

SAS/GRAPH Over the Web A Reliable Method Using PDF

By Adam Hendricks, TAD Technical Services

Sample Specifications (Server Hardware/Software):

- UNIX web server with large work area (/var/tmp) having 20Mb or more for each expected concurrent client using SAS (In addition to normal system requirements).
- DBMS (optional)
- SAS/BASE, SAS/GRAPH, SAS/ACCESS (optional), and SAS/OR (optional)
- SAS/IntrNet (Required for Open Internet Use) or server-based SAS license for unlimited users (Intranet Only).
- PERL5 or C for CGI (Both are free on UNIX)
- Adobe Acrobat 3 Distiller to convert PostScript files to PDF format.

Sample Specifications (Client Hardware/Software)

- Build the Application with the following clients in mind:
 - Client machines that are powerful enough to use the web without too much waiting around. (e.g. Pentium PC w/ 32Mb+ RAM, newer Mac or X-Terminal).
 - JavaScript Enabled Browsers using Adobe Acrobat 3 Reader (Download free) as helper application for PDF files (Netscape 3.0 or better recommended). Use of MS Word for PDF viewing on MSIE is recommended.

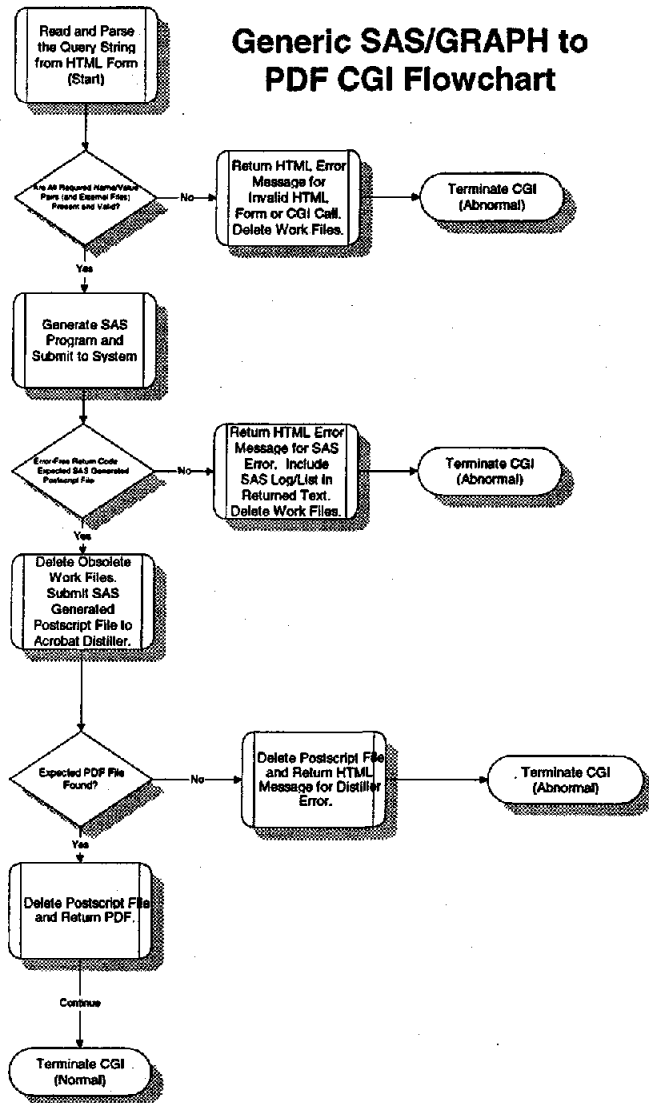
CGI Language Specifications

- Must be able to read and create server environment variables
- Must be able to read from STDIN and write to STDOUT.
- Must be executable on the web server.

CGI Recommendations

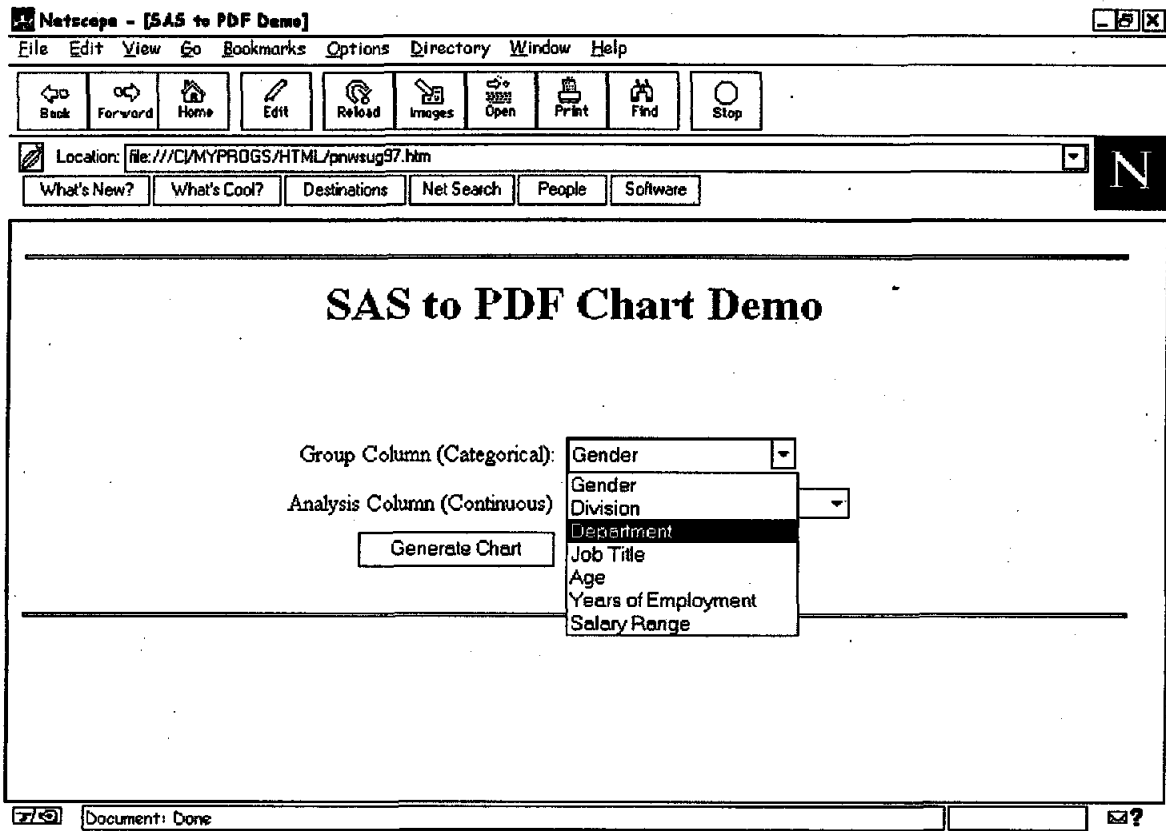
- If your programming experience is primarily in SAS or some other 4GL, then use Perl. It will be less painful. You can use the publicly available cgi-lib.pl by Steven E. Brenner or the cgi.pm utilities that come with PERL5.
- If you are comfortable with C, you can start out with a good publicly available library like CGIHTML by Eugene Eric Kim. It will save you much time. Modify any Read/Parse functions; so that CGIs can be run from the command line (debug). This will you save time and frustration during development.

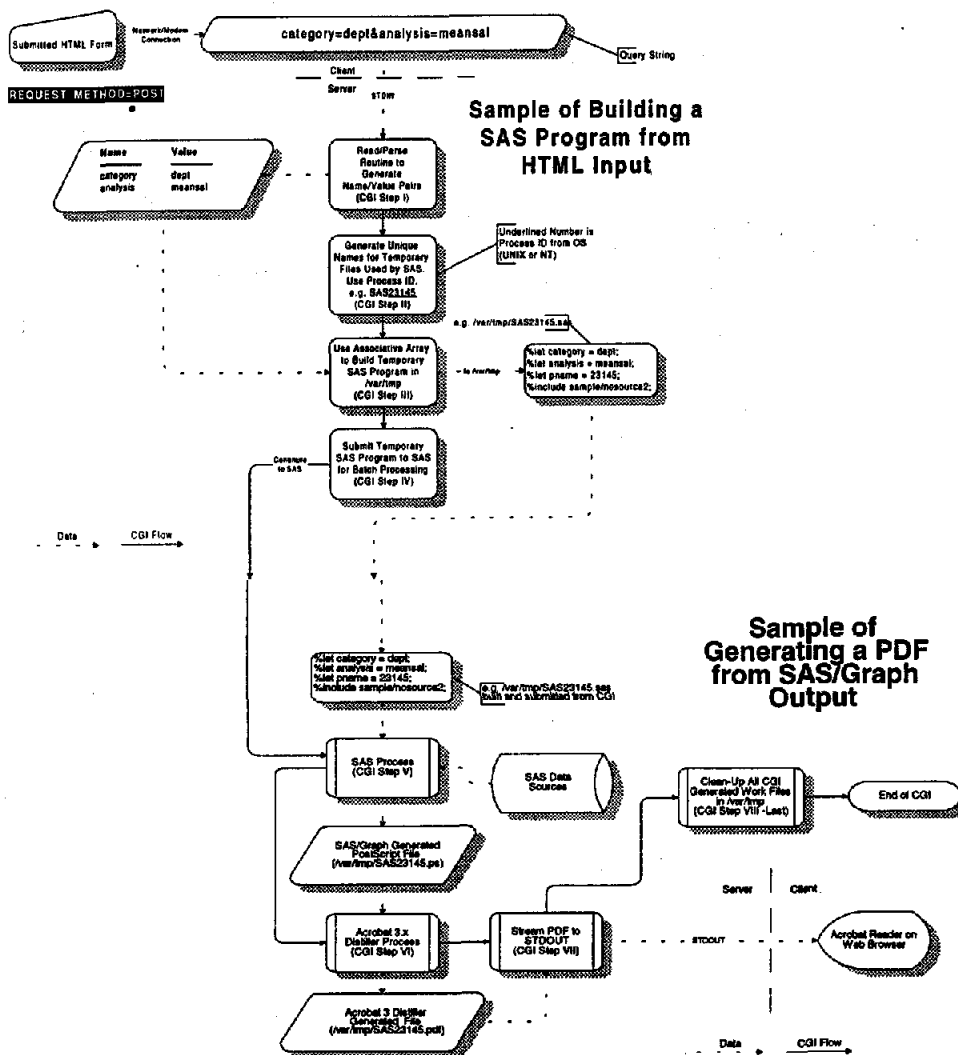
Generic SAS/GRAPH to PDF CGI Flowchart

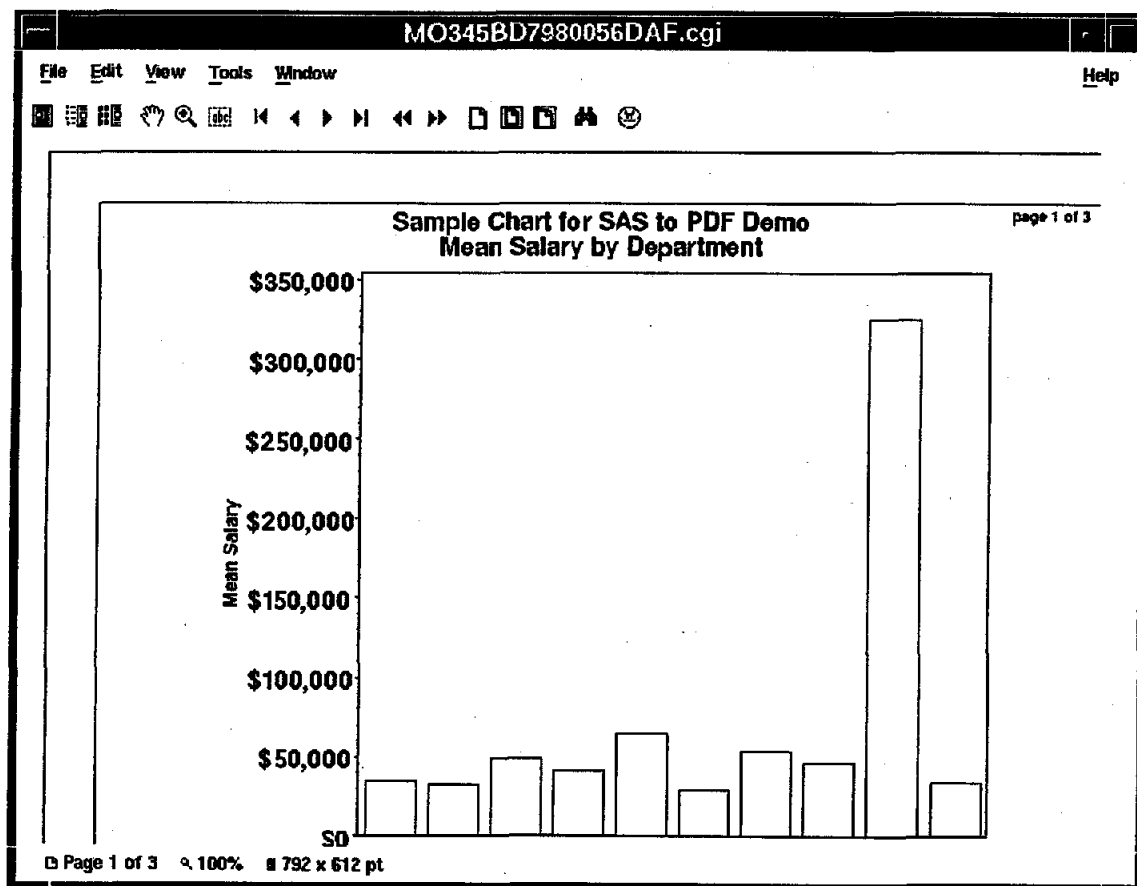


Steps for Development

1. Develop a functioning SAS program that only requires a series of discrete macrovariable definitions to generate a chart conforming to user requirements. The SAS program must be able to run on the same machine that hosts the web server. All output from the SAS program must be written to appropriate work area for your server. This includes the -saswork area defined in config.sas. Set DEVICE= in GOPTIONS to a Postscript device (PS, PSL, etc...). Make sure NODISPLAY option is in effect. Make sure all required data sources are accessible to the web server user ID (usually "http"; hopefully not "root").
2. Build a CGI (Common Gateway Interface) designed to build a small temporary SAS program containing all the macrovariables required to run the SAS program and an %INCLUDE of the SAS program (which is static). Make sure that the CGI handles all possible outcomes. Avoid sending generic error message back to client browser.
3. Create HTML form that capture all the necessary parameters to run the SAS program. Set the form to submit to the appropriate CGI. Use JavaScript for client side checking of input when appropriate.
4. Make sure that Acrobat Distiller is installed on the server and that Acrobat Reader is available to the browser used by the developer. The configuration file, AcrobatDistiller, needs to be in the root (/) directory or Acrobat Distiller will probably not be usable from any CGI call on the web server.
5. Copy the HTML to a directory configured to serve as an HTML source for the Web server (usually called /htdocs/ or <http address>/~<userid>/).
6. Copy CGI to a directory configured to serve as a CGI source for the Web server (usually called /cgi-bin/ or <http address>/cgi-bin- <userid>).
7. Test all possible input types. Test CGI on command line first if possible; then test on web.







Recommended Books for Web Programming

- CGI Developers Guide by Eugene Eric Kim. Include CD-ROM with CGIHTML Clibrary with examples in both C and PERL.
- E-Zone Books by the Waite Group for HTML, JavaScript and PERL5. All have web-based study sites with chapter exams and certificates of completion.