binary stream into individual measurement parameters. TReK provides an Application Programming Interface (API) to allow 3rd-party systems, such as EZStream, to request individual data items from the binary data stream. EZStream runs on the same computer as TReK and uses the TReK API as its data source.

NOVEL ADVANCEMENTS IN INTERNET-BASED REAL-TIME DATA TECHNOLOGIES

Gerry Myers

AZ Technology, Inc.

EZStream formats the data into web pages and servers those pages out to Internet users. The end-user needs only a standard browser and an Internet connection to be able to access the web page data displays. The ISS data within the web page is continuously updated with fresh data from the EZStream server.

In addition to running EZStream at the experimenter site, it could be run closer to the source of the data stream. In the case of ISS, at the Payload Operations Center. A data producing center, be it NASA, Army, Air Force, Navy, or corporate, could provide EZStream web page displays directly to their customers/users for those cases where ease of use and accessibility are paramount.

EZStream's Architecture

EZStream is a client/server system made up of COTS, Open Source, and custom written software components. Figure 2 shows an EZStream component diagram.

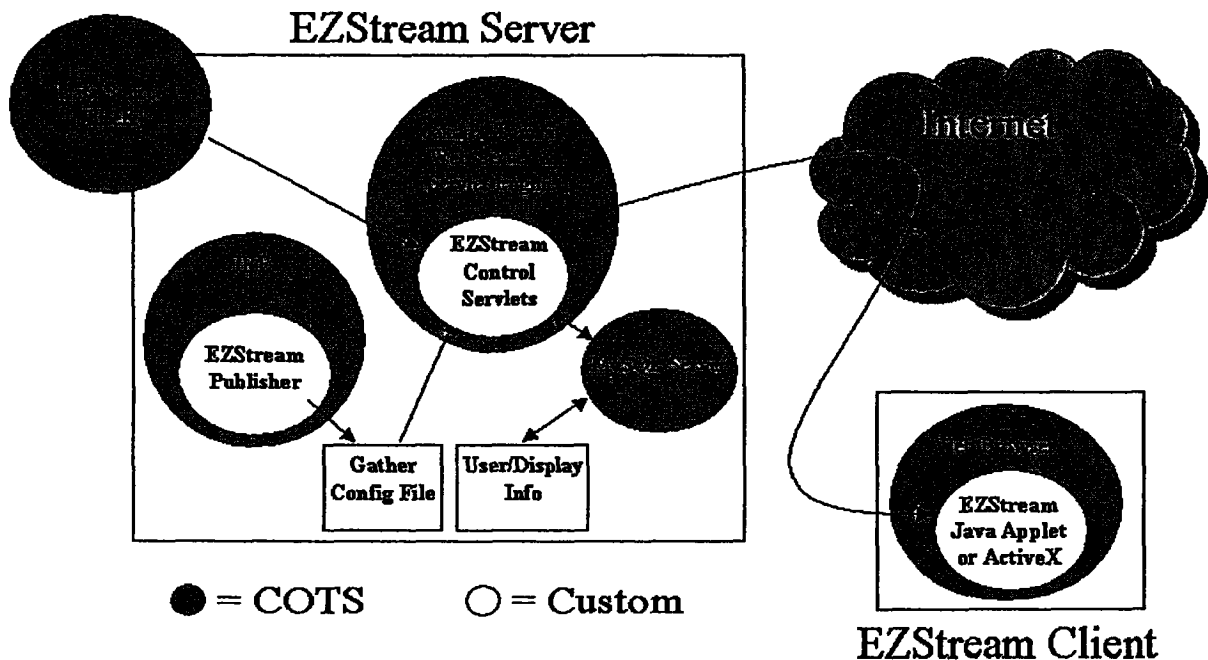


Figure 2 EZStream Component Diagram

The EZStream server component is written as Java Servlets and therefore requires a web server and servlet engine. The Open Source server Jakarta-Tomcat was chosen for that task. EZStream keeps a database of all the authorized users and the data displays they are allowed to access. The Open Source database MySQL was chosen for that function. Standalone Java applications were required to support various operations. Sun's Java Runtime Environment (JRE) handles them. On the client side, only a standard web browser is required. The EZStream client component comes in two flavors: JavaBean or ActiveX. The JavaBean client component is used to write Java Applets to be embedded in the web pages. Applets can be built with any Java Integrated Development Environment (IDE). The ActiveX client component is used to write encapsulating ActiveX display components which are then embedded in the web pages. Any MS-Windows capable IDE can create the encapsulating display ActiveX such as C++, Delphi, Visual Basic.

NOVEL ADVANCEMENTS IN INTERNET-BASED REAL-TIME DATA TECHNOLOGIES

Gerry Myers
AZ Technology, Inc.

As shown in Figure 2, data flows from the data source to the EZStream Control Servlets. In the case of ISS, this data source is TReK. The Control Servlets create measurement data packets that are unique for each display and serve them out to waiting clients to update their display web pages.

COTS

EZStream makes great use of COTS and Open Source products. As with any project, there is a trade-off between fitting COTS components into the design or writing custom code. COTS has the benefit of being done already and being fairly well debugged. Custom components have the benefit of satisfying the exact need but require development time. There are a lot of high quality COTS software components on the market in the form of JavaBeans and ActiveX components. If well defined interfaces can be designed between custom components, COTS components can more easily be integrate to fill the gaps.

For EZStream, instead of writing custom web services, the server was written as Java Servlets and the Open Source COTS Jakarta-Tomcat web and servlet engine was used to execute them. Likewise, a custom database or simple file system could have been written to keep track of user and display info. However, it was decided to use Java Database Connectivity (JDBC) calls to handle that info. That decision opened the door to any JDBC-compliant database system. MySQL was chosen since it was Open Source and had JDBC drivers.

Although Java isn't Open Source, it can be used free of charge. There are some very fine Java IDE's on the market. There are also several free or Open Source Java IDE's. Java can also be written with nothing more than a simple text editor and compiled with the freely available Sun Java Development Kit (JDK).

Standards

Initially, EZStream was not "standards" conscious. To shorten the development process, a COTS product from National Instruments (NI) called DataSocket was purchased to handle all the client-to-server communication. It worked well but used a proprietary protocol and ports. It didn't take long to realize that DataSocket had a hard time communicating through firewalls due to the proprietary port selection. Since a firewall friendly system was one of the requirements, other options had to be considered. Custom client/server communication was written based on standard HTTP messages GET and POST and using the standard HTTP ports. Client web browsers speak HTTP fluently and firewalls generally trust HTTP traffic and let it pass through. Therefore, by sticking with standards (i.e. HTTP), EZStream display web pages became easily accessible to users behind nearly any firewall.

Data Transmission/Management

The EZStream client/server communication makes efficient use of bandwidth as shown in Figure 3. To start the client/server conversation, the client's web browser makes a request for a standard web page address. The web server locates the page and sends it back to the client's browser. Once the web page is loaded in the client browser, the Java Applet or ActiveX display component starts up. The client component repeatedly makes new requests to the server for fresh data and the server sends it packaged as standard HTTP messages. The benefit of this approach is that the actual web page and Applet (or ActiveX) is transmitted only once. From then on, only the data is passed between server and client. This process requires much less bandwidth than if the entire web page had to be refreshed continuously.

NOVEL ADVANCEMENTS IN INTERNET-BASED REAL-TIME DATA TECHNOLOGIES

Gerry Myers

AZ Technology, Inc.

Through the functionality of TReK, data can be processed and stored/archived on the server if needed. Likewise, through the use of ActiveX components or signed Java Applets, data can be stored on the client as well as being displayed.

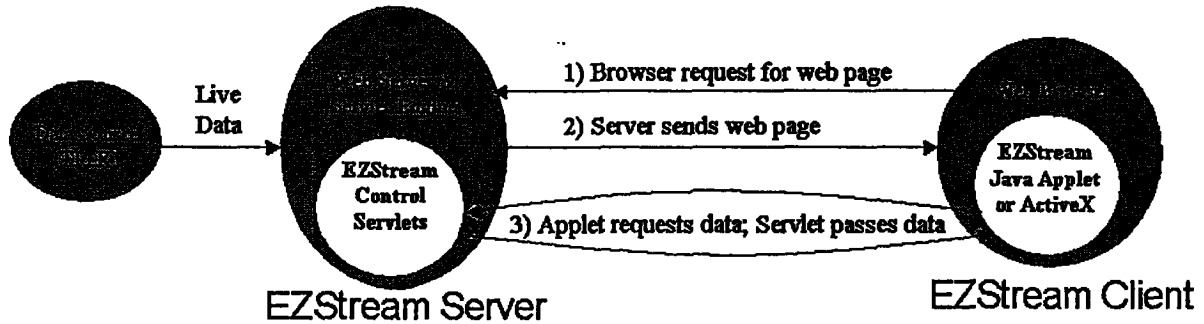


Figure 3 Client/Server Data Flow

EZStream in Action

EZStream was chosen to feed live ISS data to a kiosk at Discovery Place Museum in Charlotte, North Carolina. Figure 4 shows the exhibit layout, kiosk, and EZStream display. The display hardware is a standard PC running a standard web browser. The browser is pointed to an off-site EZStream server. As mentioned, the display web page and automatically refreshing live data travels across the public Internet. Museum patrons can stand and watch live data from outer space as it's happening.

Similar kiosks may satisfy Education/Public Outreach requirements for other projects.

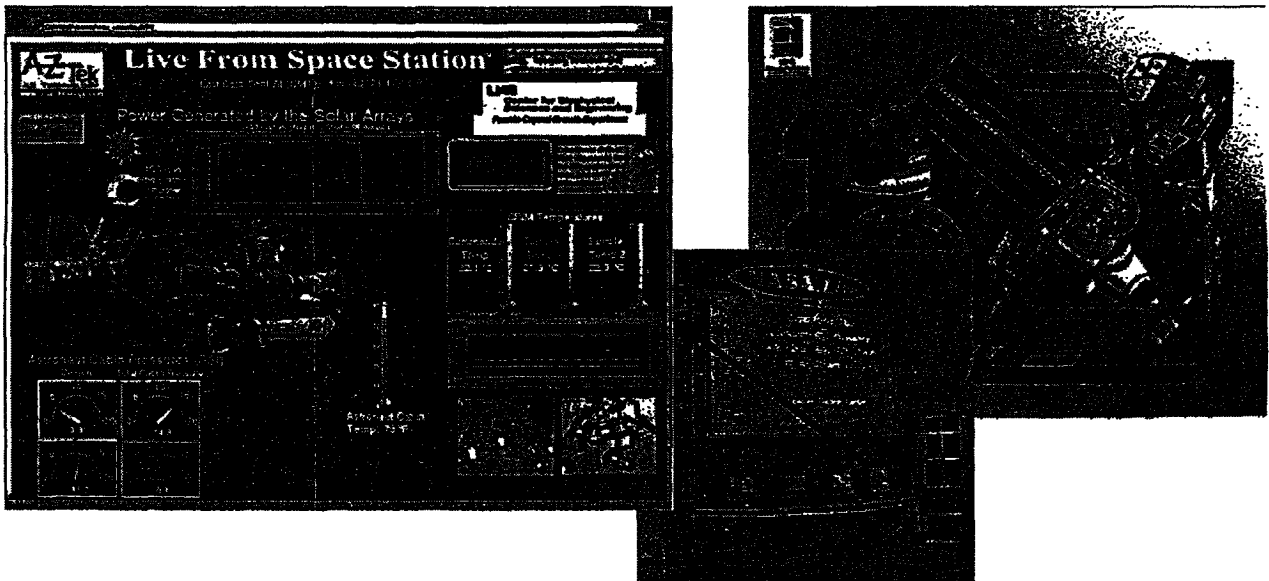


Figure 4 Discovery Place Museum Kiosk

The Iterative Biological Crystallization (IBC) project chose EZStream to drive a portion of their end-user display. Figure 5 shows their demo display with the circled area controlled by EZStream. This display web page shows the versatility of EZStream. EZStream can drive the entire display as in Figure 4 or only a portion as in Figure 5. The rest of the IBC display is driven

NOVEL ADVANCEMENTS IN INTERNET-BASED REAL-TIME DATA TECHNOLOGIES

Gerry Myers
AZ Technology, Inc.

by other systems including local databases and imaging systems. Whether EZStream is the only component on the web page or one of many, the EZStream client-to-server communication operates the same without interfering with other components.

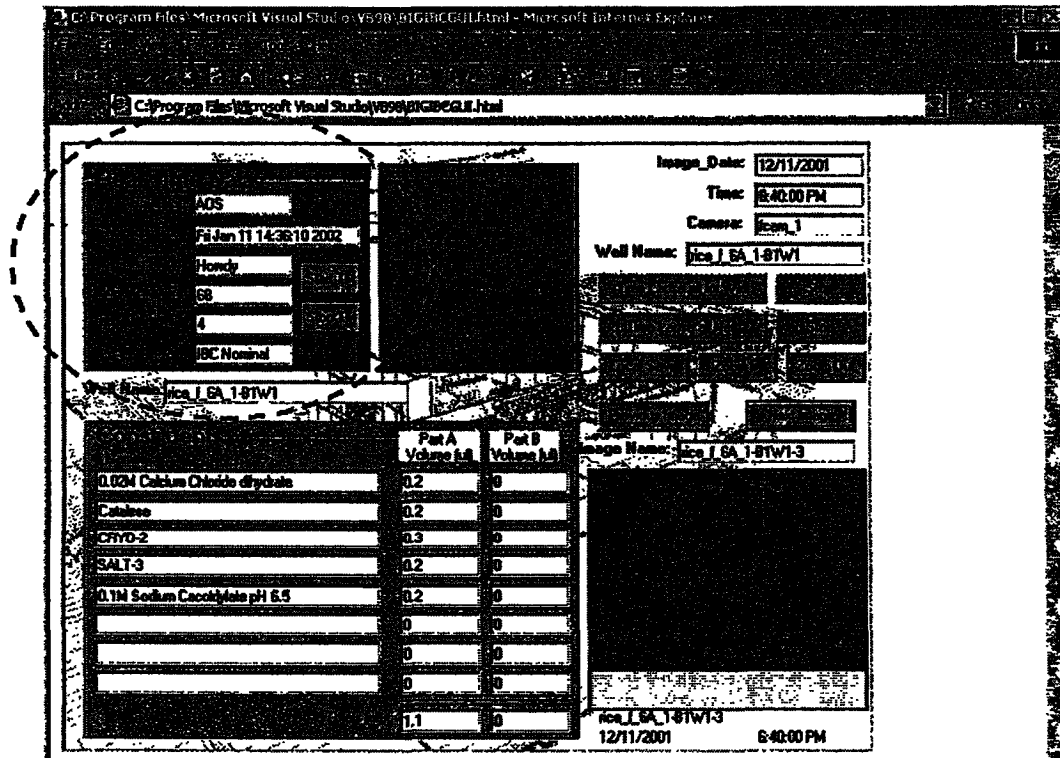


Figure 5 IBC Display

EZ on the End-User

One of the initial requirements for EZStream was to be easy for the end-user to setup and operate. A lot of thought went into the user interface and the procedure for accessing the display web pages. The process had to be extremely functional for use by demanding experimenters and researchers while at the same time simple enough for Middle & High School students to grasp when used for Education/Public Outreach. Figure 6 shows the simple two-step process for pulling up live EZStream display web pages. The user opens a standard browser (Netscape in this case) and types in the web address of the EZStream server (i.e. <http://www.somewhere.com/ezstream/login.html>). The EZStream login page is presented as shown on the left. After entering a valid EZStream username and password (established by the local EZStream administrator), the page on the right is returned. This page presents a list of all the display web pages the user has been authorized (by the local EZStream administrator) to see. Selecting the desired display from the list and clicking the button opens up the actual display web page similar to Figure 4 or 5. To recap, the end-user simply has to login and select a display. It couldn't be easier.

NOVEL ADVANCEMENTS IN INTERNET-BASED REAL-TIME DATA TECHNOLOGIES

Gerry Myers
AZ Technology, Inc.

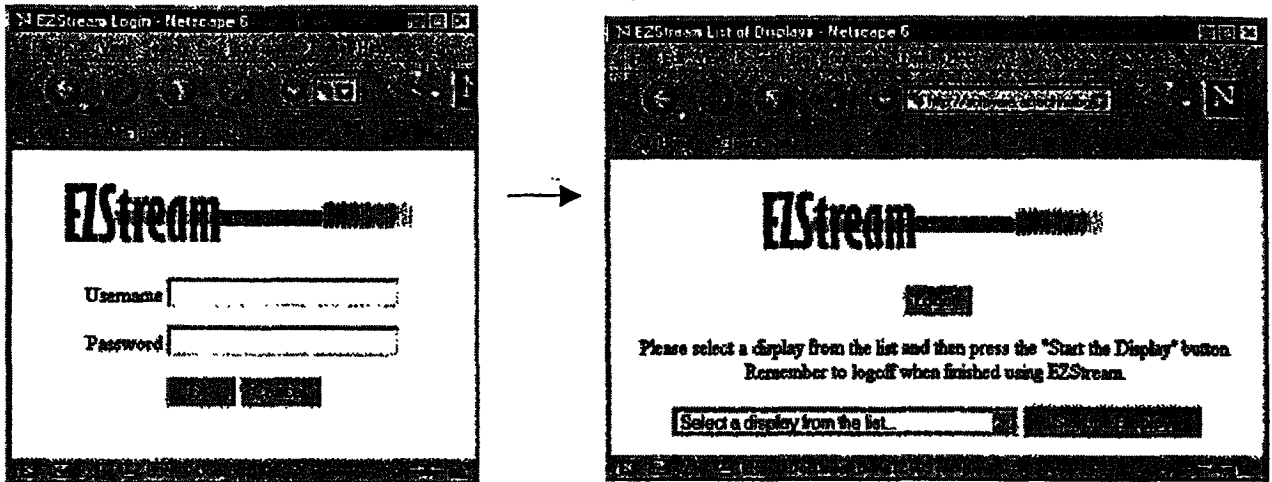


Figure 6 Ease of Use

Commercial Applications

The applicability of EZStream doesn't end with ISS. EZStream can be used in any situation that requires easily viewing live data across the Internet or Intranet. Figure 7 shows a scenario where EZStream can be used in a manufacturing application to distribute live factory floor data to management, customers, and suppliers. They would all have their own EZStream user accounts and have web pages built to display the data in a form that is meaningful for them. For example, the customers wouldn't see the same displays as the suppliers or management.



Figure 7 EZStream Commercial Applications

NOVEL ADVANCEMENTS IN INTERNET-BASED REAL-TIME DATA TECHNOLOGIES

Gerry Myers
AZ Technology, Inc.

Lessons Learned

Developing EZStream was a learning process as any project would be. The main thing to learn from the experience is to use COTS whenever possible. There are many COTS components (JavaBean and ActiveX) on the market. Custom components should be developed only when the trade-off of cost vs. time warrant or if the application is so unique that nothing else can be found. Second, Open Source components should be used where possible. Open Source used to imply low quality "hacked" products. Open Source projects now produce high quality, well documented, and tested systems at very low or no cost.

Third, for applications that deal with data transmission across the Internet, HTTP should be used. It is a well established protocol and most firewalls allow it through by default. Fourth, using Java for both the client and server components allows server platform and client browser independence. Java provided a faster and safer project life-cycle than other languages. The client's alternative ActiveX component was developed mainly to allow Visual Basic web developers to continue using the tools they were familiar with.

EZStream Future

EZStream is a work-in-progress and enhancements are planned. EZStream incorporates some security measures in that users are required to login before accessing their displays. That's a good start, however, the actual transmission of display web pages and live data occur in plain text across the Internet. Future releases of EZStream will incorporate HTTPS for all transmissions between client and server. HTTPS is the secure version of HTTP and has been used for years for things like web site credit card purchases. HTTPS is built into most web servers and browsers so its inclusion in EZStream will be cost effective.

At present, some kind of programming experience is required to build the web page displays. The displays can be built with Java, C++, Delphi, Visual Basic, or any other tool that can produce a Java Applet or ActiveX component that can be embedded in a web page. Obviously, experimenters have great knowledge about the subject matter related to their on-orbit payload. However, that knowledge may not include software programming. Likewise, Middle/High school students may not have the programming ability to develop their own display web pages when EZStream is used for Education/Public Outreach. Therefore, a non-programming web page development tool is planned for future releases of EZStream. That tool will present a simplified interface including drag-and-drop technologies to allow persons without any programming knowledge to build their own display web pages.

Currently, EZStream is a data monitoring system. Live data flows from the server to the client. With some enhancements, EZStream can be made to control as well as monitor remote processes. GUI buttons, switches, dials, etc. can be placed on the display web pages to allow user actions to initiate remote control of on-orbit or factory floor processes. This remote control path will not supercede or override existing control safeguards but will in-fact make use of them. Therefore, in critical situations like on-orbit payload commanding, authorities responsible for the final release of commands to orbit will still have the final say before commands leave the ground.

For the "gee-whiz" effect, client components will be created that allow EZStream display web pages to be viewed on wireless hand-held devices such as Personal Digital Assistants (PDA) and cell phones. Don't you know that as soon as you get on the airplane for that business trip or step on the golf course on your day off, you'll get a call from the office that something has gone wrong with the payload. Instead of dropping everything and heading for the office, you will be

**NOVEL ADVANCEMENTS IN INTERNET-BASED REAL-TIME
DATA TECHNOLOGIES**

Gerry Myers

AZ Technology, Inc.

able to pull out your wireless PDA and view the live payload data yourself right where you stand.
You'll be able to diagnose the problem and get back to what you were doing.

Contact Info

You can contact the author at:

Gerry Myers

AZ Technology, Inc.

7047 Old Madison Pike, Suite 300

Huntsville, AL 35806

256-837-9877 x112

gerry@aztechnology.com

<http://www.aztechnology.com/>