

# Creating Internet Applications with SAS/Intrnet Software: What does it take?

**David Ghan**  
**Education, SAS Canada**  
**416 307-4515**  
**David.Ghan@sas.com**

# Creating Internet Applications with SAS/Intrnet Software: What does it take?

## *Premise:*

*If you are a SAS programmer, it is relatively easy to apply that knowledge to create Internet applications for common use.*

## **Objectives:**

**Show how SAS programming is used.**

**Explain some essential basics of:**

- internet URL requests
- HTML

*(This, you already know)*

*(This will be pretty easy to pick up)*

# Static requests

Web Server



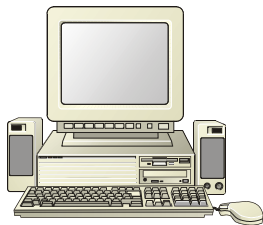
The file is returned

Request for file sent

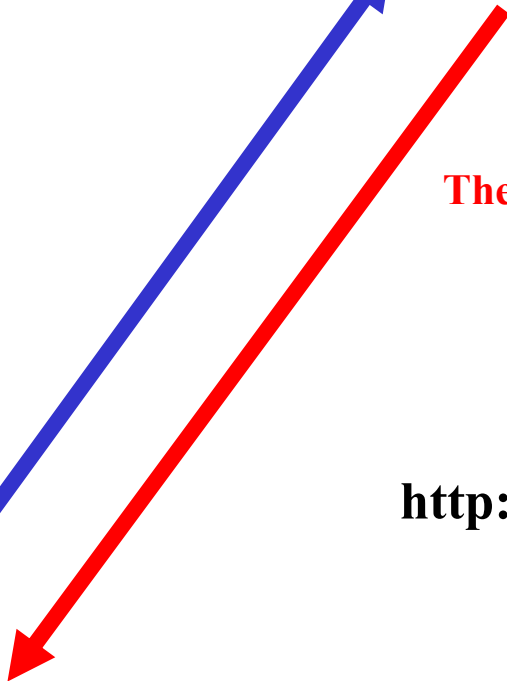
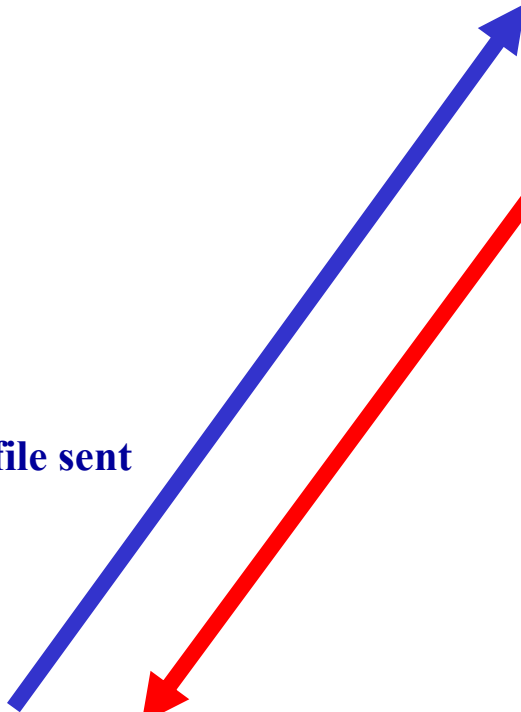
`http://localhost/TASSDemo/Hello.html`

webservice

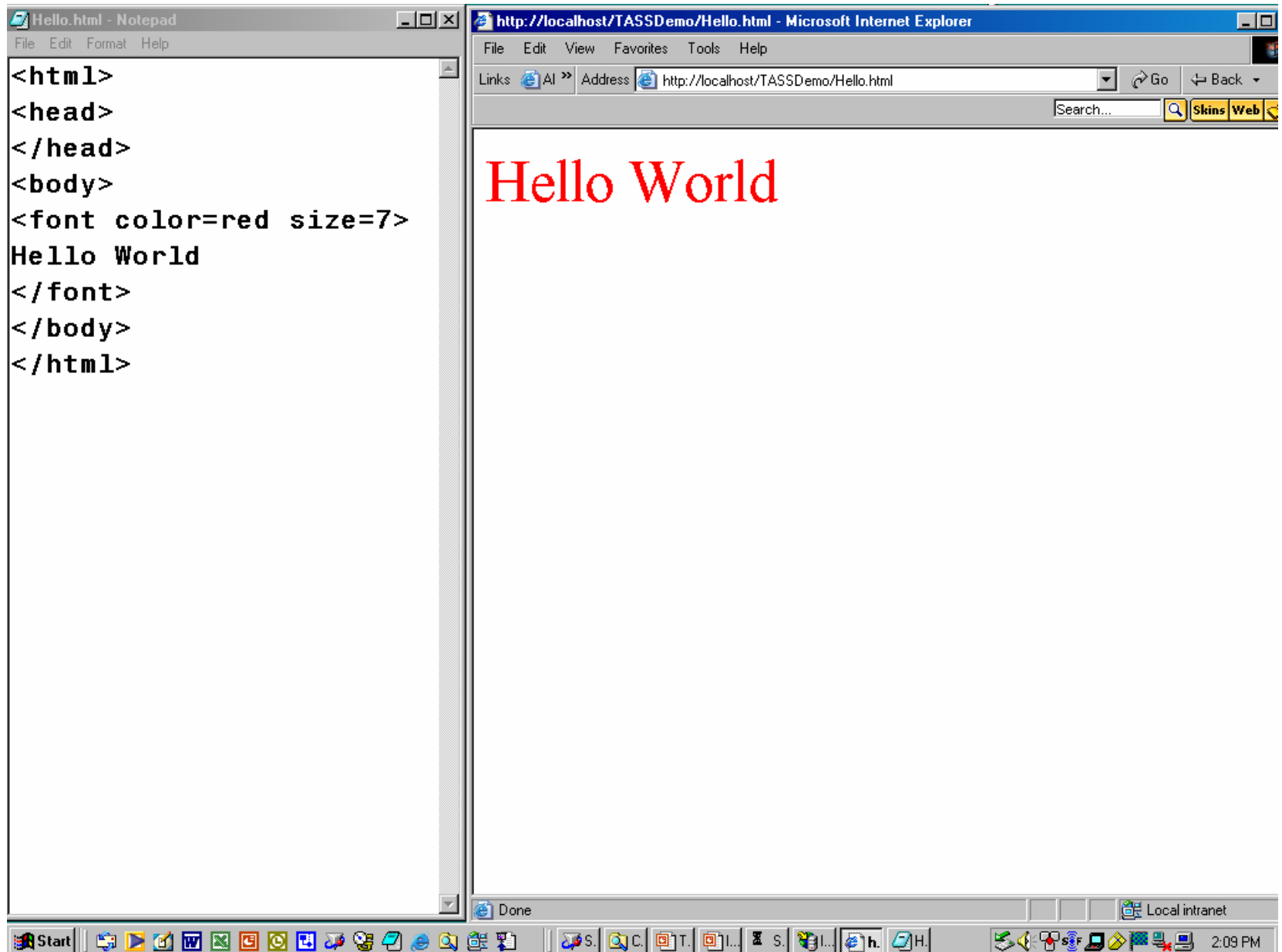
static file  
on webservice



Client machine



## The “Hello.html” file and how it is displayed in the browser



<http://localhost/TASSDemo/Hello.html>

## Methods to send URL requests

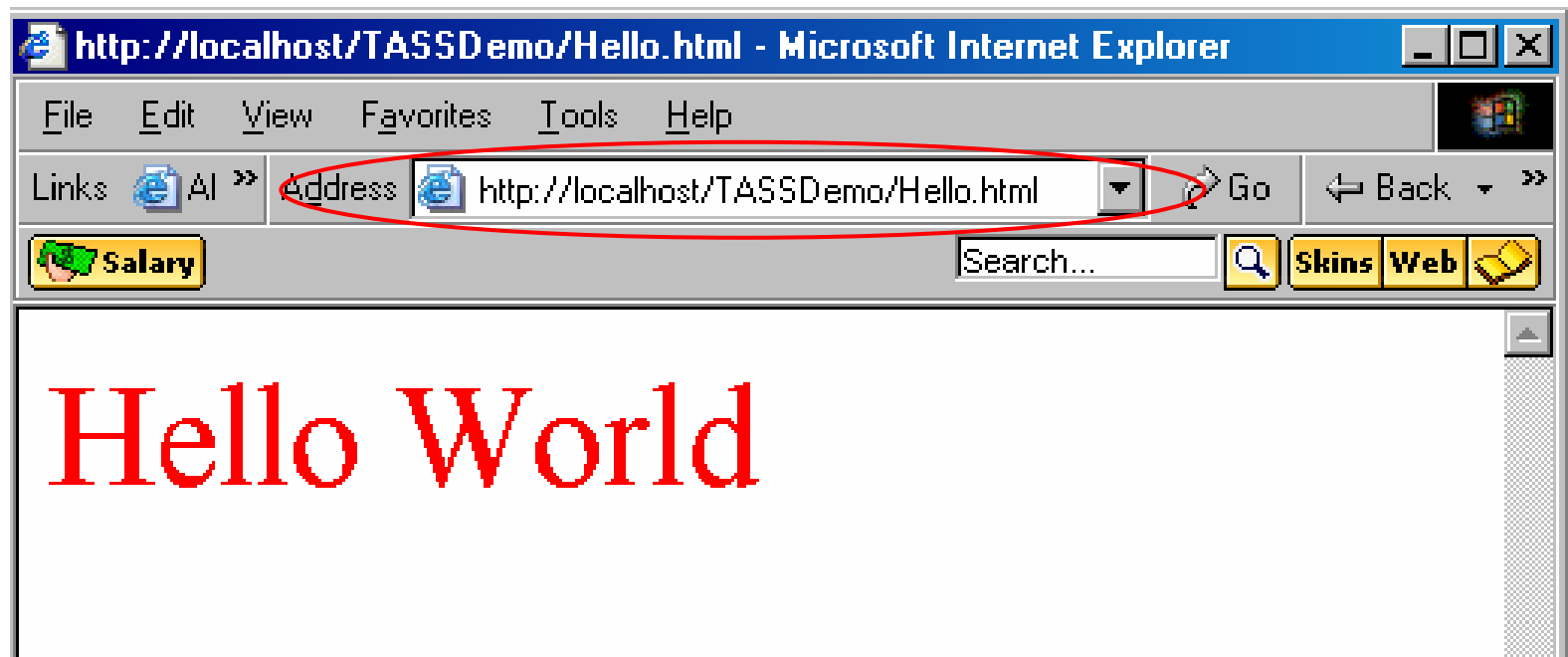
*Example URL request*

Request can be send by:

1. Typing the URL on the address line of the browser
2. Creating a hyperlink on the HTML page
3. A Submit button on an HTML form

## Methods to send URL requests

1. Typing a URL on the address line in browser:



# Methods to send URL requests

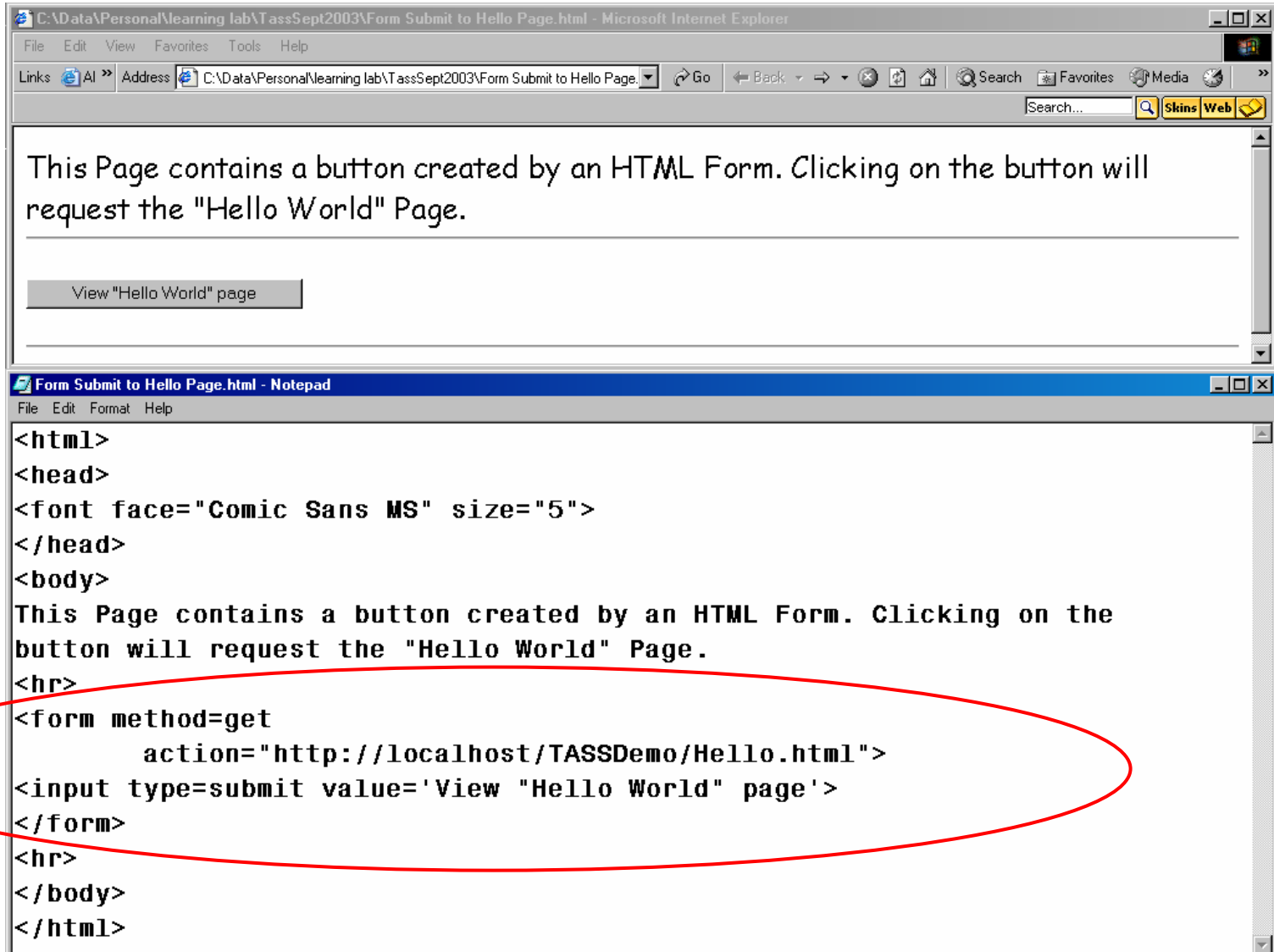
## 2. Creating a hyperlink on the HTML page

The image shows two overlapping windows. The top window is Microsoft Internet Explorer displaying a web page with the text "Click on the link below to request the 'Hello World' page:" followed by a horizontal line and a purple underlined hyperlink "View 'Hello World' page". An orange arrow points from the word "hyperlink" to this link. The bottom window is Notepad showing the HTML source code for the page. The code includes a head section with font settings and a body section with the same text as the browser. The anchor tag `<a href="http://localhost/TASSDemo/Hello.html">View "Hello World" page</a>` is circled in red, with an orange arrow pointing from the word "hyperlink" to it. The Windows taskbar at the bottom shows the Start button and various application icons, with the time 7:29 PM.

```
<html>
<head>
<font face="Comic Sans MS" size="6">
</head>
<body>
Click on the link below to request the "Hello World" page:
<hr>
<br>
<a href="http://localhost/TASSDemo/Hello.html">View "Hello World"
page</a>
</body>
```

# Methods to send URL requests

## 3. A Submit button on an HTML form



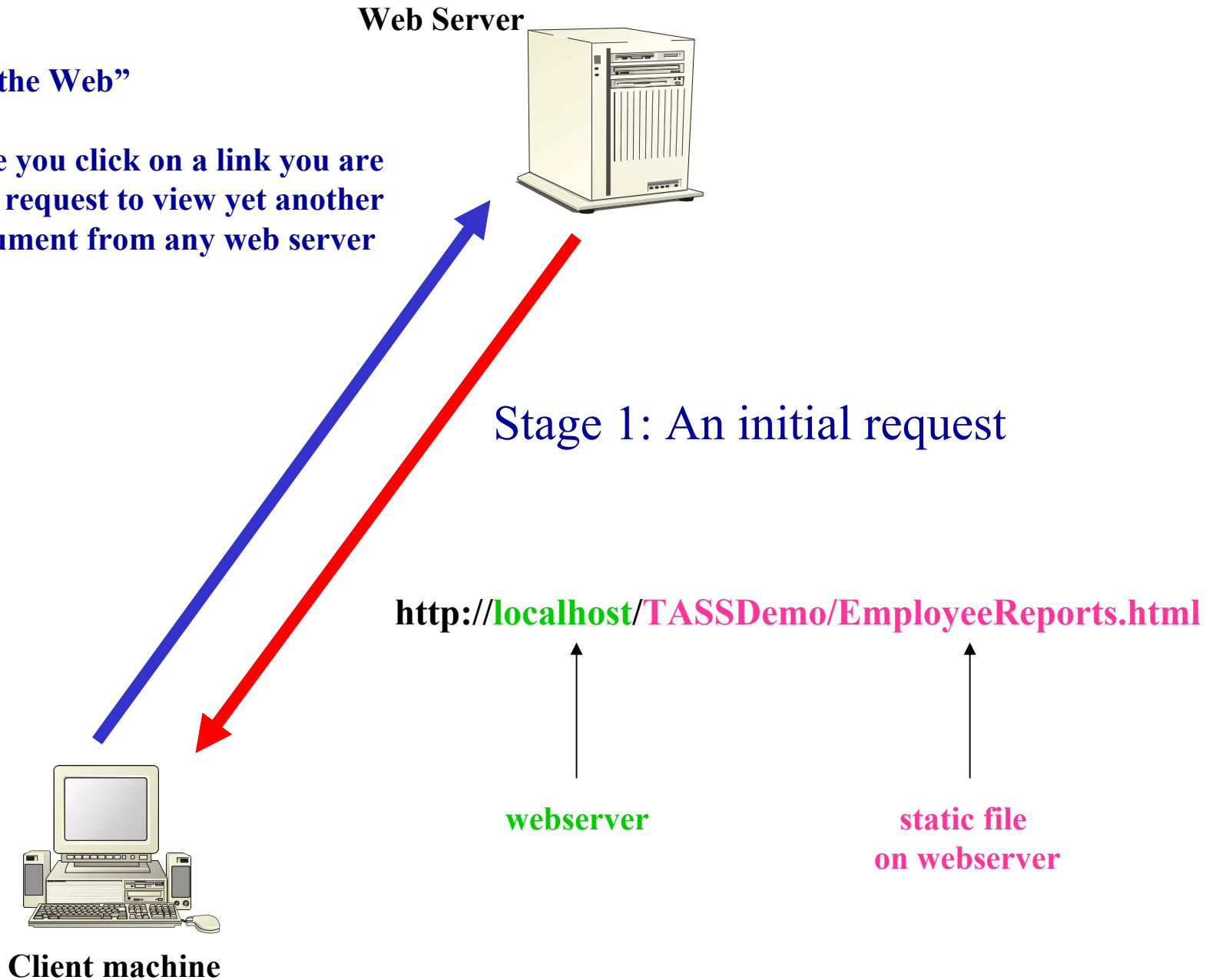
The image shows two overlapping windows. The top window is Microsoft Internet Explorer, displaying a page with the text: "This Page contains a button created by an HTML Form. Clicking on the button will request the 'Hello World' Page." Below the text is a horizontal line and a button labeled "View 'Hello World' page". The bottom window is Notepad, showing the HTML source code for the page. The code is as follows:

```
<html>
<head>
<font face="Comic Sans MS" size="5">
</head>
<body>
This Page contains a button created by an HTML Form. Clicking on the
button will request the "Hello World" Page.
<hr>
<form method=get
      action="http://localhost/TASSDemo/Hello.html">
<input type=submit value='View "Hello World" page'>
</form>
<hr>
</body>
</html>
```

A red oval highlights the form-related code in the Notepad window, specifically the `<form>` and `<input type=submit>` tags.

## “Surfing the Web”

Each time you click on a link you are making a request to view yet another html document from any web server machine.



http://localhost/TASSDemo/employeeReports.html - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Links Address http://localhost/TASSDemo/employeeReports.html Go Back Forward Stop Home Search Favorites Media

Salary Search... Skins Web

Click on the links below to request the report

---

[View Salary Summaries by Jobcode](#)

[View Age Summaries by Jobcode](#)

Local intranet

EmployeeReports.html - Notepad

File Edit Format Help

```
<html>
<head>
<font face="Comic Sans MS" size="6">
</head>
<body>
Click on the links below to request the report
<hr>
<br>
<a href="http://localhost/TASSDemo/MeanSalaryByJobcode.html">View Salary Summaries by
Jobcode</a>
<br>
<a href="http://localhost/TASSDemo/MeanAgeByJobcode.html">View Age Summaries by
Jobcode</a>
</body>
</html>
```

**Web Server**

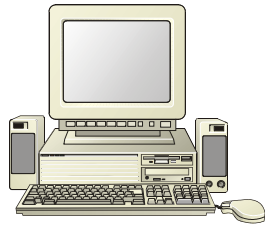


Stage 2: Click on a Link for a Subsequent Request

**http://localhost/TASSDemo/MeanSalaryByJobcode.html**

**webservice**

**static file  
on webservice**



**Client machine**

**Demo: [employeereports.html](#)**

*Mean Salary of Employees for Different Job Types*

What is this file?

*Another html file delivered by the Web Server.*

Analysis Variable : Salary Employee Salary						
Job Code	N Obs	N	Mean	Std Dev	Minimum	Maximum
BAGCLK	140	140	31442.86	7961.48	16000.00	45000.00
BAGSUP	18	18	28000.00	8983.65	16000.00	42000.00
CHKCLK	125	125	30168.00	8447.88	16000.00	45000.00
CHKSUP	18	18	29277.78	7094.37	17000.00	43000.00
FACCLK	124	124	31572.58	8908.91	16000.00	45000.00
FACMGR	17	17	31588.24	6819.63	21000.00	43000.00
FACMNT	60	60	29916.67	8839.99	16000.00	45000.00
FINACT	36	36	30166.67	8929.89	17000.00	44000.00
FINCLK	53	53	31811.32	8974.44	16000.00	45000.00
FINMGR	20	20	28950.00	8999.85	16000.00	45000.00
FLSCHD	32	32	30687.50	10234.15	16000.00	44000.00
FLSMGR	8	8	27250.00	8172.25	17000.00	38000.00
FLTAT1	62	62	32612.90	8061.96	17000.00	45000.00
FLTAT2	63	63	29079.37	8231.14	16000.00	43000.00
FLTAT3	97	97	30206.19	8176.36	16000.00	45000.00
FSVCLK	46	46	31086.06	8186.53	17000.00	44000.00

*Mean Salary of Employees for Different Job Types*

So, how was this file created?

*By a SAS programmer, using ODS.*

Analysis Variable : Salary Employee Salary						
Job Code	N Obs	N	Mean	Std Dev	Minimum	Maximum
BAGCLK	140	140	31442.86	7961.48	16000.00	45000.00
BAGSUP	18	18	28000.00	8983.65	16000.00	42000.00
CHKCLK	125	125	30168.00	8447.88	16000.00	45000.00
CHKSUP	10	10	29977.70	7894.97	17000.00	42000.00

```
<TD ALIGN=LEFT VALIGN=TOP bgcolor="#CCCCCC"><font face="Verdana, Helvetica, Helv"
size="2" color="#000000">BAGCLK</font></TD>
<TD ALIGN=RIGHT bgcolor="#CCCCCC"><font face="Verdana, Helvetica, Helv" size="2"
color="#000000">140</font></TD>
<TD ALIGN=RIGHT bgcolor="#CCCCCC"><font face="Verdana, Helvetica, Helv" size="2"
color="#000000">31442.86</font></TD>
<TD ALIGN=RIGHT bgcolor="#CCCCCC"><font face="Verdana, Helvetica, Helv" size="2"
color="#000000">16000.00</font></TD>
<TD ALIGN=RIGHT bgcolor="#CCCCCC"><font face="Verdana, Helvetica, Helv" size="2"
color="#000000">45000.00</font></TD>
</TR>
<TR>
<TD ALIGN=LEFT VALIGN=TOP bgcolor="#CCCCCC"><font face="Verdana, Helvetica, Helv"
size="2" color="#000000">BAGSUP</font></TD>
<TD ALIGN=RIGHT bgcolor="#CCCCCC"><font face="Verdana, Helvetica, Helv" size="2"
```

*Here is the program:*

```
ods html file='MeanSalaryByJobcode.html' style=statdoc;  
ods noptitle;  
title ;  
proc means data=tass.employees n mean min max nonobs;  
title 'Mean Salary of Employees for Different Job Types';  
  class jobcode;  
  var salary;  
run;  
ods html close;
```

After the program runs, the resulting html file can be placed on the web server so it is available for static requests.

**Demo “program to generate static reports.sas”**

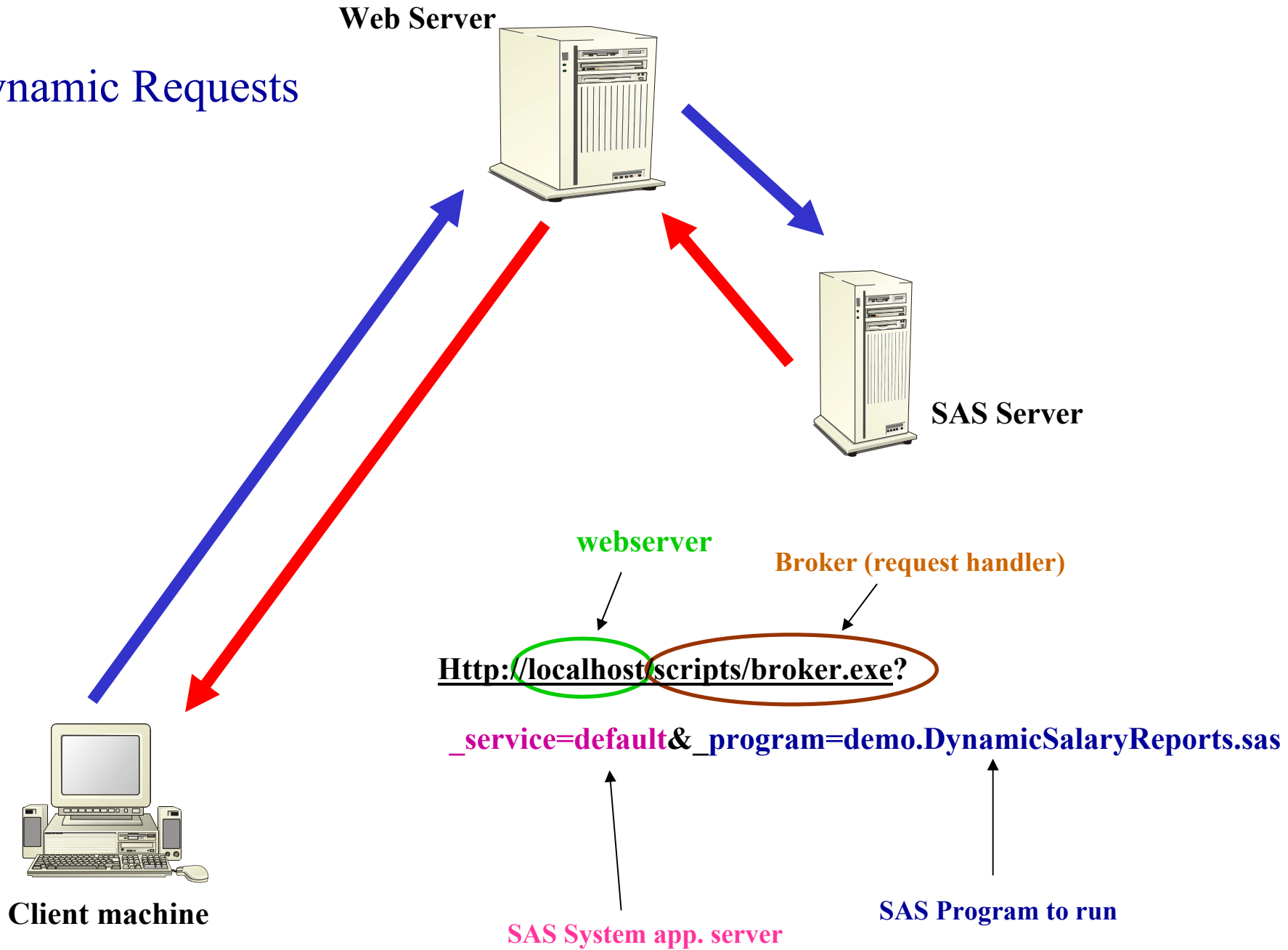
## Some limitations of the static approach

- must *pre-create* all reports and then place on the web server
- as the data *changes*, the reporting programs must be *re-run* to update and the reports must be *again* moved to the server.
- for *each* static report must provide a *unique* link. Must *pre-determine* exactly what reports will be requested.

Imagine a scenario where any given day, anywhere from 0 to 100 reports might be requested. The data that the reports are based on change often (eg: daily, or by the minute). Someone would spend much time updating the static reports and moving them to the web server. The computer processing would be intense.

So... instead .....

# Dynamic Requests



Click on the links below to request the report

---

[View Salary Summaries by Jobcode](#)  
[View Age Summaries by Jobcode](#)

When you click on this link ...

```
File Edit Format Help
<html>
<head>
<font face="Comic Sans MS" size="6">
</head>
<body>
Click on the links below to request the report
<hr>
<br>
<a
href="http://localhost/scripts/broker.exe?_service=default&_program=demo.DynamicSalaryReports.sas">
View Salary Summaries by Jobcode</a>
<br>
<a
href="http://localhost/scripts/broker.exe?_service=default&_program=demo.DynamicAgeReport.sas">View
Age Summaries by Jobcode</a>
</body>
```

# Web Server

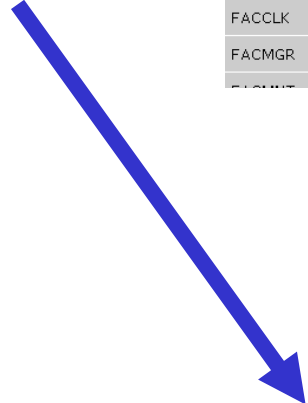
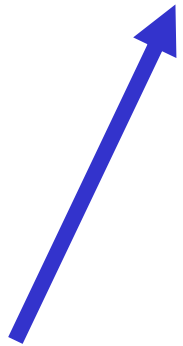


Http://localhost/scripts/broker.exe?

\_service=default&\_program=demo.DynamicSalaryReports.sas

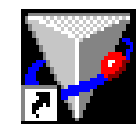
Mean Salary of Employees for Different Job Types

Analysis Variable : Salary Employee Salary				
Job Code	N	Mean	Minimum	Maximum
BAGCLK	140	31442.86	16000.00	45000.00
BAGSUP	18	28000.00	16000.00	42000.00
CHKCLK	125	30168.00	16000.00	45000.00
CHKSUP	18	29277.78	17000.00	43000.00
FACCLK	124	31572.58	16000.00	45000.00
FACMGR	17	31588.24	21000.00	43000.00

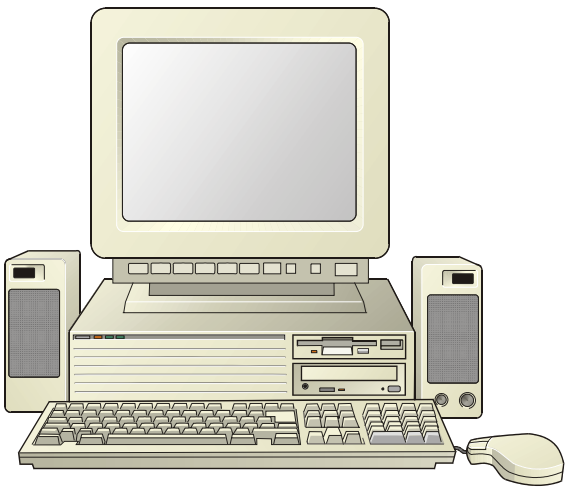


Http://localhost/scripts/broker.exe?

\_service=default&\_program=demo.DynamicSalaryReports.sas



# SAS Server

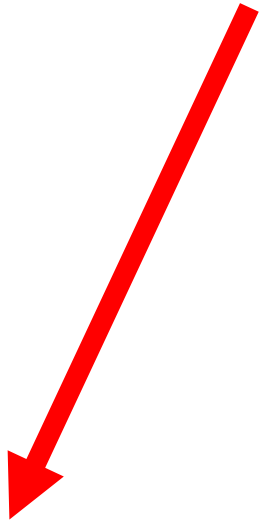


# Client machine

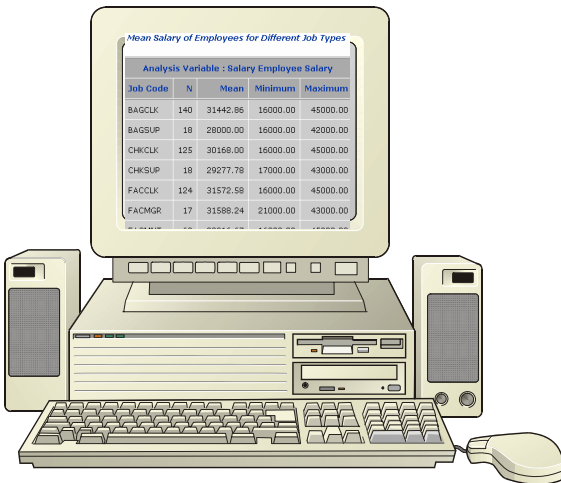
# Web Server

Mean Salary of Employees for Different Job Types

Analysis Variable : Salary Employee Salary				
Job Code	N	Mean	Minimum	Maximum
BAGCLK	140	31442.86	16000.00	45000.00
BAGSUP	18	28000.00	16000.00	42000.00
CHKCLK	125	30168.00	16000.00	45000.00
CHKSUP	18	29277.78	17000.00	43000.00
FACCLK	124	31572.58	16000.00	45000.00
FACMGR	17	31588.24	21000.00	43000.00



**SAS Server**



**Client machine**

The dynamic SAS program that runs on the SAS Server is very similar to the static program we used earlier:

### Earlier Static Program:

```
ods html file='MeanSalaryByJobcode.html'  
style=statdoc;  
ods noptitle;  
title ;  
proc means data=tass.employees  
           n mean min max nonobs;  
  title 'Mean Salary of Employees for Different Job Types';  
  class jobcode;  
  var salary;  
run;  
ods html close;
```

### The dynamic version:

```
ods html file=_webout style=statdoc;  
ods noptitle;  
title ;  
proc means data=demodata.employees  
           n mean min max nonobs;  
  title 'Mean Salary of Employees for Different Job  
Types';  
  class jobcode;  
  var salary;  
run;  
ods html close;
```

*File=\_webout ? ...*

# Web Server

Mean Salary of Employees for Different Job Types

Analysis Variable : Salary Employee Salary				
Job Code	N	Mean	Minimum	Maximum
BAGCLK	140	31442.86	16000.00	45000.00
BAGSUP	18	28000.00	16000.00	42000.00
CHKCLK	125	30168.00	16000.00	45000.00
CHKSUP	18	29277.78	17000.00	43000.00
FACCLK	124	31572.58	16000.00	45000.00
FACMGR	17	31588.24	21000.00	43000.00



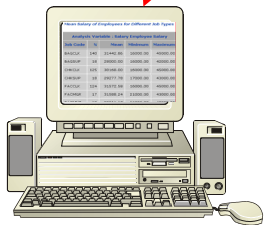
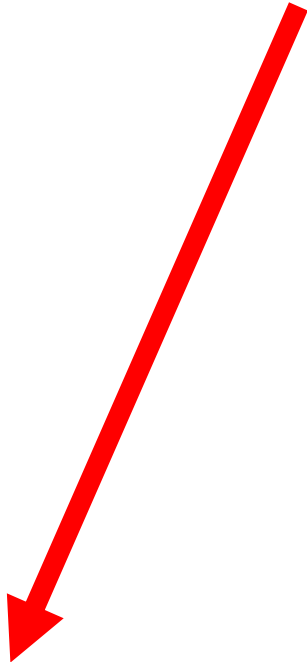
Mean Salary of Employees for Different Job Types

Analysis Variable : Salary Employee Salary				
Job Code	N	Mean	Minimum	Maximum
BAGCLK	140	31442.86	16000.00	45000.00
BAGSUP	18	28000.00	16000.00	42000.00
CHKCLK	125	30168.00	16000.00	45000.00
CHKSUP	18	29277.78	17000.00	43000.00
FACCLK	124	31572.58	16000.00	45000.00
FACMGR	17	31588.24	21000.00	43000.00



SAS Server

the fileref **\_webout** directs output from the *SAS Server* back to the *Web Server*, which in turn sends the result back to the *Client browser* for display.



Client machine

# More dynamic SAS/Intrnet SAS programs using the \_webout fileref:

## Graphs:

```
goptions device=activex hsize=25 cm vsize=14 cm;  
axis1 label=none;
```

```
ods html file=_webout style=statdoc;
```

```
title color=blue font="Comic Sans MS" height=2 'Average Salary for Airport  
Operations Employees';
```

```
title2 color=green font="Comic Sans MS" height=2 'Broken down by Job Category';
```

```
proc gchart data=demodata.employees;
```

```
  vbar3d jobcode / raxis=axis1 sumvar=salary type=mean patternid=midpoint;  
  where division='AIRPORT OPERATIONS' and jobcode NE 'VICEPR';
```

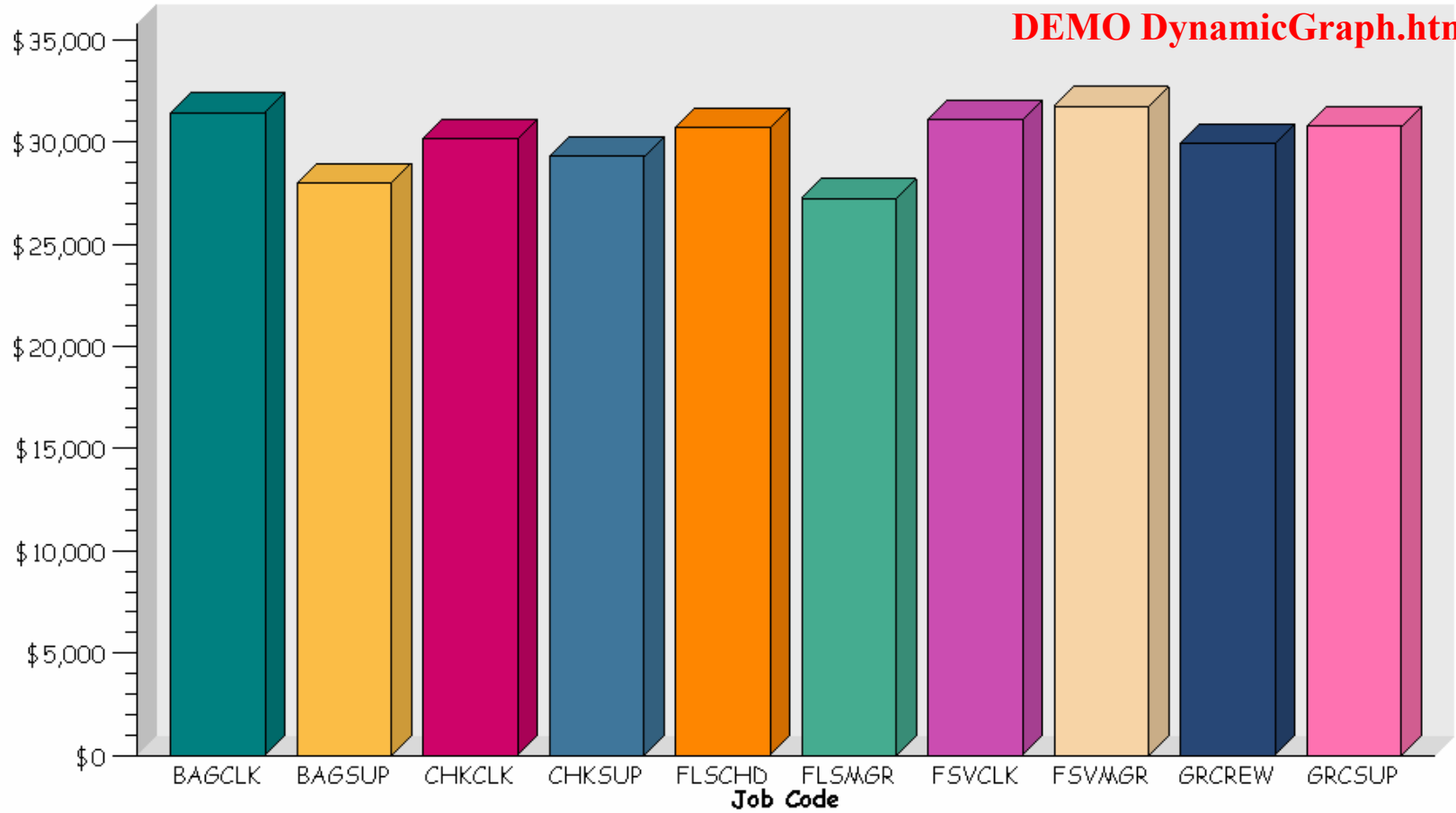
```
run;
```

```
quit;
```

```
ods html close;
```

### Average Salary for Airport Operations Employees Broken down by Job Category

DEMO DynamicGraph.html



## Custom data step reports:

```
proc sort data=demodata.employees out=emps;
  by lastname;
  where jobcode='GRCREW' and empcountry in ("CANADA", "DENMARK");
run;
ods html file = _webout style=sasweb;
title;
data _null_;
  set emps;
  file print ods=(variables=(division firstname lastname empcountry));
  if empcountry = 'CANADA' then place=emplocation;
  else place=empcountry;
  yearsemployed='years employed=' !!
    put(int((today()-hiredate)/365.25),2. -L);
  annsalary='Annual salary=' !! put(salary, dollar9.0 -L);
  put @1 division @2 firstname @3 lastname @4 place /
    @4 jobcode /
    @4 yearsemployed /
    @4 annsalary / ;

  label firstname="Firstname"
    lastname="Lastname"
    empcountry="Location";
run;
ods html close;
```

```

file print ods=(variables=(division firstname lastname empcountry));
put @1 division @2 firstname @3 lastname @4 place /
      @4 jobcode /
      @4 yearsemployed /
      @4 annsalary / ;

```

Division	Firstname	Lastname	Location
AIRPORT OPERATIONS	JOHN W.	PAUL	TORONTO
			GRCREW
			years employeed=13
			Annual salary=\$26,000
AIRPORT OPERATIONS	PHILLIP C.	PENDER	TORONTO
			GRCREW
			years employeed=22
			Annual salary=\$26,000
AIRPORT OPERATIONS	ROBERT W.	PERSAUD	TORONTO
			GRCREW
			years employeed=10
			Annual salary=\$31,000
AIRPORT OPERATIONS	HANS	PETERSEN	DENMARK
			GRCREW
			years employeed=12
			Annual salary=\$27,000

**Demo dynamicdatastep.html**

```

proc sort data=demodata.employees out=emps;
  by lastname;
  where jobcode='GRCREW' and
        empcountry in ("CANADA", "DENMARK");
run;
ods html file = _webout style=sasweb;
title;
data _null_;
  set emps;
  ...
  ...

```

Division	Firstname	Lastname	Location
AIRPORT OPERATIONS	JOHN W.	PAUL	TORONTO
			GRCREW
			years employeeed=13
			Annual salary=\$26,000
AIRPORT OPERATIONS	PHILLIP C.	PENDER	TORONTO
			GRCREW
			years employeeed=22
			Annual salary=\$26,000
AIRPORT OPERATIONS	ROBERT W.	PERSAUD	TORONTO
			GRCREW

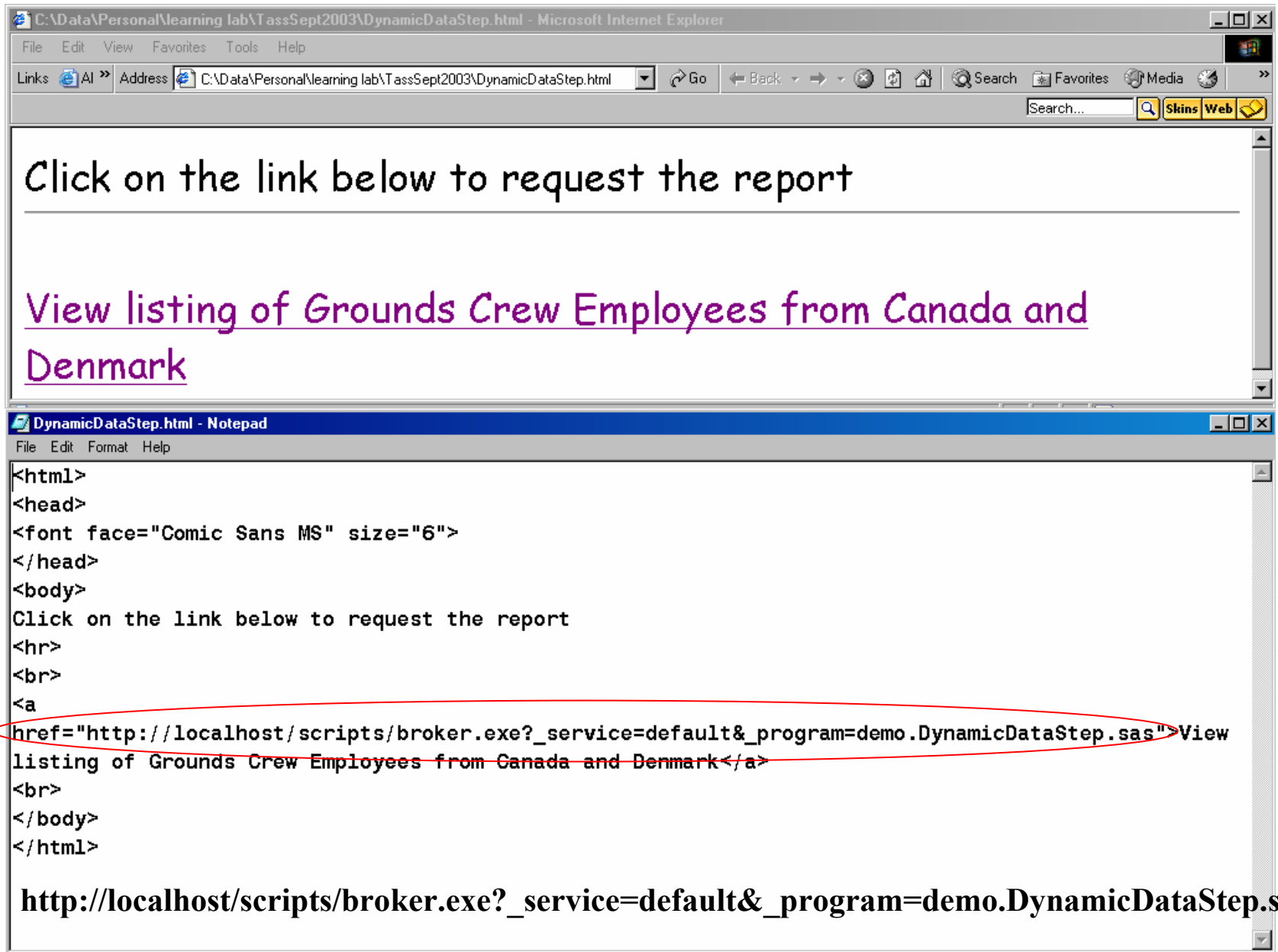


*This report for Canadian and Danish Ground Crew only.*

**Could I allow the application user to choose who the report is created for?**

*With SAS/Intrnet, you can pass SAS macro variables as part of the URL*

# Currently:



The image shows two overlapping windows. The top window is Microsoft Internet Explorer displaying a web page with the text "Click on the link below to request the report" and a purple underlined link: "View listing of Grounds Crew Employees from Canada and Denmark". The bottom window is Notepad showing the HTML source code for the page. A red oval highlights the href attribute of the link in the code: href="http://localhost/scripts/broker.exe?\_service=default&\_program=demo.DynamicDataStep.sas".

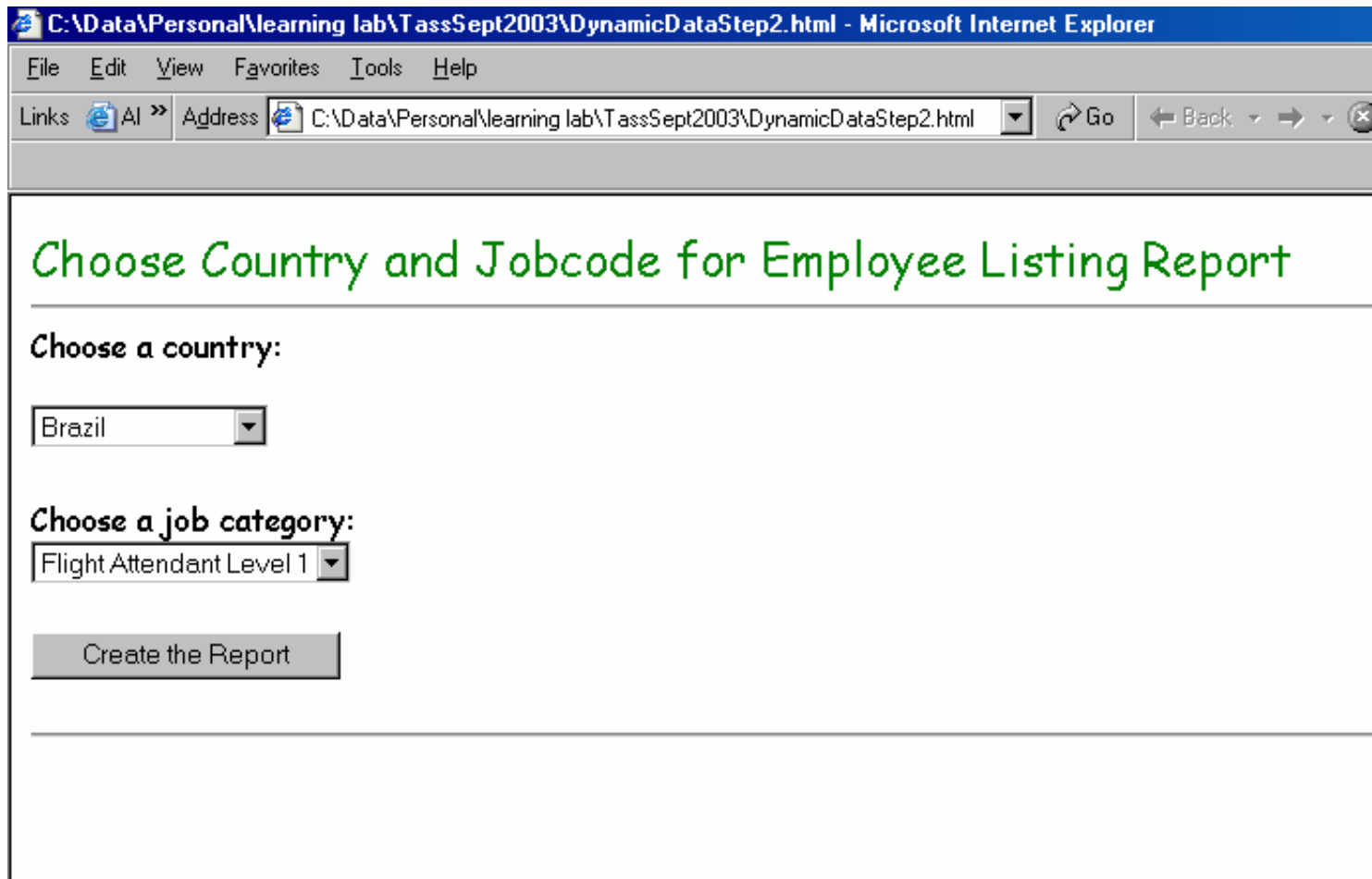
Click on the link below to request the report

[View listing of Grounds Crew Employees from Canada and Denmark](#)

```
<html>
<head>
<font face="Comic Sans MS" size="6">
</head>
<body>
Click on the link below to request the report
<hr>
<br>
<a
href="http://localhost/scripts/broker.exe?_service=default&_program=demo.DynamicDataStep.sas">View
listing of Grounds Crew Employees from Canada and Denmark</a>
<br>
</body>
</html>
```

[http://localhost/scripts/broker.exe?\\_service=default&\\_program=demo.DynamicDataStep.sas](http://localhost/scripts/broker.exe?_service=default&_program=demo.DynamicDataStep.sas)

Instead allow users to input parameters to the SAS program:



The screenshot shows a Microsoft Internet Explorer browser window. The title bar reads "C:\Data\Personal\learning lab\T assSept2003\DynamicDataStep2.html - Microsoft Internet Explorer". The address bar contains the file path "C:\Data\Personal\learning lab\T assSept2003\DynamicDataStep2.html". The main content area displays a form titled "Choose Country and Jobcode for Employee Listing Report". The form includes two dropdown menus: "Choose a country:" with "Brazil" selected, and "Choose a job category:" with "Flight Attendant Level 1" selected. Below the dropdowns is a button labeled "Create the Report".

[http://localhost/scripts/broker.exe?country=BRAZIL&job=FLTAT1&\\_service=default&\\_program=demo.dynamicDatastep2.sas](http://localhost/scripts/broker.exe?country=BRAZIL&job=FLTAT1&_service=default&_program=demo.dynamicDatastep2.sas)

## HTML FORMS:

```
<form method=get action="http://localhost/scripts/broker.exe">
<select name="country">
<option value="all" selected>all
<option value="AUSTRALIA">Australia
<option value="BRAZIL">Brazil
<option value="CANADA">Canada
<option value="CHINA">China
<option value="DENMARK">Denmark
<option value="FRANCE">France
<option value="GERMANY">Germany
<option value="ITALY">Italy
<option value="JAPAN">Japan
<option value="SOUTH AFRICA">South Africa
<option value="SWITZERLAND">Switzerland
<option value="UNITED KINGDOM">United Kingdom
<option value="USA">USA
</select>
<input type=hidden name=_service value=default>
<input type=hidden name=_program value=demo.dynamicDatastep2.sas>
<input type=submit value="Create the Report">
</form>
```

Choose a country:

Brazil ▼

Choose a job category:

Flight Attendant Level 1 ▼

Create the Report

[http://localhost/scripts/broker.exe?country=BRAZIL&job=FLTAT1&\\_service=default&\\_program=demo.dynamicDatastep2.sas](http://localhost/scripts/broker.exe?country=BRAZIL&job=FLTAT1&_service=default&_program=demo.dynamicDatastep2.sas)

[http://localhost/scripts/broker.exe?country=BRAZIL&job=FLTAT1&\\_service=default&\\_program=demo.dynamicDatastep2.sas](http://localhost/scripts/broker.exe?country=BRAZIL&job=FLTAT1&_service=default&_program=demo.dynamicDatastep2.sas)

When the web server passes this request to the SAS server, 2 macro variables are created:

&country=BRAZIL  
and  
&job=FLTAT1

These are used to subset the data used for the report.

To see macro variable use see “dynamicDatastep2.sas”

# Creating Internet Applications with SAS/Intrnet Software: What does it take?

## *Premise:*

*If you are a SAS programmer, it is relatively easy to apply that knowledge to create Internet applications for common use.*

## **Objectives:**

**Show how SAS programming and/or SQL is used.**

**Explain some essential basics of:**

- internet URL requests
- HTML

*(This, you already know)*

*(This will be pretty easy to pick up)*

# Requirements for Static Internet Reporting

## Hardware

- A Web server
- Internet connectivity

## Software

- Base SAS for all users creating reports
- SAS Software NOT required on client browser machine

## Programming Skills

- SAS Programming Skills in reporting and ODS

# Additional Requirements for Dynamic Internet Reporting

## Hardware

- A SAS Server

## Software

- SAS/Intrnet Software
- SAS Software STILL NOT required on client browser machine

## Programming Skills

- SAS Programming Skills in macro and some HTML knowledge

Presentation will be made available at:

<http://www.sas.com/canada>

Click “events”

Then “Canadian User Groups”

or

[http://www.sas.com/offices/NA/canada/9-services\\_support/214-customer\\_value\\_en.html](http://www.sas.com/offices/NA/canada/9-services_support/214-customer_value_en.html)