Astronomical Data Analysis Software and Systems IV
ASP Conference Series, Vol. 77, 1995
R. A. Shaw, H. E. Payne, and J. J. E. Hayes, eds.

Designing an Open E-Mail Interface to a Data Archive

J. Richon

Space Telescope Science Institute, 3700 San Martin Drive, Baltimore, MD 21218

Abstract. An active and useful data archive often has to support two conflicting goals: providing an easy way to retrieve data, and preventing people from getting proprietary data. The retrieval method should be insulated from the user interface so that changes to one system do not force changes to the other. This paper discusses the issues associated with data retrieval and system isolation, and how the Space Telescope Science Institute developed a flexible and open means to retrieve data from the Hubble Data Archive that can support almost any user front-end system.

1. Introduction

In 1993 the Hubble Data Archive group reviewed the original design of the Space Telescope Data Archive and Distribution System (STDADS) for retrieving data. The design called for any user interested in retrieving data to have an account on the operational archive system, and to use a command line syntax to retrieve data. This made it impossible to use a query tool to easily retrieve data. After reviewing the current system, we decided to create an open interface to the retrieval system for the archive.

2. Design Goals

Our first goal was to make the new system simple so that it would be easy to get public data out of the archive. Next, it had to be open so that almost anybody could get public data. Yet it had to be secure so that proprietary data would never be released to the wrong people. Finally, we wanted a unified system that worked with StarView, our user interface for querying the database of observations. While not a design goal, we also wanted to find a solution that would meet our goals quickly with a minimum impact on the current system.

3. Assumptions

In creating our design we recognized two assumptions we had made. First, that the user, or the user's system, knew what datasets they wanted. In other words, the retrieval system would not support queries, only requests for specific datasets. The second assumption was that the user had access to the Internet. Since our initial design, access to the Internet has become even easier.

4. The Design

To achieve our design goals we defined a fairly simple syntax to be used to request datasets. Basically, the user must specify an archive username, where the data is to be delivered, and a list of datasets to be retrieved. Additionally there are a few global settings that can be used to simplify the request of datasets.

Once the request text is complete, it is sent as an e-mail message to an account at ST ScI where a mail daemon reads, parses, validates, and converts the message into the STDADS commands to retrieve the data. The daemon will send a message to the archive user's registered e-mail address with the request identification number if all is successful. After the datasets have been retrieved from the optical disks, they are then transfered, using FTP, to either the user's host computer, or to one of the staging systems (*stdatu.stsci.edu* or *stdata.stsci.edu*) at ST ScI. When the last dataset has been transfered, a message detailing the status of the transfer is sent to the user's registered e-mail address.

To retrieve proprietary data, a password verifying the archive username must be supplied. No password is required for retrieving non-proprietary data. To ensure the security of the password, the message must be encrypted, and currently StarView is the only tool which can send an encrypted message to STDADS.

5. The Reasoning Behind the Design

We recognize that e-mail is not a secure way to send passwords in the clear. For access to public data we do not need to validate the archive username since any archive user can fetch public data, and so a password is not required. A password is required to verify the archive user to retrieve proprietary data. Additionally, the request must include a private host computer, account, and password as part of the specification of the destination for the data. It is the user's responsibility to properly secure the destination directory.

As mentioned above, we send all confirmation mail to a registered e-mail address. This is done so that if an unscrupulous person sends a request using another archive user's name, the rightful owner of the account will be informed of the activity. Users who know they are going to use StarView only, and do not want to worry about unauthorized use of their accounts, can request that a password be required for all requests made for their archive user name.

Once we have a valid archive user name, the mail daemon will perform several checks on the user account. It will determine the level of access the user has to the data. If the user is privileged, they can retrieve any data from the archive. This privileged access can be revoked for a request if the request attempts to transfer the data to a public system, or if the archive password is missing. In this case any non-proprietary data requested will be retrieved.

All user requests are then checked to see if they have exceeded certain limits. There is an expiration date for the account, and a lifetime limit on the amount of data one can request. There is a limit on the total number of requests a user can currently have in the system, and a limit on the number of bytes that can be transfered in a day. These checks are done to protect STDADS from being overwhelmed by a single user.

6. How We Met Our Goals

Our design goals are to have a system that is simple, open, secure, and unified. The syntax we created is simple and straightforward, and can easily be generated by a user, StarView, or another query tool. By using e-mail and FTP for requesting and transferring data, the Hubble Data Archive is open to anyone on the Internet with an archive account at ST ScI. Security of proprietary data is preserved by various checks in the system, and the requirement for encrypted requests. Since StarView can generate and send the request for a user based upon the selections, we have achieved a unified query and retrieval system.

7. What We Learned

In adding a new layer or interface, we found that by making a clean break between the two systems we insulated the internal retrieval mechanisms from the external interface. This simplifies modifying the STDADS system without having to notify users of the changes, and to enhance the request interface without having to update STDADS. One of the advantages to our approach of writing a separate interface was that we did not have to make extensive changes to the STDADS system to support the mail interface. In doing this we were able to add a great deal of new capability without the risks associated with modifying large amounts of existing code.

In designing the message syntax, we asked the people who would be making requests of the archive to help us refine the syntax and define what the system should deliver. By getting users involved in the design, the system should meet the needs and expectations of most of our archive users. We intentionally kept the syntax of the request simple and obvious so that it would be easy for users to write their own requests, and so that other query tools could easily generate a request message. Having a simple syntax also makes it easier on our part to test and verify the correctness of the system.

As we moved from design to implementation we discovered that by taking advantage of other tools that we could easily add new functionality. For example, if the STDADS system is busy, we want to queue user requests to be processed later. Rather than writing our own message queuing routines, we take advantage of the mail system's natural queuing capabilities and move requests to various mail folders depending upon the action that needs to be performed.

Probably the most surprising thing we learned is that once you have created a new tool, other people will find new, and unexpected ways, to use that tool. Other groups within the STDADS project have started to incorporate the e-mail interface into their designs as an easy way to retrieve data. It is hoped that people outside ST ScI will also find interesting ways to use the e-mail interface as an extension of their catalog query tools.

As of this writing the the e-mail interface has not been made available to the public. The Space Telescope Science Institute Archive[1] home page will be updated to point to the specifications once they are made public.

---

[1] http://www.stsci.edu/archive-html/archive.html