# Concepts of the NIST EXPRESS Server

Don Libes

Factory Automation Systems Division
National Institute of Standards and Technology
Gaithersburg, MD 20899

## Abstract

*The National Institute of Standards and Technology (NIST) has built numerous software toolkits and applications for manipulating STEP and EXPRESS data. The NIST EXPRESS Server is a computational facility at NIST, which provides the ability to run these toolkit-based applications remotely without installing them locally. Users e-mail EXPRESS Schemas and other data files to the Server. The Server runs the requested applications on the files and returns any diagnostics or output, also by e-mail. Applications requiring interaction can either be returned via e-mail so that they can be run locally, or run remotely by telnet or rlogin across the Internet.*

*Access to the EXPRESS Server is available at zero cost to anyone who can send e-mail. No initial registration is required. Use is anonymous by default, however it is possible to use the Server as a collaborative testbed in which case results can be immediately shared with other Server users. The Server is capable of restricting file access to one user or a subset of users. It is also possible to make files publicly available. The Server maintains many STEP-related standards and draft standards for public access. Machine-processable standards such as STEP schemas can be incorporated automatically when processing user files even if they are not publicly available.*

*The Server dramatically lowers the traditional start-up cost and manpower required to obtain and install STEP and EXPRESS tools as well as the continuing support costs to upgrade and maintain the software, by leveraging NIST research, software support and installation, and computing facilities. The Server enables people to experiment or demonstrate STEP without a significant investment of time and money, allowing them to build experience and make informed decisions about their future needs for STEP.*

*Keywords: EXPRESS compiler; implementation; National PDES Testbed; PDES; STEP, CAD, CAM*

## 1: Background

The PDES (Product Data Exchange using STEP) activity is the United States' effort in support of the Standard for the Exchange of Product Model Data (STEP). STEP is an emerging international standard for the interchange of product data between various vendors' CAD/CAM systems and other manufacturing-related software [1]. The National PDES Testbed has been established at the National Institute of Standards and Technology (NIST) to provide testing and validation facilities for the emerging standard. The Testbed is funded by the Continuous Acquisition and Life-Cycle Support (CALS) program of the Office of the Defense CALS Executive (ODCE).

As part of the testing effort, NIST is charged with providing software for manipulating STEP data. Provided in the form of tools and toolkits for building new tools, the software is research-oriented and evolving. This document is one of a set of reports which describe various aspects of the software [2].

## 2: Introduction

The National Institute of Standards and Technology (NIST) has built numerous software toolkits and applications for manipulating STEP and EXPRESS data [3]. The NIST EXPRESS Server is a computational facility at NIST. The Server provides access to these tools so as to ease experimentation, use, and transition to STEP and EXPRESS tools. In other words, we want to counter any reasons that industry might have against using STEP.

Access is provided in several forms:

- Programs may be invoked by commands submitted through e-mail.
- Programs may be invoked remotely and displayed on a local X server.
- Data may be uploaded to the Server for remote processing.
- Data may be accessed and/or downloaded from the Server for local processing.
- Programs may be downloaded from the Server for local use.

Each of these options will be discussed in more detail. However, first a word about the distinction between programs and data. In our domain, the distinction is hazy. For example, EXPRESS schemas contain executable portions but also serve as data to other programs, such as translators which either convert them to programs or yet a different form of the data.

## 2.1: Programs may be invoked by commands submitted through e-mail

A number of programs require no or little interaction. For example, the fedex program [4] is a tool which analyzes EXPRESS schemas for syntax and semantic problems. Users can e-mail their data and the Server will e-mail back the analysis.

While e-mail limits some of the interactions that can occur with programs run remotely, it has the advantages that it averages out the load on the Server and it can be used by people who are not directly on the Internet (or are far enough away that interactive use doesn't work well).

## 2.2: Programs may be invoked remotely and displayed on a local X server

The Server allows programs to run remotely using X windows. In particular, the user logs in to the Server and identifies a local host on which to display windows. The Server then directs the applications to display on the local host.

The Server restricts people after a certain load average has been reached in order to preserve good performance for users who have already logged in.

## 2.3: Data may be uploaded to the Server for remote processing

In order to work on their own data, users upload their files to the Server. This is currently done by e-mailing the file to the Server. If the user is sending a single file for, say, analysis, the command and data can be embedded in the same mail message.

A simple scheme exists that enables users to keep files grouped together by task. This is especially helpful if users are repeatedly correcting mistakes in just one of several files. A security mechanism prevents people from accessing each others files except with permission.

## 2.4: Data may be accessed and/or downloaded from the Server for local processing

Data can be accessed from the Server and in most cases downloaded as well to the local system. The user can always download data that they uploaded, but there are two more interesting sources: application output and standard reference material.

**Application Output.** Some of the applications produce new data or programs. For example, the shtolo program [5] converts a STEP short form to an annotated listing. This listing is normally the end product and for this reason is sent back automatically. The Data Probe generator [6] converts a set of EXPRESS schemas into a schema-dependent data browser complete with an X GUI (graphic user interface). A Data Probe may in turn be run at the server, so it is not sent back unless specifically requested.

**Standard Reference Material.** Traditionally, standards are expressed in a form intended primarily for consumption by people. However, more and more standards include parts that can be directly processed by software. For instance, the STEP standards include EXPRESS schemas. Given the standards in machine-readable form, the schemas can be automatically extracted and used by other software. Unfortunately, standards (including STEP) are not commonly distributed in machine-readable form.

A significant part of NIST work includes preparation of standard reference material. NIST standards are always placed in the public domain and are not copyrighted. However, in the STEP and EXPRESS projects, NIST is just one of many representatives participating by ISO rules. In this context, NIST abides by ISO, which copyrights all standards and draft standards.

NIST necessarily has access to many copyrighted standards and drafts but cannot distribute them publicly. However, NIST has permission to distribute selected portions of them. NIST can also distribute certain draft standards that are not yet copyrighted. These draft standards and portions of final standards are available through the Server.

In some cases, NIST cannot get permission even to release critical portions of standards or draft standards. This is particularly problematic since some draft standards are effectively not available through ISO by any means yet are of great interest to people not on the committees. In this case, the Server does not allow downloading but provides access indirectly only through the Server's local programs. For example, the Server can analyze user schemas that depend on unreleased and non-public draft schemas. The Server will send back the results but will not accept requests to download the non-public shemas.

### 2.5: Programs may be downloaded from the Server for local use

The Server contains a number of libraries and compilers that may be used to construct new tools. For instance, several of the tools are translators, typically translating EXPRESS to some other language such as C++. The new forms created by the Server can be downloaded by (or automatically e-mailed to) users.

## 3: Benefits

The Server provides several notable benefits over the pre-Server days when we would release a tool simply by placing it in our public ftp repository.

- Users need not dedicate the time or expertise to install and maintain tools. Some of the tools draw together a wide collection of other experimental toolkits (other compilers, window systems, databases, etc.) which are difficult to install and make work together.
- Users do not have to upgrade when a bug is discovered. Instead, they report the bug to us, we fix it, and all Server users automatically start using the new fixed version. Some of the tools are fairly exotic and we fix bugs in them every few days. Sending updates out at this frequency would be overwhelming to users.
- When a new draft of a standard is released, we immediately update the Server applications or data.
- Users have a neutral site at which to collaborate. No formal registration or fees are required. The *raison d'etre* for the STEP standard is data interchange between vendors, so having a neutral low-cost testbed fulfills several needs.
- Server usage patterns can illuminate critical needs within the community. We will be able to tell which standards are frequently referenced and which are not. This will help direct our attention when working on our other standards.

## 4: Problems

The Server has encountered a few conceptual problems. One notable problem is that Server sends out some programs that are necessarily platform specific. This is a difficult issue for NIST since NIST does not want to be seen as endorsing any particular vendors. Yet NIST cannot possibly maintain object files and binaries for a large number of platforms. Providing a server at NIST to run programs is a compromise but obviously incomplete. Short of a universal distribution form (e.g., OSF's ANDF), it is unlikely that this problem will be solved in a general way.

One problem that has not appeared but is worrisome nonetheless is that of abuse. The Server provides only enough security to prevent users from reading and writing other user's data. However it is possible to deny services to others by, for example, sending sufficiently large files to fill the entire disk. A more serious and likely problem is that of pirates using the Server as a drop site to transfer illegal software to others. It is not clear how to solve this without dramatically reducing the convenience to our real users.

## 5: Extensions and Other Uses

The Server is part of a larger project called the Application Protocol Development Environment (APDE) [7]. The APDE is a software workbench for the purpose of constructing Application Protocols (APs). (The definition of an AP is not relevant to this paper and will not be discussed.) The Server provides many of the analyses necessary to the AP development process.

While using the APDE, the user points and clicks through a GUI to achieve many actions. Many of the transactions with the Server are similarly initiated through the APDE. The APDE understands when a connection to the Server is required and allows control to move there as appropriate. Server results can also be automatically entered into the local database used by the APDE so that the user can continue the AP development process entirely locally.

The result is that the APDE uses the Server as a backend processor transparently to the user. As use of the APDE itself spreads, new APDE sites can also make use of the Server at NIST. At the same time, we also plan to incorporate most of the APDE itself into the Server so that little is required to use the APDE at a different site.

Another proposed extension to the Server is to incorporate the Multipurpose Internet Mail Extensions (MIME) [8]. This would simplify some of the Server's user interface. For example, users must currently understand how to uncompress and uudecode (and unsplit if output is sufficiently large) output files returned from the Server. However MIME requires specialized user interfaces which are not in common use among the Server's users at this time. When MIME-aware interfaces are more common, we will add support for it to the server.

## 6: Implementation

The Server is written in about 1000 lines of Expectk, a Tcl-based program for performing interaction automation and building GUIs [9]. The code to handle e-mail requests is approximately 900 lines while the code executed at login is 100 lines. Application specific processing (e.g., EXPRESS parsing) is performed by toolkit software written in C and C++ with yacc and lex.

Expect, Tcl, and Tk have worked out quite well. The total size of the project seems quite small considering the functionality and robustness of the system. More objective notes on the implementation and a full description of the user interface are provided elsewhere [10].

## 7: Traffic and Throughput Statistics

The Server runs on a Sun SPARCstation II owned and supported by the Factory Automation Systems Division at NIST. To preserve good response time, users are prevented from logging in when three or more users are effectively using 100% of the CPU time for more than one minute. This figure was not scientifically determined and may be readjusted as we gain experience with the Server.

One of the obvious concerns of any public Server is that demands upon it can grow so great that the underlying hardware cannot keep up. This is certainly a concern with the Server but may be ameliorated for several reasons:

- As users become more familiar with the software and more dependent on it, more sophisticated users request copies for themselves. This lowers the load (except when they need access to our standard reference material).
- The Server's primary users are AP development teams. Currently there are only 25 such teams, and it is expected that the number will not grow rapidly even though there is no finite maximum on the number of future APs. We believe that our hardware resources will continue to keep pace with the demand for service.

The Server has only been in operation for a year, but use has been well within the bounds for good response time. If anything, our initially conservative estimates have left us with a Server that is overkill for the job. It is

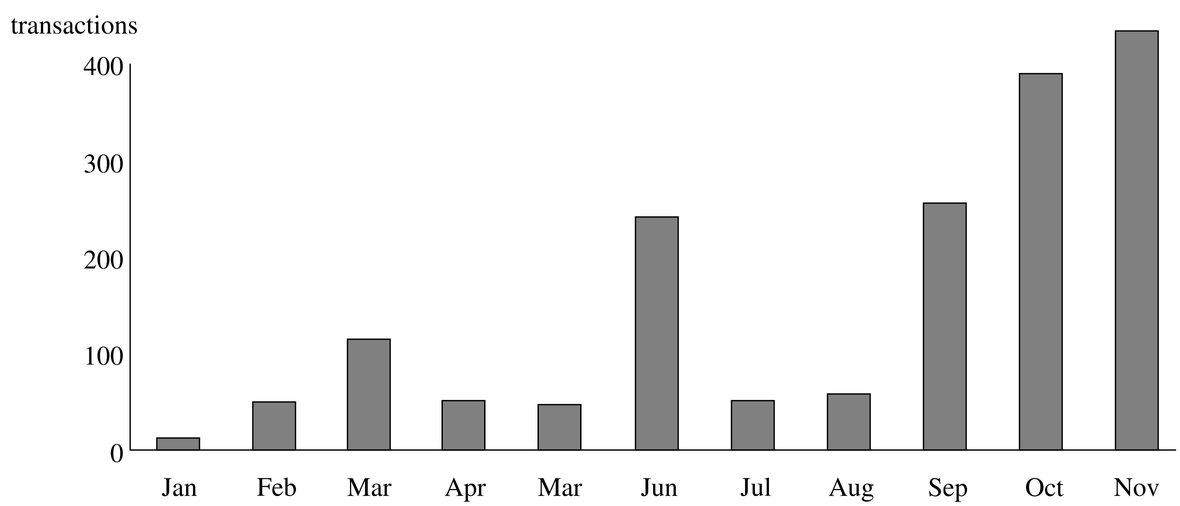


Figure 1: Number of Server Transactions during 1993
(Server began operation December 1992)

difficult to characterize the demands of a typical transaction. They range from less than a second to five minutes for the most complex request.

The Server runs on a dedicated Sun IPX. It has about a gigabyte of space dedicated to incoming user requests. After one month of total inactivity (meaning a user has not logged in or sent e-mail), an account is deleted (The account owner is informed of this.) As this paper is being written, seven accounts are consuming 7Mb total. The number of transactions during 1993 is shown in the Figure figure 1 indicating non-exponential, but healthy increases over the last few months as the Server has become better known in the AP community. The unusually high usage in June was due to the culmination of the first official AP which had developers working feverishly as the deadline neared.

The Server records who uses it but does not retain any user-provided data files once a transaction is completed. Logging information is intended only to show usage and justify the project to the funding sponsor. Users will never be retroactively billed.

## 8: Experiences and Conclusions

Providing a computational server free to the public can be a delicate proposition. We have tried to straddle a fine line, by providing only those services for which we have expertise (and software) and that would most help our users – at the same time without letting our resources be overused to the point that they would help no one. We are also taking advantage of being in the position of having access to machine-readable but copyrighted standards and draft standards. By allowing indirect access to these standards, we provide critical access to users and yet protect the property of the copyright holders. Also, we provide a neutral collaboration site to encourage companies to work together at essentially no cost to them.

Many users have taken us up on these benefits and have leveraged our research and support efforts to obtain constant access to state-of-the-arts tools and standards. More researchers and developers are turning to it each month and we are getting more requests for additional applications to support. According to users, the Server has been very helpful to AP developers and other users of EXPRESS.

## 9: Acknowledgments

The EXPRESS Server was funded by the NIST Scientific and Technical Research Services and is part of the Persistent Object Base Evaluation project. Some of the programs controlled by the Server were designed and built as part of the Application Protocol Development Environment (APDE) project which is funded jointly by the Continuous Acquisition and Life-Cycle Support (CALS) program of the Office of the Defense CALS Executive (ODCE) and ARPA. Development of shtolo was funded by CALS as part of the APDE project.

The author gratefully acknowledges Scott Paisley for significant ideas that become components in the Server, and Kent Reed, Barbara Goldstein, Allison Barnard, and Douglas Martin who were very patient and understanding while trying to use early versions of the Server.

## 10: Disclaimers

Trade names and company products are mentioned in the text in order to adequately specify experimental procedures and equipment used. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the products are necessarily the best available for the purpose.

## 11: References

[1] Mason, H., ed., "Industrial Automation Systems – Product Data Representation and Exchange – Part 1: Overview and Fundamental Principles", Version 9, ISO TC184/SC4/WG PMAG Document N50, December 1991.

[2] Libes, Don, "The NIST EXPRESS Toolkit – Introduction and Overview", National Institute of Standards and Technology, Gaithersburg, MD, to appear in *The CALS Journal*, Spring 1994.

[3] Spiby, P., ed., "ISO 10303 Industrial Automation Systems – Product Data Representation and Exchange – Part 11: Description Methods: The EXPRESS Language Reference Manual", ISO DIS 10303-11:1992(E), July 15, 1992.

[4] Libes, Don, "The NIST EXPRESS Toolkit – Using Applications", NISTIR 5206, National Institute of Standards and Technology, Gaithersburg, MD, June 9, 1993.

[5] Libes, Don, "Shtolo – Converting STEP Short Listings to Annotated Listings", NISTIR 5291, Gaithersburg, MD, November 8, 1993.

[6] Sauder, Don, "Data Probe User's Guide", NISTIR 5141, National Institute of Standards and Technology, Gaithersburg, MD, March 1993.

[7] Clark, Stephen N., Barnard Feeney, Allison, and Fowler, Jim E., "Specifications for an Application Protocol Development Enviroment", NISTIR 5248, National Institute of Standards, Gaithersburg, MD, August 1993.

[8] Borenstein, N., and Freed, N., "MIME (Multipurpose Internet Mail Extensions): Mechanisms for Specifying and Describing the Format of Internet Message Bodies", RFC 1341, Bellcore. Innosoft, June 1992.

[9] Libes, Don, "Expect: Scripts for Controlling Interactive Programs", *Computing Systems*, pp. 99-126, Vol. 4, No. 2, University of California Press Journals, CA, Spring 1991.

[10] Libes, Don, "The NIST EXPRESS Server – Usage & Implementation", Internal Report, National Institute of Standards, Gaithersburg, MD, December 1, 1993.