Paper 194-28

Creating Dynamic Web Based Reporting

Prepared by

destiny
CORPORATION

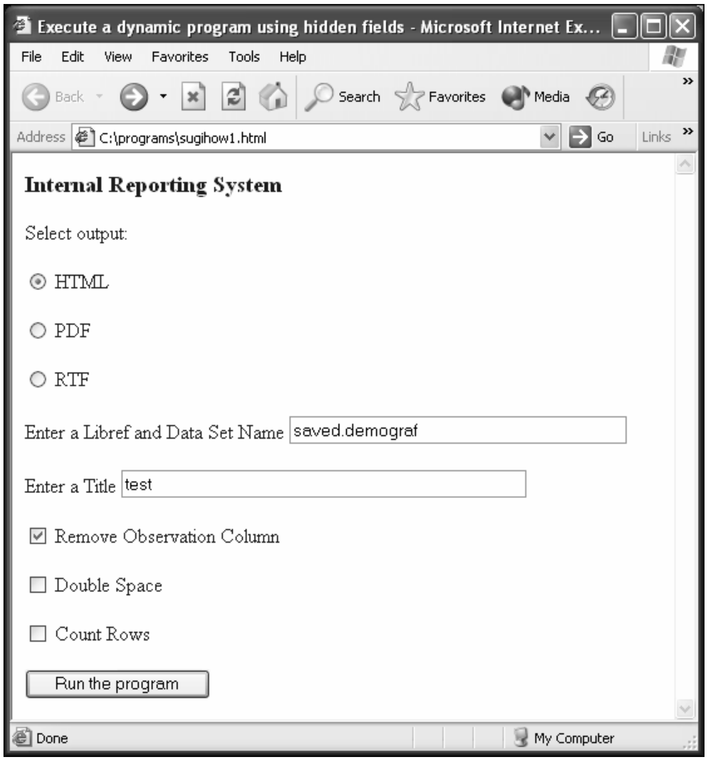
Global Business Intelligence Solutions

Destiny Corporation – 100 Great Meadow Rd Suite 601
Wethersfield, CT 06109-2379
Phone: (860) 721-1684 - 1-800-7TRAINING
Fax: (860) 721-9784
Email: info@destinycorp.com
Web: www.destinycorp.com

In this hands on workshop, we'll demonstrate and discuss how to take a standard or adhoc report and turn it into a web based report that is available on demand in your organization. In the workshop, attendees will modify an existing report and display the results in various web based formats, including HTML, PDF and RTF.

To do this, we'll use Dreamweaver software as a GUI tool to create HTML web pages. We'll use SAS/Intrnet software as a back end tool to execute SAS programs with parameters selected on the HTML screen presented to the user.

Our goal is to create the following screen for user input.


Execute a dynamic program using hidden fields - Microsoft Internet Ex...
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\programs\sugihow1.html Go Links
Done
My Computer

Internal Reporting System

Select output:

● HTML

○ PDF

○ RTF

Enter a Libref and Data Set Name saved.demograf

Enter a Title test

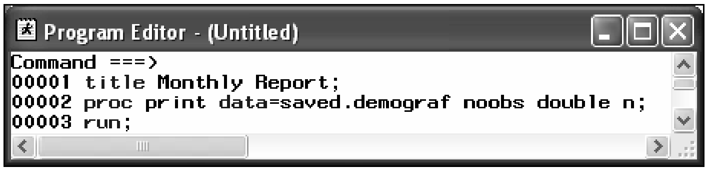
☑ Remove Observation Column

□ Double Space

□ Count Rows

Run the program

We will use a simple program that looks like the following and modify it.



```
title Monthly Report;
proc print data=saved.demograf noobs double n;
run;
```

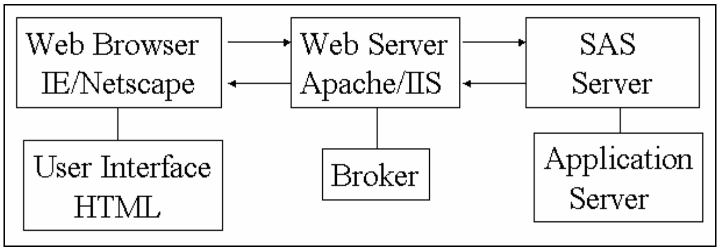
Overview of SAS/INTRNET Software

First, it is important to understand SAS/INTRNET software and its use.

Three components are required for the SAS/INTRNET software to work.

- Web Server Software – such as Microsoft's Personal Web Server/Internet Information Services, or the Apache Web Server.
- Web Browser – Such as Microsoft's Internet Explorer or Netscape's Navigator.
- SAS/INTRNET Software – Called the Application Dispatcher. It is composed of 2 pieces.
  - SAS Application Server – A SAS program on a Server licensed with the SAS/INTRNET Module.
  - Application Broker – A Common Gateway Interface (CGI) program that resides on the web server and communicates between the Browser and the Application Server.

These components can all reside on the same system, or on different systems.



Types of Services

- Socket Service: Is constantly running, waiting for incoming Transactions.
- Launch Service: A new service is started for each request.
- Pool Service: Provides a pool of servers that will be started when needed.

Installing and Running a Web Server

In order to execute the SAS/INTRNET software you need a "Web Server".

There are numerous Web Servers on the market today. Microsoft IIS/Personal Web Server or Apache are amongst the most common since they are free.

- SAS/INTRNET needs to be installed in your environment.
- You still need to configure the Software for use.
- You need to select the type of Service you wish to use and set it up for use.

Socket Service Description

Socket services consist of one or more application servers that run continuously servicing client requests.

Socket services start when a machine is restarted (either manually or by an operating system mechanism for starting processes at boot or login time).

The service usually runs until the machine is shut down.

Socket services are relatively simple to configure and manage.

Socket services are adequate for most development applications

Advantages

- Socket services are supported on all SAS/IntrNet platforms. Other service types are not supported everywhere.
- The server is already running by the time a client request appears. Clients do not have to wait for a server to start.
- The administrator has explicit control of resources allocated to the service. The administrator can control how many servers are run on each system and what resources are allocated to each server.
- Increasing load can be handled by adding more servers to the service.

Disadvantages

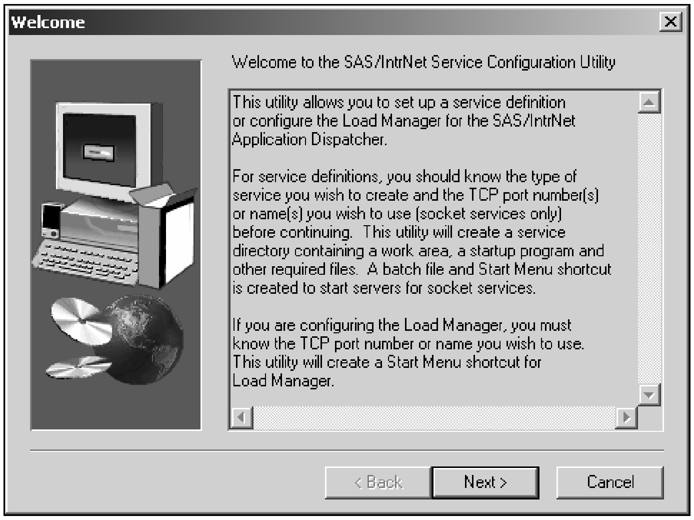
- Servers must be started and stopped manually or by the operating system. No automated start-up and shutdown is provided by SAS/IntrNet software.
- No dynamic scaling to meet increasing loads is provided. A fixed number of servers is available to handle all client requests. A few long-running requests can slow the entire service for all clients.

Setting up the Application Server

SAS provides the INETCFG utility to setup the Application Server.

To run it select the Windows Start menu, by selecting Start, then Programs, then The SAS System, then INTRNET, then Create a New INTRNET Service.

The Welcome Screen is displayed.


Welcome

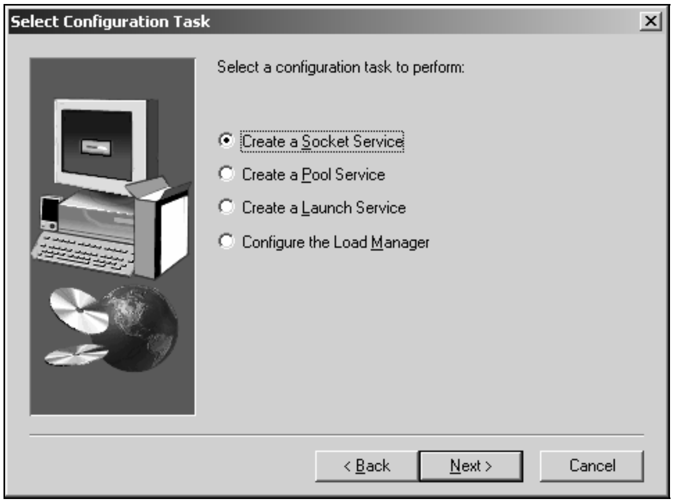
Welcome to the SAS/IntrNet Service Configuration Utility

This utility allows you to set up a service definition or configure the Load Manager for the SAS/IntrNet Application Dispatcher.

For service definitions, you should know the type of service you wish to create and the TCP port number(s) or name(s) you wish to use (socket services only) before continuing. This utility will create a service directory containing a work area, a startup program and other required files. A batch file and Start Menu shortcut is created to start servers for socket services.

If you are configuring the Load Manager, you must know the TCP port number or name you wish to use. This utility will create a Start Menu shortcut for Load Manager.

Select Create a Socket Service.

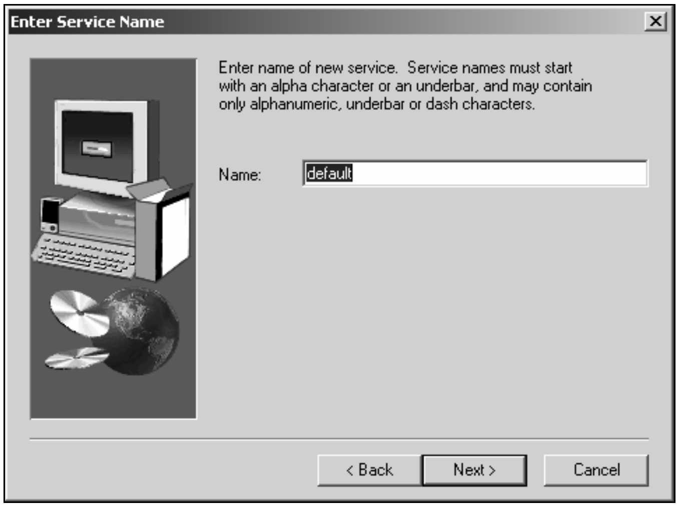

Select Configuration Task

Select a configuration task to perform:

● Create a Socket Service
○ Create a Pool Service
○ Create a Launch Service
○ Configure the Load Manager

Select a name for your service.

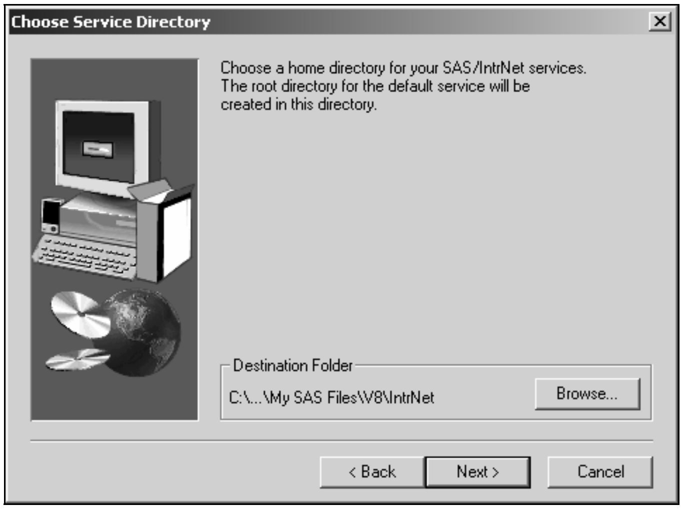
You can choose any name for your service. We are going to choose default.


Enter Service Name

Enter name of new service. Service names must start with an alpha character or an underbar, and may contain only alphanumeric, underbar or dash characters.

Name: default

Select a Destination Folder for your service. Usually you will use the Default.

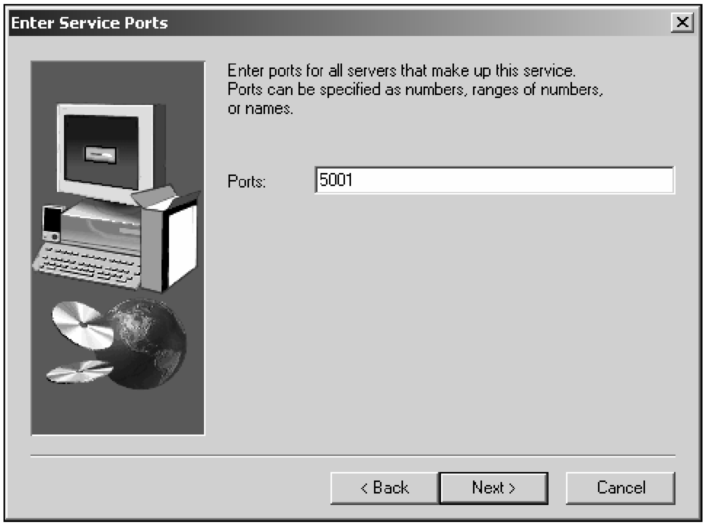

Choose Service Directory

Choose a home directory for your SAS/IntrNet services. The root directory for the default service will be created in this directory.

Destination Folder
C:\...\My SAS Files\V8\IntrNet

Select an available port as the PORT number for your service.

This will be the where your application Server “listens” for incoming requests from the broker.


Enter Service Ports

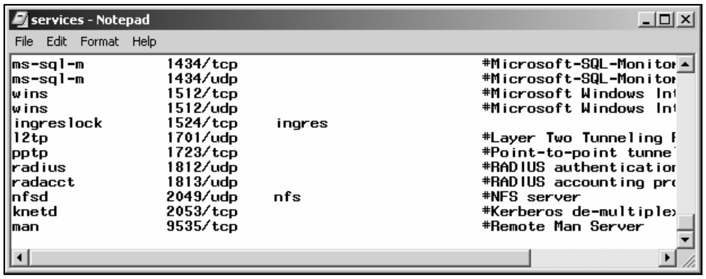
Enter ports for all servers that make up this service. Ports can be specified as numbers, ranges of numbers, or names.

Ports: 5001

If you are in doubt as to which port to select, you can consult the services file.

You do not have to make an entry in this file to make the port work for the Application Server.

This file is located in the c:\winnt\system32\drivers\etc directory under Windows 2000 and Windows NT.



```
ms-sql-m          1434/tcp                        #Microsoft-SQL-Monitor
ms-sql-m          1434/udp                        #Microsoft-SQL-Monitor
wins              1512/tcp                        #Microsoft Windows Int
wins              1512/udp                        #Microsoft Windows Int
ingreslock        1524/tcp    ingres
l2tp              1701/udp                        #Layer Two Tunneling P
pptp              1723/tcp                        #Point-to-point tunne
radius            1812/udp                        #RADIUS authenticatio
radacct           1813/udp                        #RADIUS accounting pro
nfsd              2049/udp    nfs                 #NFS server
knetd             2053/tcp                        #Kerberos de-multiplex
man               9535/tcp                        #Remote Man Server
```

It lists all the ports that are currently in use.

Typically ports 5001 and above are available for this purpose.

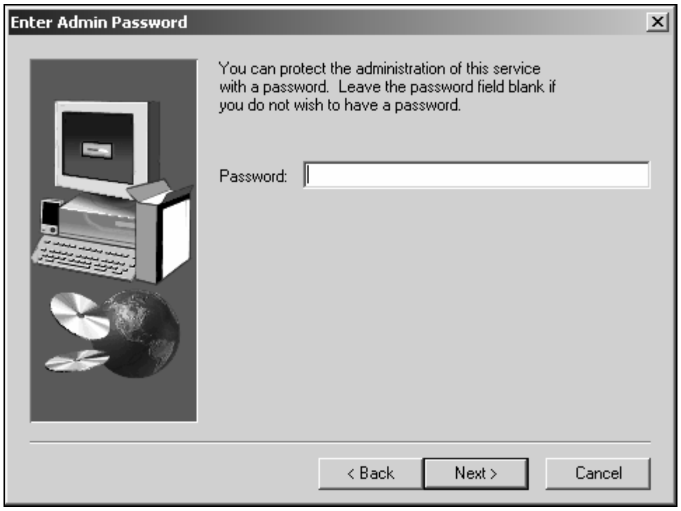
Select an Administrator password for your service.

SAS/INTRNET allows you to Administer your service through your browser.

You can restart or even shutdown your service.

Assigning a password forces you to use a password with these commands.

For local development, a password is not necessary.

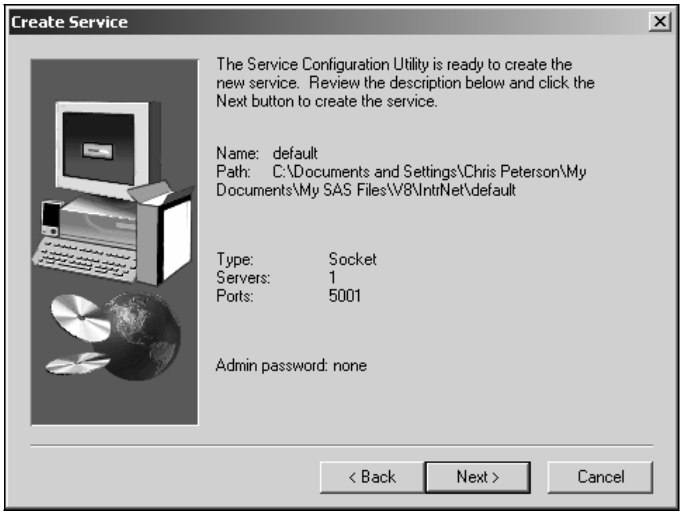

Enter Admin Password

You can protect the administration of this service with a password. Leave the password field blank if you do not wish to have a password.

Password:

You will be shown a summary of your selections.

Click Next.


Create Service

The Service Configuration Utility is ready to create the new service. Review the description below and click the Next button to create the service.

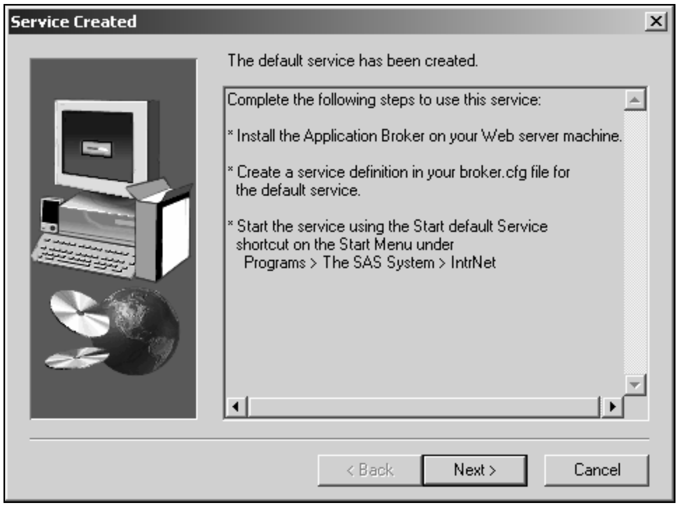
Name: default
Path: C:\Documents and Settings\Chris Peterson\My Documents\My SAS Files\V8\IntrNet\default

Type: Socket
Servers: 1
Ports: 5001

Admin password: none

You will be given some instructions.

In particular to edit the Broker.CFG file.


Service Created

The default service has been created.

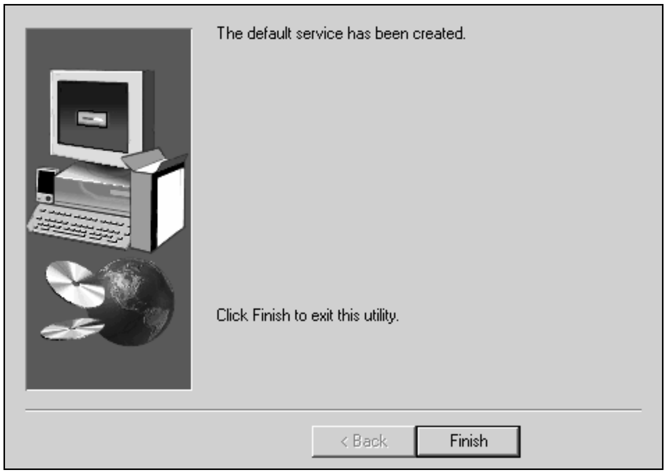
Complete the following steps to use this service:

* Install the Application Broker on your Web server machine.

* Create a service definition in your broker.cfg file for the default service.

* Start the service using the Start default Service shortcut on the Start Menu under Programs > The SAS System > IntrNet

Select Finish to complete the installation.



The default service has been created.

Click Finish to exit this utility.

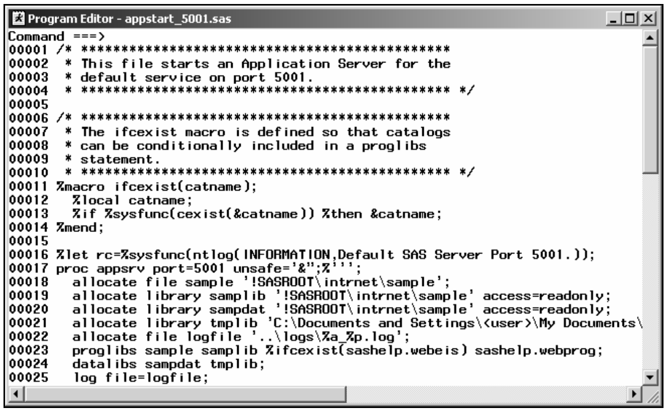
The system has setup the Default Service for us.

It has created the following SAS program.

This program runs proc appsrv and establishes data and program libraries for us to use.

This program will be located in the directory that you chose above.

Under Windows 2000 the default location is c:\Documents and Settings\<USER>\My Documents\My SAS Files\INTRNET\v8\<SERVICE NAME>\



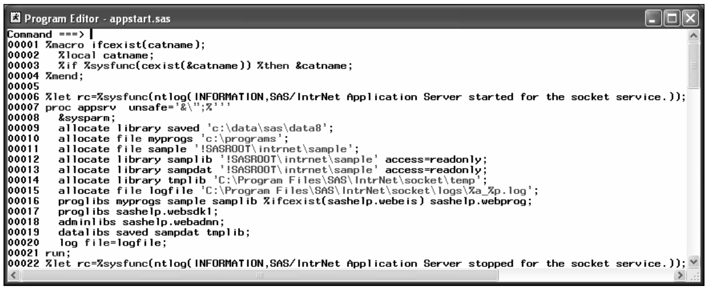
Program Editor - appstart_5001.sas

```
Command ===>
00001 /* **********************************************
00002  * This file starts an Application Server for the
00003  * default service on port 5001.
00004  * ********************************************** */
00005
00006 /* **********************************************
00007  * The ifcexist macro is defined so that catalogs
00008  * can be conditionally included in a proglibs
00009  * statement.
00010  * ********************************************** */
00011 %macro ifcexist(catname);
00012   %local catname;
00013   %if %sysfunc(cexist(&catname)) %then &catname;
00014 %mend;
00015
00016 %let rc=%sysfunc(ntlog(INFORMATION,Default SAS Server Port 5001.));
00017 proc appsrv port=5001 unsafe='&";%''';
00018   allocate file sample '!SASROOT\intrnet\sample';
00019   allocate library samplib '!SASROOT\intrnet\sample' access=readonly;
00020   allocate library sampdat '!SASROOT\intrnet\sample' access=readonly;
00021   allocate library tmplib 'C:\Documents and Settings\<user>\My Documents\
00022   allocate file logfile '..\logs\%a_%p.log';
00023   proglibs sample samplib %ifcexist(sashelp.webeis) sashelp.webprog;
00024   datalibs sampdat tmplib;
00025   log file=logfile;
```

Line	Function
Line 17:	Starts PROC APPSRV and listens at port 5001.
Line 18:	Allocates a file library to reference .SAS programs.
Line 19 – 21:	Allocates data and program libraries.
Line 23:	Sets program libraries.
Line 24:	Sets data libraries.
Line 25:	Sets the logfile.

You can and will need to allocate your own program and data libraries.

The program to add libraries would look like the following.



Program Editor - appstart.sas

```
Command ===>
00001 %macro ifcexist(catname);
00002   %local catname;
00003   %if %sysfunc(cexist(&catname)) %then &catname;
00004 %mend;
00005
00006 %let rc=%sysfunc(ntlog(INFORMATION,SAS/IntrNet Application Server started for the socket service.));
00007 proc appsrv  unsafe='&\";%'''
00008   &sysparm;
00009   allocate library saved 'c:\data\sas\data8';
00010   allocate file myprogs 'c:\programs';
00011   allocate file sample '!SASROOT\intrnet\sample';
00012   allocate library samplib '!SASROOT\intrnet\sample' access=readonly;
00013   allocate library sampdat '!SASROOT\intrnet\sample' access=readonly;
00014   allocate library tmplib 'C:\Program Files\SAS\IntrNet\socket\temp';
00015   allocate file logfile 'C:\Program Files\SAS\IntrNet\socket\logs\%a_%p.log';
00016   proglibs myprogs sample samplib %ifcexist(sashelp.webeis) sashelp.webprog;
00017   proglibs sashelp.websdk1;
00018   adminlibs sashelp.webadmn;
00019   datalibs saved sampdat tmplib;
00020   log file=logfile;
00021 run;
00022 %let rc=%sysfunc(ntlog(INFORMATION,SAS/IntrNet Application Server stopped for the socket service.));
```

Line	Function
Line 10:	Allocates a location called myprogs. The use of allocate file indicates that this location will contain .SAS programs
Line 9:	Allocates a library called saved.
Line 16	Designates these libraries as Program Libraries. Multiple proglibs lines are allowed.
Line 19	Designates these libraries as data libraries. Multiple datalibs statements are allowed.
	Libraries defined here are available only in this Application Server.

Starting Your Service

There are several ways to run your SAS/INTRNET Server.

1. Start SAS. Open the proc appsrv program you have created, and run it.

2. Use the Shortcut SAS prepares for you.

3. Setup the SAS/INTRNET Server as a Windows NT/2000 Service.

We will use the Shortcut SAS creates for us.

Click on Start on the Taskbar, then Programs, then The SAS System, then INTRNET, then Start Default Service.

Editing the Broker Configuration File

The broker is an executable program that is placed in a directory under your Web Server.

In our example it is located in C:\Program Files\Apache Group\Apache\cgi-bin.

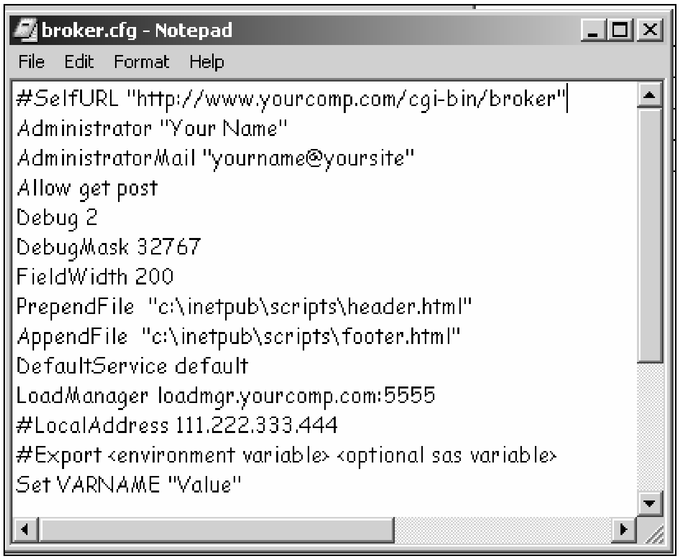
The broker.CFG file is the configuration file for this program.

It is located in the same directory.

It contains several global variables and then a set of entries for each service.

When the file is installed from SAS it contains a lot of comments.

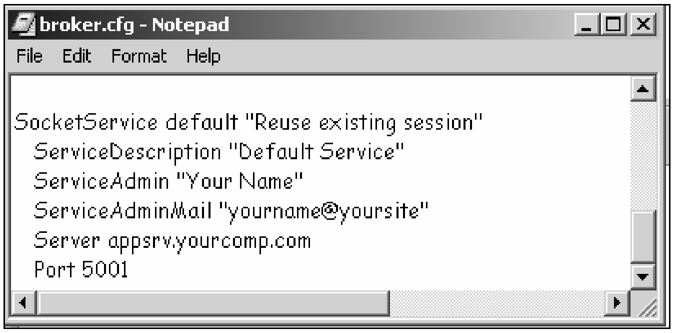
In the following pages the GLOBAL options are displayed. The comments have been removed.



```
#SelfURL "http://www.yourcomp.com/cgi-bin/broker"
Administrator "Your Name"
AdministratorMail "yourname@yoursite"
Allow get post
Debug 2
DebugMask 32767
FieldWidth 200
PrependFile  "c:\inetpub\scripts\header.html"
AppendFile  "c:\inetpub\scripts\footer.html"
DefaultService default
LoadManager loadmgr.yourcomp.com:5555
#LocalAddress 111.222.333.444
#Export <environment variable> <optional sas variable>
Set VARNAME "Value"
```

Option	Function
SelfURL	Sets the URL of the BROKER. Normally not set.
Administrator	Name of your System Administrator
Administrator Mail	Email Address of your System Administrator
Allow get post	Allows you to toggle the get and post CGI methods
Debug	Allows you to set a default value for Debug
DebugMask	Sets the maximum allowable value for Debug
FieldWidth	Sets the field width for variables being passed. Default is 80.
PrependFile	File inserted at the beginning of every HTML page generated.
AppendFile	File inserted at the end of every HTML page generated.
DefaultService	Default service used in the event no service value is available
LoadManager	Specifies the machine/port that the Load Manager is running
LocalAddress	Overrides the automatic determination of the local host IP Address.
Export	Allows you to export any environment variable.
Set	Allows you to specify any other variables you want.

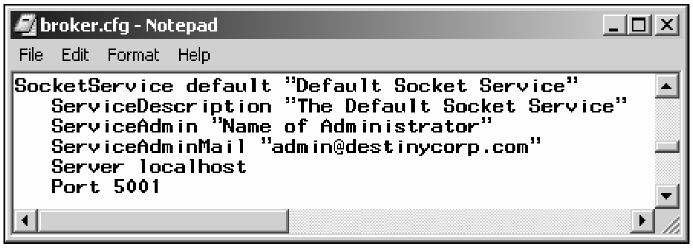
The typical Socket Service entry appears below.



```
SocketService default "Reuse existing session"
   ServiceDescription "Default Service"
   ServiceAdmin "Your Name"
   ServiceAdminMail "yourname@yoursite"
   Server appsrv.yourcomp.com
   Port 5001
```

Option	Function
Service Description	Description of the Service
Service Admin	Name of the System Administrator
ServiceAdminMail	Email Address of your System Administrator
Server	The name of the Server running the Application Server.(Required). Multiple Servers are allowed.
Port	The port number the Application Server is listening too. (Required). Multiple ports are allowed.
	You can override the Global options here. If they are left blank then the defaults are used.

The socket service after editing is displayed below.



```
SocketService default "Default Socket Service"
   ServiceDescription "The Default Socket Service"
   ServiceAdmin "Name of Administrator"
   ServiceAdminMail "admin@destinycorp.com"
   Server localhost
   Port 5001
```

Testing Your Service

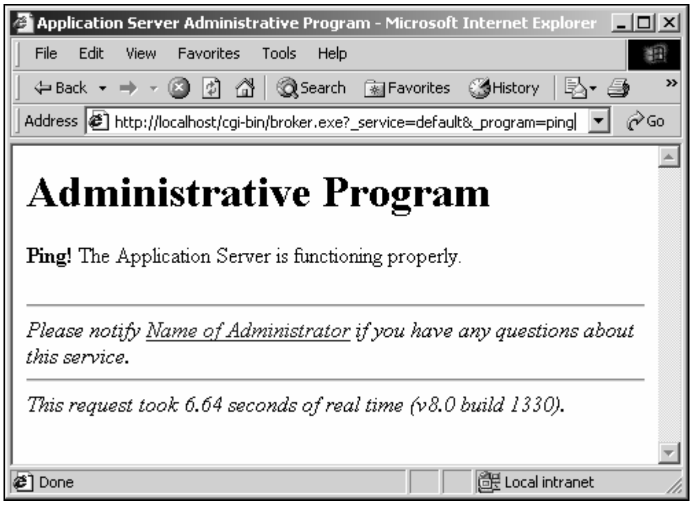
SAS provides the PING program to test your service.

To run it you need to open your browser and enter the following:

http://localhost/cgi-bin/broker.exe?_service=default&_program=ping

Where

- Localhost/cgi-bin/broker.exe represents the URL of your BROKER executable.
- _service represents your service name.
- _program represents your program name.


Application Server Administrative Program - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://localhost/cgi-bin/broker.exe?_service=default&_program=ping
Go
Done
Local intranet

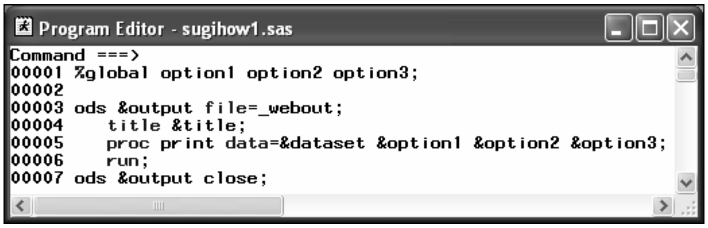
Administrative Program

Ping! The Application Server is functioning properly.

Please notify Name of Administrator if you have any questions about this service.

This request took 6.64 seconds of real time (v8.0 build 1330).

First, the program needs to be modified to use ODS statements and several macro variables that will be passed from the HTML screen. We choose to specify a title, data set, three options and type of output.

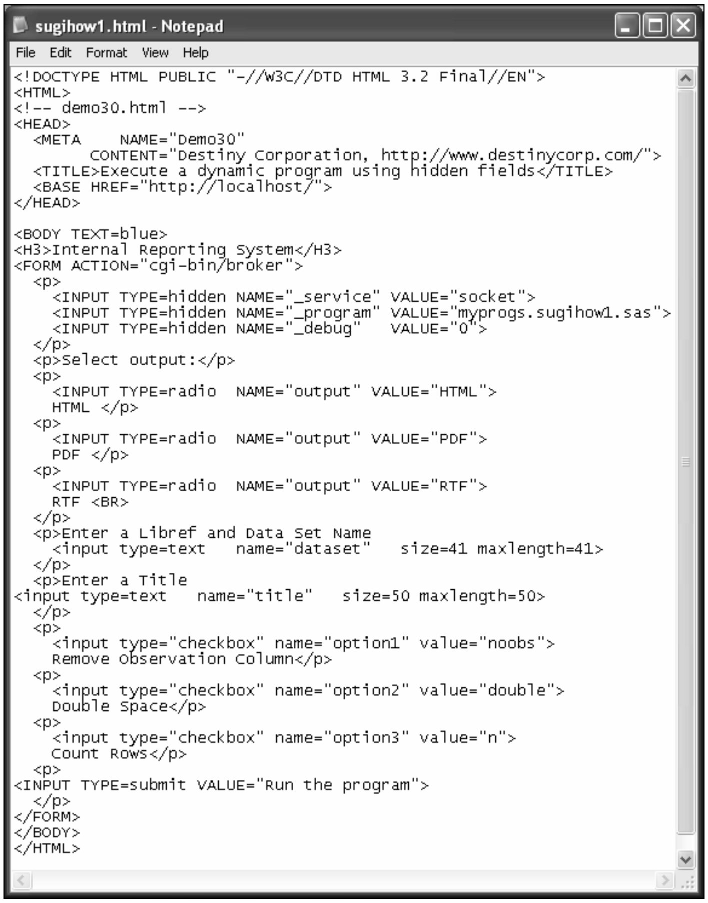


```
Command ===>
%global option1 option2 option3;

ods &output file=_webout;
    title &title;
    proc print data=&dataset &option1 &option2 &option3;
    run;
ods &output close;
```

The code has been modified to use several global options in case a user does not select some options on the HTML web page.

The follow HTML code is created to build the web page listed earlier in this paper. The specifics for SAS and how it ties to the web page will be discussed further in the workshop.

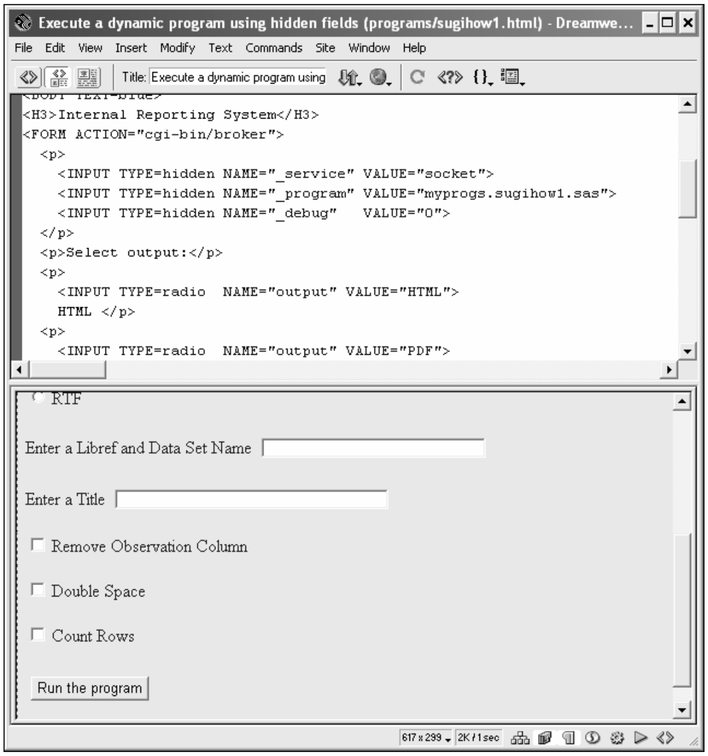


```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
<!-- demo30.html -->
<HEAD>
  <META     NAME="Demo30"
        CONTENT="Destiny Corporation, http://www.destinycorp.com/">
  <TITLE>Execute a dynamic program using hidden fields</TITLE>
  <BASE HREF="http://localhost/">
</HEAD>

<BODY TEXT=blue>
<H3>Internal Reporting System</H3>
<FORM ACTION="cgi-bin/broker">
  <p>
    <INPUT TYPE=hidden NAME="_service" VALUE="socket">
    <INPUT TYPE=hidden NAME="_program" VALUE="myprogs.sugihow1.sas">
    <INPUT TYPE=hidden NAME="_debug"   VALUE="0">
  </p>
  <p>Select output:</p>
  <p>
    <INPUT TYPE=radio  NAME="output" VALUE="HTML">
    HTML </p>
  <p>
    <INPUT TYPE=radio  NAME="output" VALUE="PDF">
    PDF </p>
  <p>
    <INPUT TYPE=radio  NAME="output" VALUE="RTF">
    RTF <BR>
  </p>
  <p>Enter a Libref and Data Set Name
    <input type=text   name="dataset"   size=41 maxlength=41>
  </p>
  <p>Enter a Title
<input type=text   name="title"   size=50 maxlength=50>
  </p>
  <p>
    <input type="checkbox" name="option1" value="noobs">
    Remove Observation Column</p>
  <p>
    <input type="checkbox" name="option2" value="double">
    Double Space</p>
  <p>
    <input type="checkbox" name="option3" value="n">
    Count Rows</p>
  <p>
<INPUT TYPE=submit VALUE="Run the program">
  </p>
</FORM>
</BODY>
</HTML>
```

The key areas to focus on are the hidden tags that specifically call SAS.

The following screen is how this looks in Macromedia's Dreamweaver software. Dreamweaver software provides an easy to use HTML interface with a menuing system to allow for easy modification of HTML.


Execute a dynamic program using hidden fields (programs/sugihow1.html) - Dreamwe...
File Edit View Insert Modify Text Commands Site Window Help
Title: Execute a dynamic program using
617 x 299
2K / 1 sec

```
<H3>Internal Reporting System</H3>
<FORM ACTION="cgi-bin/broker">
  <p>
    <INPUT TYPE=hidden NAME="_service" VALUE="socket">
    <INPUT TYPE=hidden NAME="_program" VALUE="myprogs.sugihow1.sas">
    <INPUT TYPE=hidden NAME="_debug"   VALUE="0">
  </p>
  <p>Select output:</p>
  <p>
    <INPUT TYPE=radio  NAME="output" VALUE="HTML">
    HTML </p>
  <p>
    <INPUT TYPE=radio  NAME="output" VALUE="PDF">
```

○ RTF

Enter a Libref and Data Set Name

Enter a Title

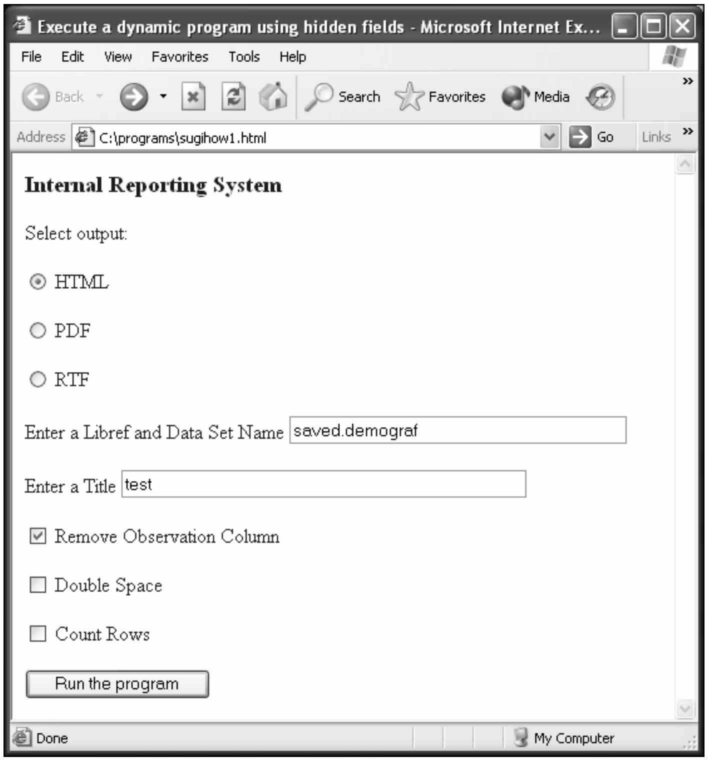
☐ Remove Observation Column

☐ Double Space

☐ Count Rows

Run the program

When all of the pieces are put in place, the resulting HTML screen


Execute a dynamic program using hidden fields - Microsoft Internet Ex...
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
C:\programs\sugihow1.html
Go
Links
Done
My Computer

Internal Reporting System

Select output:

● HTML

○ PDF

○ RTF

Enter a Libref and Data Set Name saved.demograf

Enter a Title test

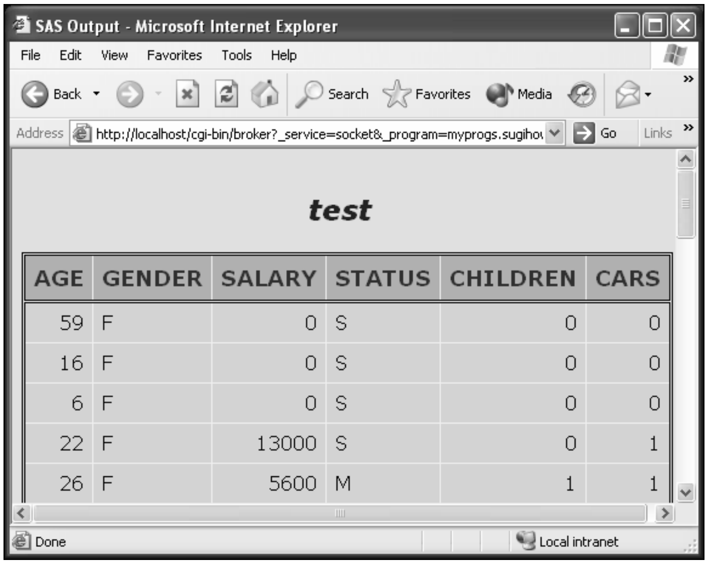
☑ Remove Observation Column

☐ Double Space

☐ Count Rows

Run the program

Will produce the following report.


SAS Output - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
Go
Links
Done
Local intranet

http://localhost/cgi-bin/broker?_service=socket&_program=myprogs.sugihou

test

AGE	GENDER	SALARY	STATUS	CHILDREN	CARS
59	F	0	S	0	0
16	F	0	S	0	0
6	F	0	S	0	0
22	F	13000	S	0	1
26	F	5600	M	1	1

For more information, please come to the Hands on Workshop or contact Destiny Corporation 800-787-2464.

Trademarks:

SAS is a registered trademark of SAS Institute, Cary, NC.
Dreamweaver is a registered trademark of Macromedia, Inc., San Francisco, CA