

## Paper GH-02

**A General Purpose SAS® Report Portal for the Web**

Craig Ray, Westat, Rockville, MD

**ABSTRACT**

In order to meet the demands of today's information consumers, SAS report programmers frequently need to make their content available interactively to end users. Web technology provides an excellent method for users to request on-demand reports and supply any necessary parameters, but setting up the necessary web infrastructure can be a complex task, requiring a skill set very different from that needed for SAS report programming. Recognizing these needs, we developed and implemented the Westat Automated Report Portal (WARP), which utilizes SAS/IntrNet® software to provide a generalized and secure environment for SAS programmers to make their reports web-accessible without requiring any web programming. Since its launch in late 2003, WARP has provided a cost-effective mechanism for dozens of Westat projects with interactive reporting needs to leverage the power of SAS to develop sophisticated, parameterized reports. This paper describes the overall architecture of the system and summarizes its capabilities in areas such as output format (HTML, Excel, PDF, RTF), administration, customization, and project integration.

**INTRODUCTION**

The SAS System provides programmers with a rich environment for developing reports. At the same time, the web environment enables end users to request report output from the public Internet without installing additional software. The essential dilemma is that the skills to develop SAS programs are very different from the skills required to make the reports accessible on the web. Requiring that projects need to have both SAS and web programmers to make SAS programs available on the web requires additional programming resources, thus increasing the cost to the projects.

The Westat Automated Report Portal (WARP) provides a convenient and generalized system for programmers to make SAS reports accessible to end users on the public Internet in a secure environment. SAS programmers can concentrate on report content, not web site configuration, development, and support. WARP contains a number of advanced features, including the ability for users to view output in several formats, a SAS-for-the-web testing environment, project administration, project customization options, and extensive built-in web security. The WARP web site was developed with Active Server Pages and SQL Server and accesses SAS via SAS/IntrNet.

Users of WARP are presented with an easy-to-use interface to select a report and supply values for run-time parameters. They can also select a preferred output format: HTML, Excel, PDF, or RTF. The Excel option provides a convenient mechanism to enable users to download data via the Internet.

**WARP FEATURES**

WARP contains many reusable features that are commonly required or desirable in web-based business intelligence systems. These features greatly reduce the cost of programming and deploying SAS report programs.

- WARP is entirely web-based. The results are rendered in a standard browser and no special client installation is required. In addition, the administrative system is entirely web-based.
- User and administrator authentication is built into the system. Users must provide a user name and password to access the system. In addition, user requests and results are fully encrypted with Secure Sockets Layer (SSL).

- The system provides full authorization capabilities. Access to specific reports and features is role-based; users are designated to view reports and/or perform administrative functions. Users are restricted to only run reports for which they have been explicitly granted access.
- SAS report programs may be configured with parameters. Parameters are passed to the underlying SAS program as macro variables via SAS/IntrNet. Different types of parameters may be configured (numeric, character, date, or drop-down-list) and the web site checks for the validity of the user entry. In addition, drop-down-list parameters may have fixed choices or be dynamically generated choices from an underlying database lookup table.
- The run-time environment automatically prompts users for the desired output format. WARP supports HTML, PDF, Excel, and RTF. The system generates all the necessary SAS ODS statements.
- SAS programs configured within WARP can access data in any format that SAS can read, including SAS datasets, relational data, Excel, Microsoft Access, and ASCII data. The data does not need to migrate to the WARP web environment. Rather, it can remain in project-secure areas. The data only needs to be in a security zone accessible from the SAS application server.
- WARP contains completely separate and parallel test and production environments. Test results are viewed in the browser environment so the programmer can be sure that the program runs correctly in the web. Developers are able to view the SAS log in the test environment, which simplifies debugging. After migrating programs to production, developers still retain the option to debug (that is, view the SAS log), although non-programmers are prevented from viewing the SAS log in order to maintain system security.
- WARP contains a full web-based administrative system. SAS developers can configure their own reports and parameters. System administrators are able to configure users and grant rights to access reports.
- WARP contains several features to permit seamless integration of its web-based SAS reporting into larger systems web systems. Project logos are easily incorporated into the user interface. External web sites can pass WARP credentials to bypass the default WARP login page (single-signon), and WARP reporting can be accessed as a web service, bypassing the WARP user interface entirely.

## WARP ARCHITECTURE

WARP is a single metadata-based system, within which any number of projects can be configured. The web site does not need to be replicated for each new project. The overall architecture of the WARP system is depicted in Figure 1.

The WARP approach contains a full test environment located within the corporate firewall, and a production environment, which end users access for their reports over the public Internet.

The foundation of the system is the metadata database, in which projects, reports, report parameters, and user and user group permissions are established. In addition, all user activity is tracked in the database.

The web site performs user authentication, and based on the user's privileges, presents a menu of project and report selections. Users select a desired report and are then presented with a list of the report's specific parameters. The web site performs basic edits on the parameter values (for instance, ensuring a date parameter value is a valid date), before invoking the SAS/IntrNet broker. The broker makes requests of the SAS application server, which is where the SAS programs reside.

A SAS program can access any data within its zone of security, using SAS/ACCESS® software. This means that the data for the SAS report programs do not need to migrate into the WARP area. This maximizes data security by eliminating the need to create separate, redundant databases solely for reporting purposes.

SAS programs and the corresponding metadata and user permissions are developed and tested in the testing environment. After development and testing have been completed, the programs and metadata are migrated to the production environment using WARP's built-in administrative utilities.

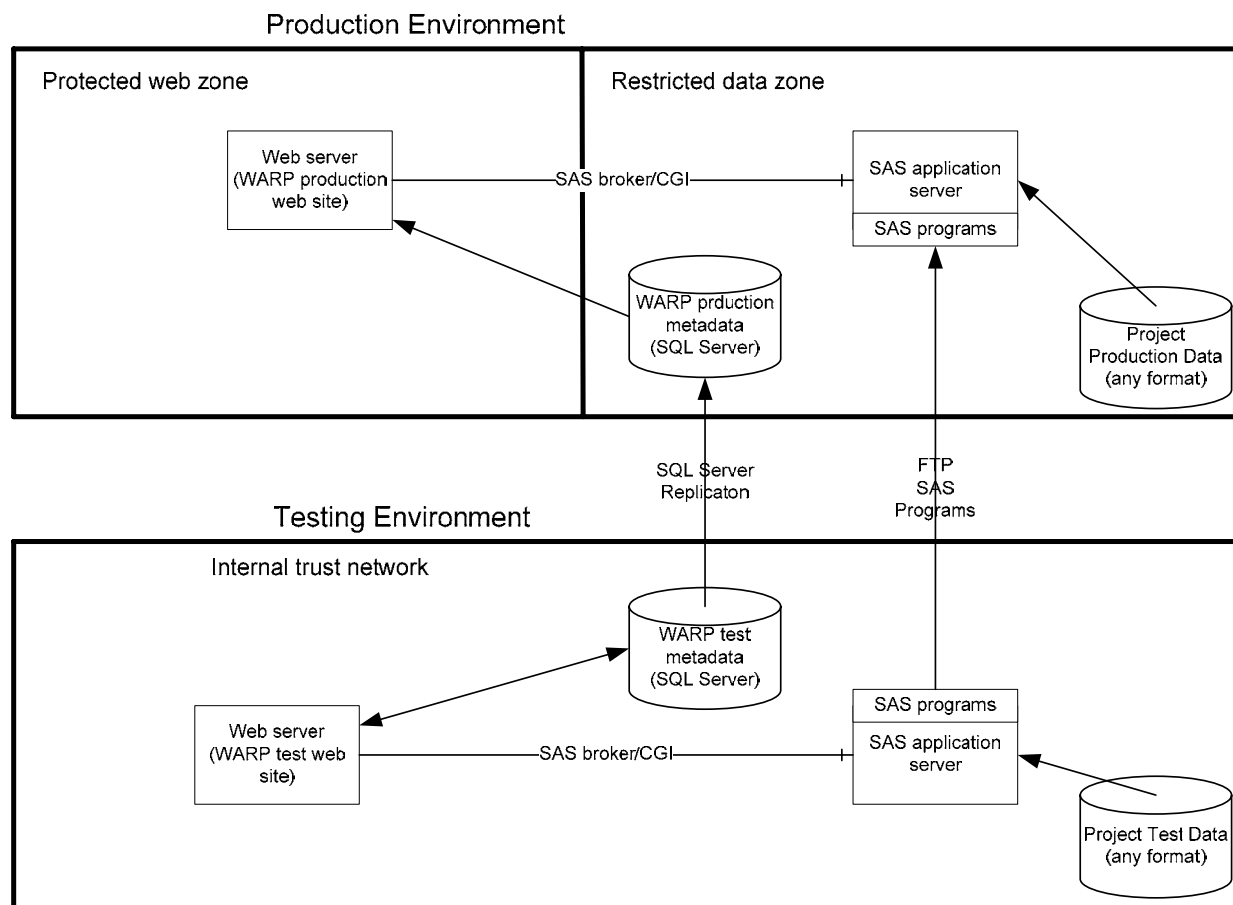


Figure 1

## USER RUNTIME ENVIRONMENT

After users log into the system, they are presented with a list of authorized projects, followed by a list of authorized reports for the selected project (Figure 2).



Figure 2

After selecting the report, the user is presented with the list of parameters for which to supply values. Figure 3 illustrates all various parameter types that can be configured in the metadata:

- numeric,
- character,
- date (rendered with a calendar date-picker),
- enumerated (hard-coded pick-list),
- dynamic (enumerated list dynamically generated from a project database),
- Select-many enumerated list,
- Secure parameter (masked input, which can be used, for example, for passwords).

The user also can select the output format from the drop-down list seen in Figure 3. Choices are:

- HTML,
- PDF,
- RTF,
- Excel.

In the testing environment (and in production for administrators), programmers can choose to 'debug,' which will cause the SAS log to display in the browser, along with the SAS program's output.

☐ Debug Parameter Value (for Super Admin only!)

**Demo SAS 2 - Many Parameters** Remember Parameter Values: ☒

This report is designed to demonstrate many of the features of WARP. The user can see the output rendered in a variety of formats, see the various parameter types, and view the effects of many SAS styles.

Numeric Parm: ?	10 * 123...
Character Parm: ?	abcdefghi *
Date Parm: ?	mm/dd/yyyy
Enum Parm: ?	Male (M) *
Dynamic Parameter: ?	Dynamic *
Dynamic Stored Procedure: ?	<div> 1: Dynamic  2: Static </div>
Secure Parm: ?	<input type="password"/> *
Report Style: ?	Analysis *

**Report Type:**  
SAS

**Debug Level:**  
NO DEBUG

**Output Format:**  
HTML

Figure 3

## PROGRAMMING AND TESTING IN WARP

The WARP system is designed to simplify SAS programming for the web environment. Programmers can develop and test their programs directly in WARP, so they can see how the output will be displayed in a browser. This avoids the inefficient process of developing programs in external environments, such as in the SAS interactive environment, only to have to port the programs and have to make modifications. To develop programs in the WARP environment, programmers map a directory to the SAS server within the Internal Trust Network (Figure 1). They can edit their programs in any standard text editor, saving their programs directly on the SAS server. Then they can switch to their browser and run the programs directly from WARP, as seen in Figure 3. By selecting the Debug Level to view the SAS log, they can debug directly from the SAS server.

The output format for programs configured within WARP is intended to provide flexibility. As seen in Figure 3, the user can select the Output Format. The WARP environment contains SAS macros to generate the necessary SAS ODS code, so SAS programmers generally do not need to supply program-specific ODS statements.

The WARP web runtime environment creates a series of parameters that are passed to the SAS programs as automatic macro variables. These can be very useful for SAS programmers to enhance their programs or restrict access to data, as necessary. These include the following:

- Format: the user-selected output format, either HTML, PDF, RTF, or Excel. The programmer can use this macro variable to enhance the output, as needed,
- Environment: Either TEST or PROD, which informs the SAS program whether the program was invoked from the testing or production environment.
- Project: Informs the SAS program of the project selected by the user. SAS programs can then be reused across different projects, with appropriate run-time changes to the logic as needed.
- LoginID: The WARP login ID used to access the system. This macro variable can be used to restrict data access, for instance.

## ADMINISTRATION

WARP is a metadata-based system, with metadata about the SAS report programs and users who have access to them stored in a SQL Server database. A key factor in the overall success of the system is having a fully functional administration module within the WARP web system. SAS programmers who are granted the role of 'project administrators' can configure their own metadata, without requiring support from the corporate WARP system administrator.

Project administrators can define the report's metadata and define what user groups should be authorized to run the report. Figure 4 illustrates the web screen to define the characteristics for a report, including:

- Report Label: The label the end user sees to select the desired report,
- Report Name: The name of the specific SAS program that the web site should run,
- Help Text: An extended description to be displayed to the end user,
- Applied Projects: The projects to which the report is assigned,
- Applied Groups: Groups of users who can access the report program.

The screenshot displays the WESTAT Automated Reports Portal. At the top is a navigation bar with links: HOME, PROJECT, MAINTENANCE, HELP, and LOG OFF. Below this, the current project is set to 'DemoProject\_SAS'. A 'View Associated parameters' link is visible on the right.

The main configuration area includes several fields:

- Report Label:** Demo SAS 2 - Many Parameters \*
- Report Desc:** Demo report with all style of parameter \*
- Report Name:** DemoRpt2 \*
- Report Relative Path:** (empty field)
- Report Type:** Radio buttons for SAS (selected), Crystal, and Static.
- Report Status:** Radio buttons for Active (selected) and Inactive.
- Unicode:** A checkbox that is currently unchecked.
- Report Help Text:** A text area containing the message: "This report is designed to demonstrate many of the features of WARP. The user can see the output rendered in a variety of formats, see the various parameter types, and view the effects of many SAS styles."

Below the configuration fields, there are two selection sections:

- Please select the applied projects:** A list box on the left contains 'AARP', 'ACTG', 'ACTG\_IMPACT', 'ACTG\_REGULAR', 'ADMA\_001', and 'ANGES0205'. On the right, 'DemoProject\_SAS' is listed.
- Please select the applied groups:** A list box on the left contains 'AARP\_ALL', 'AARP\_DEV', 'AARP\_MGMT', 'ACTG\_ALL', 'ACTG\_Dev', and 'ACTG\_IMPACT\_ALL'. On the right, 'DemoAdminGroup' and 'DemoUserGroup' are listed.

At the bottom of the form are four buttons: 'Spawn New', 'Update Record', 'Delete Record', and 'Reset'.

Figure 4

Each report can have any number of associated parameters, which the project administrator can define within the WARP administrative module. Figure 5 displays the screen where project administrators define parameters, which include:

- Parameter Label: the label displayed to the user,
- Parameter Help Text: extended help text for the user,
- Parameter Name: the name of the macro variable in the SAS program into which the parameter value will be passed,
- Parameter Type: type of parameter. The parameter type defines how the parameter should be rendered in the browser and the screen edits to be made on the values,
- Parameter Values: the way to list possible values in a drop-down list.

Appropriately trained and authorized staff can be designated as System Administrators. System administrators have additional privileges to define users and user groups, and migrate programs and metadata to production. WARP also tracks the date and time each report was requested and the total time the program took to run. System administrators have access to a suite of system monitoring reports that display statistics on system usage. These reports have been developed in SAS and are configured within WARP itself, in a project that only the system administrators can access.



The screenshot shows the WESTAT Automated Reports Portal interface. At the top, there is a navigation bar with links: HOME, PROJECT, MAINTENANCE, HELP, and LOG OFF. Below this, the project is set to 'DemoProject\_SAS' and the report is 'Demo SAS 2 - Many Parameters'. The main form is for creating a parameter with the following fields:

Parameter Label:	Date Parm *		
Parameter Desc:	Sample Date Parameter		
Parameter Help Text:	Only a valid date is permitted for this parameter. Note: no asterisk meaning this is not a required parameter.		
Parameter Name:	DateParm *	Parameter Category:	Static
Parameter Type:	<input type="radio"/> Numeric <input type="radio"/> Text <input checked="" type="radio"/> Date <input type="radio"/> Enum <input type="radio"/> Secure		
Parameter Max Length:	10 *		
Data Range:	Min: <input type="text"/> Max: <input type="text"/>		
Default Value:	<input type="text"/>	Multiple Selection:	<input type="checkbox"/>
parameter sequence:	3	Required Parameter:	<input type="radio"/> Yes <input checked="" type="radio"/> No
# Of Parameter Values:	0	<input type="button" value="Enter Parameter Values"/>	

At the bottom of the form, there are four buttons: Spawn New, Update Record, Delete Record, and Reset.

Figure 5

## CONCLUSION

The Internet is a very important tool to deliver real-time results to both internal and external end users. However, developing secure web sites for SAS programs is a costly activity and requires very different skills than SAS programming. A general, reusable metadata-driven web site that automates the web functionality can improve overall programmer productivity, reduce cost, and help ensure web security by standardizing best practices. The Westat Automated Report Portal (WARP) is an example of such a web site and has provided Westat projects with an effective tool over the past several years.

## ACKNOWLEDGMENTS

The author would like to thank Mike Rhoads for his careful review and feedback and thank Karen Hewitt for her preparation of this manuscript.

## CONTACT INFORMATION

Craig Ray  
[Rayc1@westat.com](mailto:Rayc1@westat.com)  
 Westat  
 1600 Research Blvd.  
 Rockville, MD 20850  
 (301) 517-8077

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. □□ Other brand and product names are trademarks of their respective companies.