

# SAS BI Content Syndication with the REST Framework

Mike Vanderlinden, Business Analytics Practice, Experis



Experis™  
ManpowerGroup



---

Twin Cities Area SAS User Group  
July 28, 2011

# Agenda

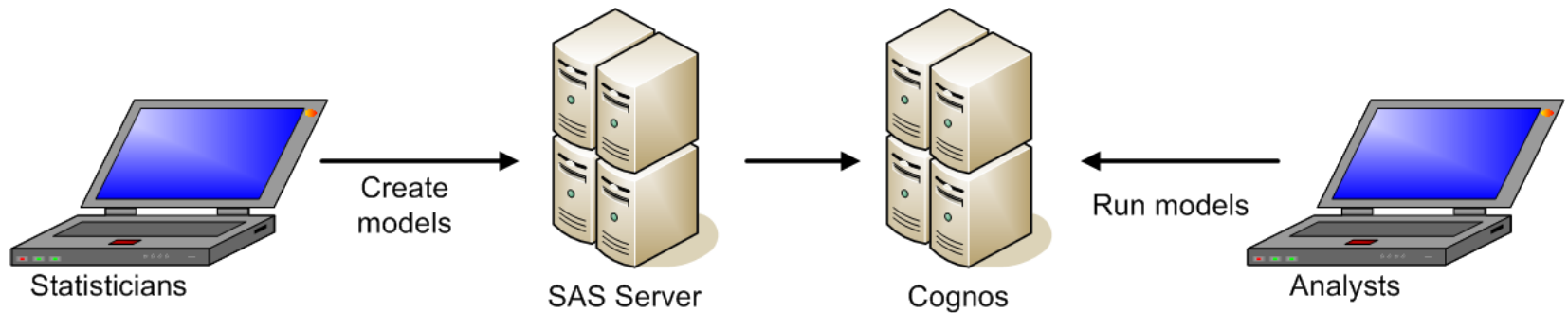
- History, Goals
- Concepts, Technologies
- Implementation
- Examples



# Request

- Surface statistical models from Cognos
- Rich Web interface
- If possible, no additional license cost

→ scope seems too narrow



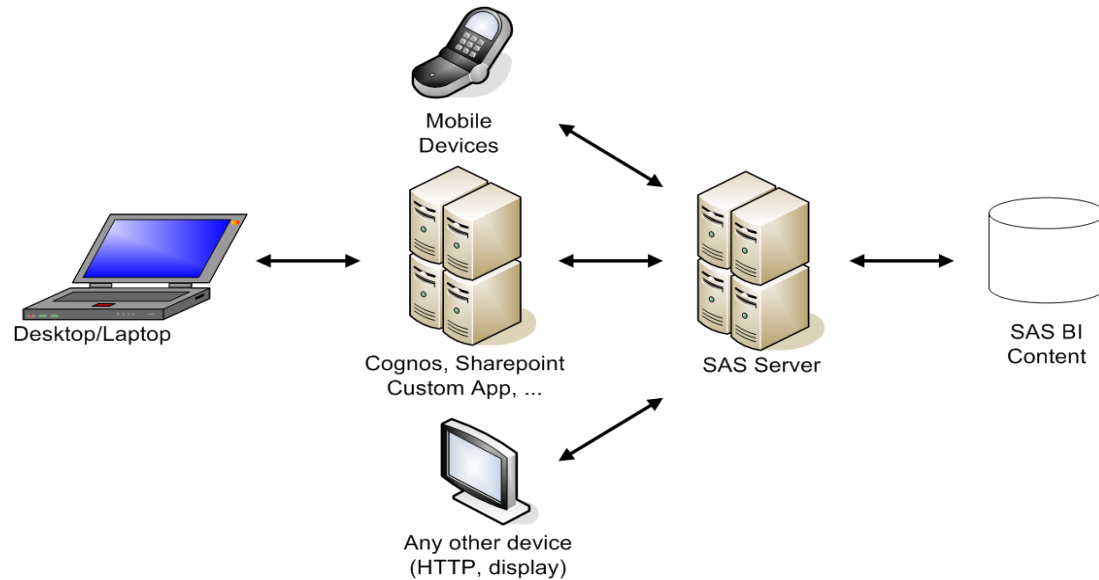
# Scope & Technologies

- No development on client side
  - Allows for development if desired
- Work from any device using a simple protocol
- Query and display all SAS BI content
  
- SAS, Metadata, Stored Processes
- Web Services
- SOA
- ...



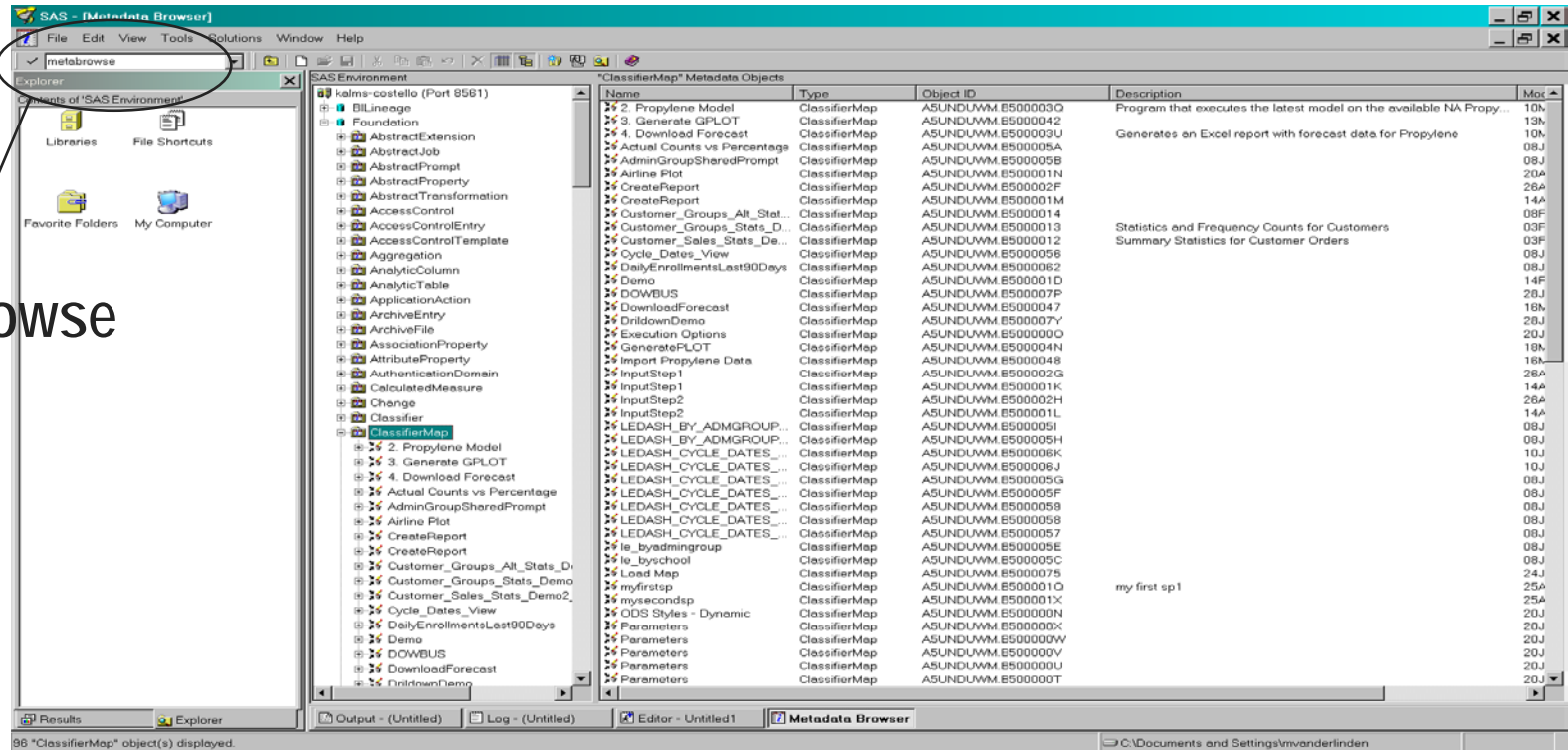
# Goal

- To syndicate the SAS BI Content
  - To any client (Cognos, custom application, mobile device)
  - Only requires SAS Integration Technologies
  - Provides interface for zero-footprint clients



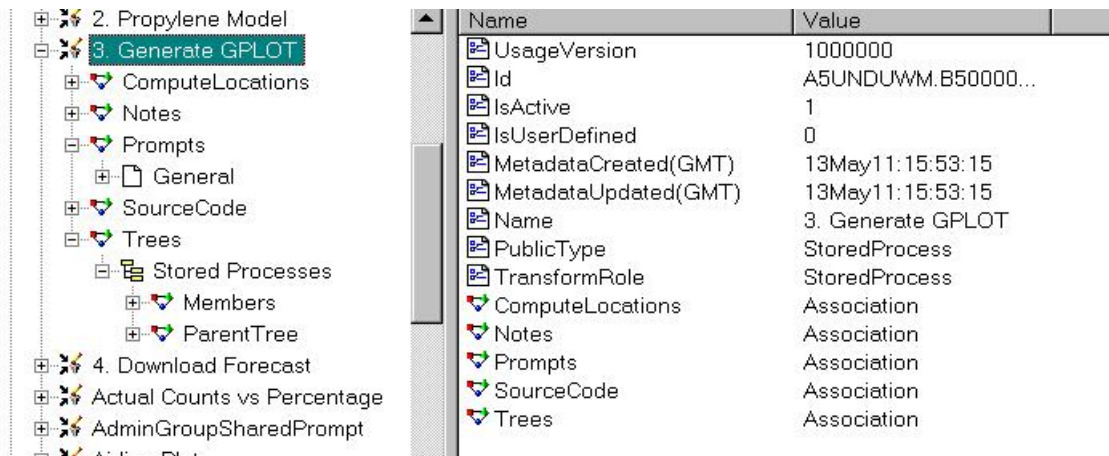
# SAS Metadata

- Metadata definitions for Libraries, Tables, Permissions, Servers, Information Maps, Stored Processes, ...



# SAS Metadata

List of Stored Processes → retrieve ClassifierMap objects which have a public type of “StoredProcess”.



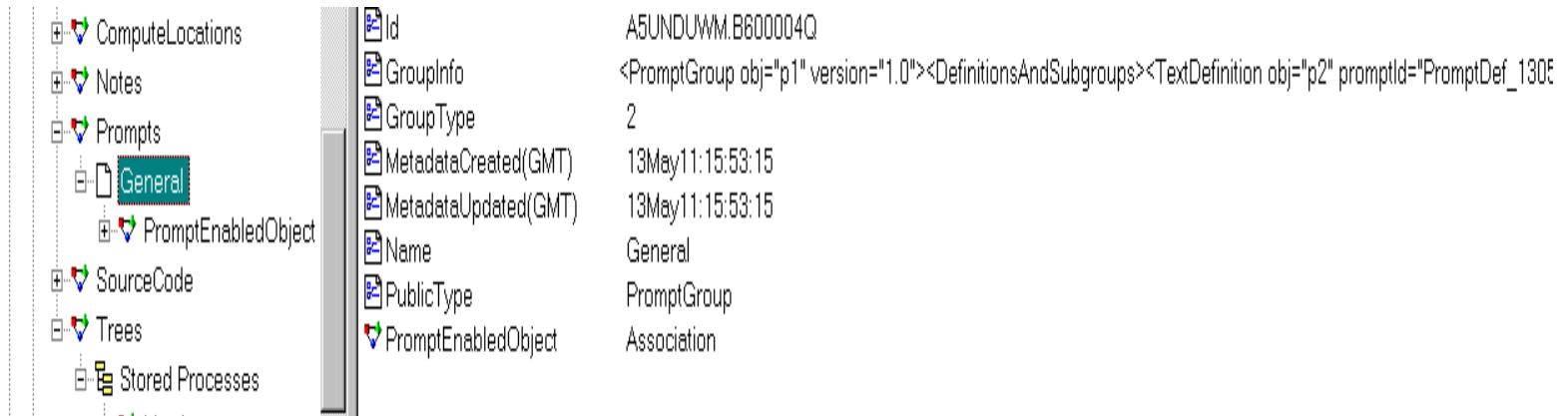
Name	Value
UsageVersion	1000000
Id	A5UNDUWM.B50000...
IsActive	1
IsUserDefined	0
MetadataCreated(GMT)	13May11:15:53:15
MetadataUpdated(GMT)	13May11:15:53:15
Name	3. Generate GPLOT
PublicType	StoredProcess
TransformRole	StoredProcess
ComputeLocations	Association
Notes	Association
Prompts	Association
SourceCode	Association
Trees	Association

The different Associations objects allow us to retrieve additional information about a specific Stored Process.



# SAS Metadata

Retrieve the prompts defined for this Stored Process, query the Prompts Association object which conveniently returns XML with all prompts definitions.



The screenshot shows the SAS Metadata browser interface. On the left, a tree view displays the hierarchy: ComputeLocations, Notes, Prompts (with 'General' selected), PromptEnabledObject, SourceCode, Trees, and Stored Processes. The right pane shows the details for the selected 'PromptEnabledObject'.

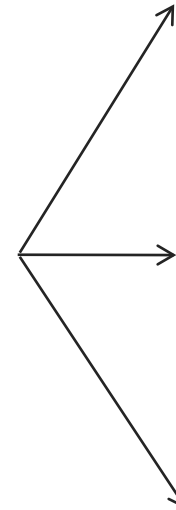
Id	A5UNDUWM.B600004Q
GroupInfo	<PromptGroup obj="p1" version="1.0"><DefinitionsAndSubgroups><TextDefinition obj="p2" promptId="PromptDef_130E
GroupType	2
MetadataCreated(GMT)	13May11:15:53:15
MetadataUpdated(GMT)	13May11:15:53:15
Name	General
PublicType	PromptGroup
PromptEnabledObject	Association



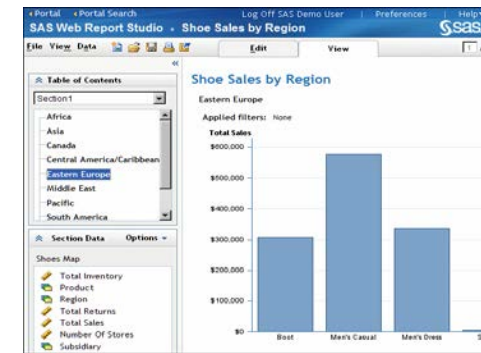
# SAS Foundation Services

Set of infrastructure and services to create applications that interface with the different components of a SAS Intelligence platform.

- managing IOM server connections
- locating and binding to services
- event notification
- repository federation and searching
- logging
- publishing
- authenticating and security management
- context and resource management
- stored processes retrieval and execution
- user session management



A screenshot of a web browser displaying the 'SAS Stored Processes' web application. The page has a blue header with the SAS logo and the text 'SAS Stored Processes'. Below the header, it says 'SAS Stored Process Web Application' and 'Welcome to the Version 9 SAS Stored Process Web Application. This application allows you to execute Stored Processes from a Web browser.' There are three links: 'Stored Process Samples', 'List Available Stored Processes', and 'SAS Integration Technologies Documentation - requires Internet access'.



# The REST Framework

- Software architecture for distributed hypermedia, a simpler-than-before client-server architecture based on the success of the World Wide Web.
- Looks a lot like SOA (Service Oriented Architecture), but sometimes called a Resource Oriented Architecture.
- A resource is any source of information that can be referenced with a global identifier.
- Global identifiers are highly visible and addressable.
- Resources are manipulated through a uniform interface, PUT, POST, GET and DELETE.
- Resources are separate from representations.
- Hypermedia as the engine of application state.



# The REST Framework

- Constraints to be RESTful:
  - Clear separation between the client and the server who communicate through a uniform interface.
  - Sessions between a client and a server are stateless.
  - Server responses must clearly define themselves as cacheable (or not) to ensure clients are always aware whether the information they receive is up-to-date.
  - Intermediaries between the client and the server provide scalability, performance and sometimes enforce security constraints. Clients must not be able to tell whether they are connected to an end server or an intermediary, this is referred to as a layered system.
- Following these constraints promotes simplicity, portability and performance which is just what we need to build a SAS BI Content Syndication solution.



# The REST Framework

- Identify resources and design a resource model for interfacing with them.

URI	Method	Representation	Options	Description
http://...	GET	HTML		Retrieve resource in HTML
http://...	GET	XML		Retrieve resource in XML
http://...	PUT	XML		Create a new resource

- Options for non-hierarchical elements
  - Commas when order matters, eg. /Earth/37.0,-95.2
  - Semicolons when order doesn't matter, eg. /blends/red;blue



# The REST Framework

- Resource Model Examples (GETs only)
- To retrieve a list of tables in HTML format  
<http://.../services/Tables/html>
- To retrieve the list of tables from folder XYZ or ZYX  
<http://.../services/Tables/html/xyz;zyx>
- To retrieve the prompts of a stored process in XML  
<http://.../services/Models/input/xml>



# Implementation

- Libraries, Tables and Stored Processes (XML & HTML)

URI	Method	Representation	Description
http://.../Libraries/html	GET	HTML	Retrieves collection of libraries in HTML format
http://.../ Libraries/xml	GET	XML	Retrieves collection of libraries in XML format
http://.../Libraries/objectID/html	GET	HTML	Retrieves a library identified by object ID in HTML format
http://.../Libraries/objectID/xml	GET	XML	Retrieves a library identified by object ID in XML format
http://.../tables/html	GET	HTML	Retrieves collection of tables in HTML format
http://.../ Tables/xml	GET	XML	Retrieves collection of tables in XML format
http://.../Tables/objectID/html	GET	HTML	Retrieves records from table identified by object ID in HTML format
http://.../Tables/objectID/xml	GET	XML	Retrieves records from table identified by object ID in XML format



# Implementation

- Libraries, Tables and Stored Processes (XML & HTML)

URI	Method	Representation	Description
http://.../Models/html	GET	HTML	Retrieves collection of models in HTML format
http://.../Models/xml	GET	XML	Retrieves collection of models in xml format
http://.../Models/objectID/input/html	GET	HTML	Retrieves the prompts of a model identified by object ID in HTML format
http://.../Models/objectID/input/xml	GET	XML	Retrieves the prompts of a model identified by object ID in xml format
http://.../Models/objectID/output	GET		Retrieves the output of a model identified by object ID



# Implementation

- Need Java code for:
  - Class definitions for tables, libraries and stored processes
  - SAS Metadata connection
  - Retrieve repository
  - Retrieve collections
  - Retrieve members
  - Query for table records
  - Retrieve stored process input (prompts)
  - Execute stored process
- XSL transformation to HTML



# Implementation

- Class definitions

*@XmlElement*

```
public class Table {  
    private String objectId;  
    private String name;  
    ...  
    public void setObjectId(String objectId) {this.objectId = objectId;}  
    public String getObjectId() {return this.objectId;}  
    ...  
    public Table(String objectId, String name, String description, String folder,  
        String library, String libraryLink, String dataLink) {  
        setObjectId(objectId); ...}}
```



# Implementation: ModelResource

```
@Path("xml")
```

```
@GET
```

```
public StreamingOutput getModelsXML() {
```

```
SASMetadataConnection smdc = new SASMetadataConnection();
```

```
    smdc.connectToServer();
```

```
    CMetadata repository = smdc.getRepositories();
```

```
    ArrayList<Model> models = smdc.getAllStoredProcesses(repository);
```

```
    models.addAll(modelMap.values());
```

```
    return getListXML(models);
```

```
}
```



# Implementation

```
@Path("html")
```

```
@GET
```

```
@Produces("text/html")
```

```
public StreamingOutput getModelsHTML() {
```

```
SASMetadataConnection smdc = new SASMetadataConnection();
```

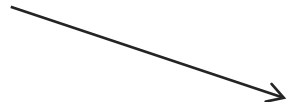
```
    smdc.connectToServer();
```

```
    CMetadata repository = smdc.getRepositories();
```

```
    List<Model> models = smdc.getAllStoredProcesses(repository);
```

```
    return getListHTML(models);
```

```
}
```

- 
- Call to getListXML
  - Transform XML into HTML using XSL



# Implementation

```
@Path("/{id}/input/xml")
```

```
@GET
```

```
@Produces("application/xml")
```

```
public StreamingOutput getModelInputXML(@PathParam("id") String cld) {  
    SASMetadataConnection smdc = new SASMetadataConnection();  
    smdc.connectToServer();  
    CMetadata repository = smdc.getRepositories();  
    Model model = smdc.getStoredProcessInfo(repository, cld);  
    return getInputXML(model);  
}
```

ID is metadata unique object ID



# Implementation

```
@Path("/{id}/input/html")
```

```
@GET
```

```
@Produces("text/html")
```

```
public StreamingOutput getModelInputHTML(@PathParam("id") String cld) {
```

```
    SASMetadataConnection smdc = new SASMetadataConnection();
```

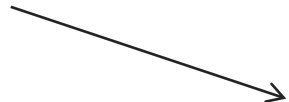
```
    smdc.connectToServer();
```

```
    CMetadata repository = smdc.getRepositories();
```

```
    Model model = smdc.getStoredProcessInfo(repository, cld);
```

```
    return getInputHTML(model);
```

```
}
```

- 
- Call to getInputXML
  - Transform XML into HTML using XSL



# Implementation

```
@Path("/{id}/output")
```

```
@GET
```

```
@Produces("text/html")
```

```
public StreamingOutput getModelOutputHTML(@PathParam("id") String cld,  
    @Context UriInfo info) {
```

```
    SASMetadataConnection smdc = new SASMetadataConnection();
```

```
    smdc.connectToServer();
```

```
    CMetadata repository = smdc.getRepositories();
```

```
    Model model = smdc.getStoredProcessInfo(repository, cld);
```

```
    return getOutput(model);
```

```
}
```



Executes stored process and streams back results



# Implementation: XSL Transformation

```
<xsl:output method="html" />
<xsl:param name="objid" />
<xsl:template match="/">
  <u>html>
  <form id="myForm" action="..." method='GET'>
  <input type='hidden' name="sp" value="{$objid}" />
  <table>
  <xsl:for-each select="PromptGroup/DefinitionsAndSubgroups/TextDefinition">
  <tr><td><xsl:value-of select="Label/Text" /></td><td>
  <u>xsl:choose>
  <xsl:when test="ValueProvider/StaticValueProvider/Values">
  <input name="{@name}" id="{@name}" type="hidden" value="" />
  ...
```



# Example: Custom Web Application

- Rich Web Interface
- Select a model
- Modify model parameters
- Display output

HTML, CSS

Javascript Frameworks

AJAX

GET .../models/html

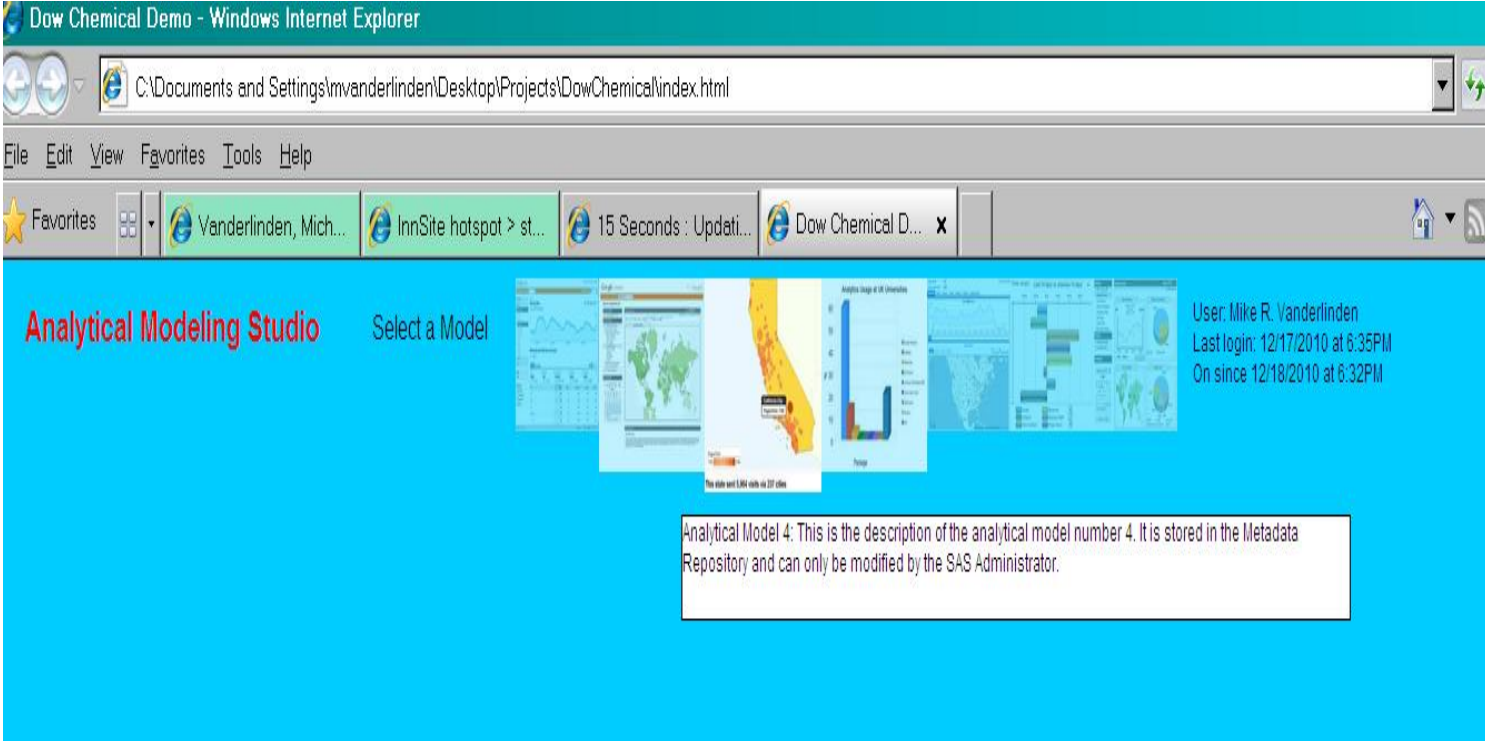
GET .../models/id/input/html

GET .../models/id/output



# Custom Web Application: GET models

Calls: <http://.../services/models/xml>



The screenshot shows a Windows Internet Explorer browser window titled "Dow Chemical Demo - Windows Internet Explorer". The address bar displays the URL "C:\Documents and Settings\mvanderlinden\Desktop\Projects\DowChemical\index.html". The browser's menu bar includes "File", "Edit", "View", "Favorites", "Tools", and "Help". The Favorites bar shows several bookmarks, including "Vanderlinden, Mich...", "InnSite hotspot > st...", "15 Seconds : Updati...", and "Dow Chemical D...".

The main content area has a blue background and features the text "Analytical Modeling Studio" in red. To its right is the text "Select a Model". Below this text are several small thumbnail images representing different analytical models, including a world map, a map of Florida, and a bar chart. On the right side of the main content area, there is a user information box that reads: "User: Mike R. Vanderlinden", "Last login: 12/17/2010 at 6:35PM", and "On since 12/18/2010 at 6:32PM".

At the bottom of the main content area, there is a white text box with a black border containing the text: "Analytical Model 4: This is the description of the analytical model number 4. It is stored in the Metadata Repository and can only be modified by the SAS Administrator."



# Custom Web Application: GET model input

Calls: <http://.../services/models/xxx/input/xml>

Transforms xml into HTML form with XSL

The screenshot shows a Windows Internet Explorer browser window displaying a web application titled "Analytical Modeling Studio". The browser's address bar shows the URL "C:\Documents and Settings\mvanderlinden\Desktop\Projects\DowChemical\index.html". The application interface has a blue header with the text "Analytical Modeling Studio" and "Select a Model". Below the header, there is a window titled "Analytical Model 2" with tabs for "Parameters" and "Results". The "Parameters" tab is active, showing a form with the following fields:

- Transform: Log10
- Analysis Variable: Salary
- Input Variables: hits runs rbi walks years f
- Analyses: Diagnostics, RStudent, CooksD, Residuals, DFFITS
- Format: PDF
- Style: (empty text box)
- Execute: (button)

# Custom Web Application: GET model output

Calls: <http://.../services/models/xxx/output>

**Analytical Modeling Studio** Select a Model

**Analytical Model 2 Parameters Results**

The REG Procedure  
Model: MODEL1  
Dependent Variable: logSalary Log10(Salary)

Number of Observations Read	322
Number of Observations Used	263
Number of Observations with Missing Values	59

Analysis of Variance

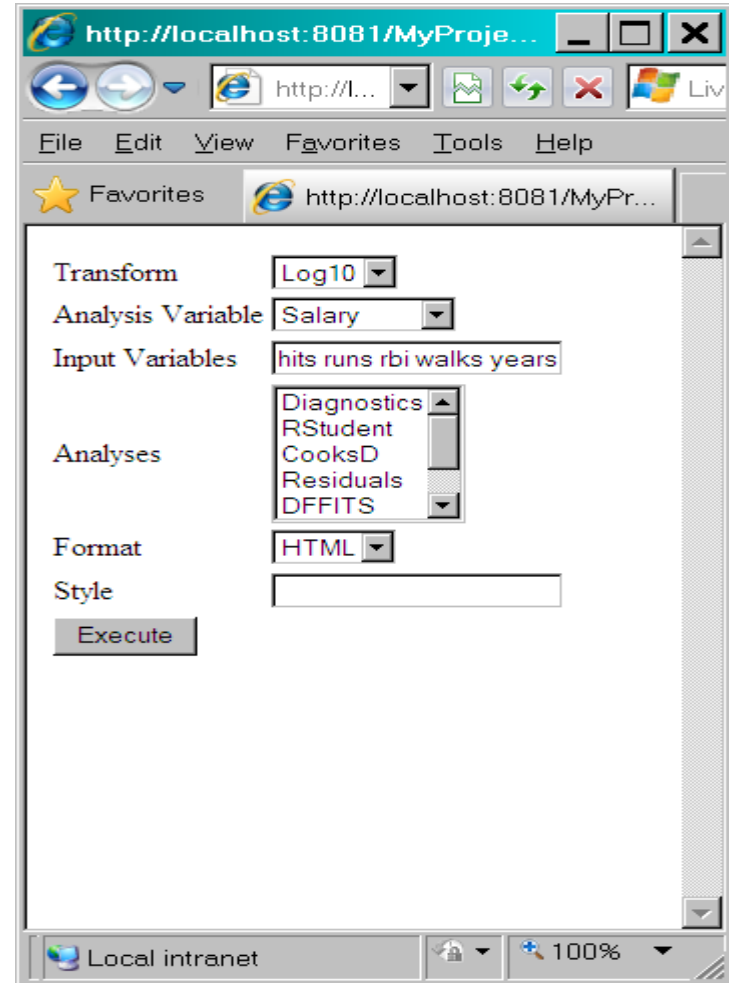
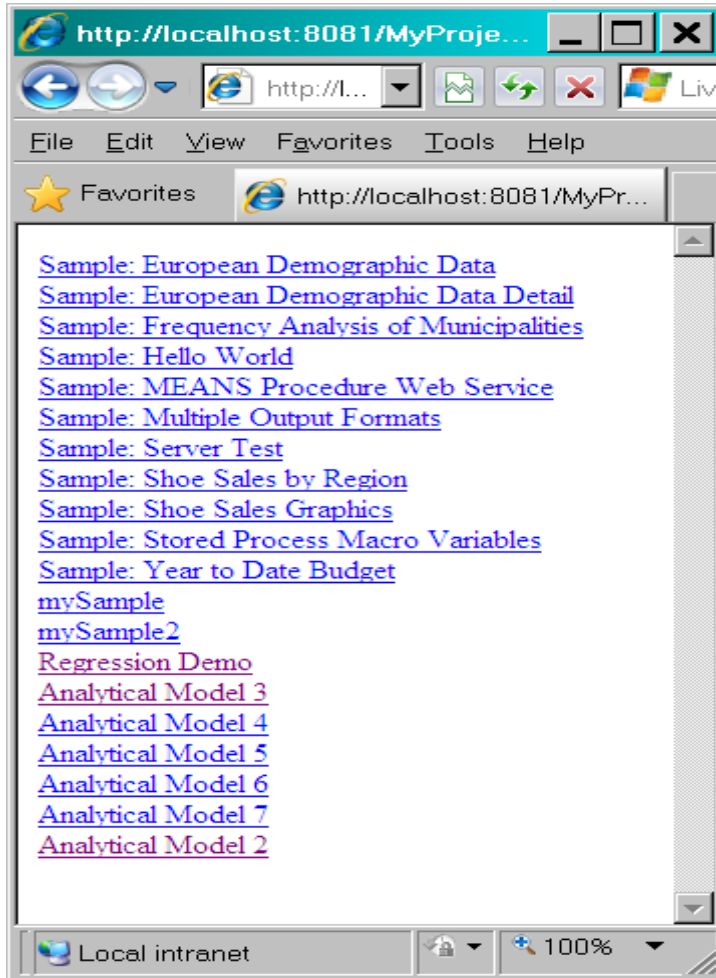
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	6	19.88079	3.31346	44.24	<.0001
Error	256	19.17344	0.07490		
Corrected Total	262	39.05423			

Root MSE 0.27367 R-Square 0.5091  
Dependent Mean 2.57424 Adj R-Sq 0.4975  
Coeff Var 10.63115

Parameter Estimates

Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr >  t
Intercept	Intercept	1	1.91903	0.06176	31.07	<.0001
hits	Hits	1	0.00209	0.00099331	2.10	0.0368
runs	Runs	1	0.00049780	0.00189	0.26	0.7928
rbt	Runs Batted In	1	0.00053240	0.00112	0.47	0.6355
walks	Walks	1	0.00246	0.00113	2.17	0.0308
years	Years in the Major Leagues	1	0.02224	0.00901	2.47	0.0142
hitssc	Career Hits	1	0.00015512	0.00006847	2.27	0.0243

# Example: IE... or Sharepoint, Cognos...



# Example: IE... or Sharepoint, Cognos...

SAS Output - Windows Internet Explorer

http://localhost:8081/MyProject/StoredProcessDriver?sp=A5WDB6K3.B500001J&dependent=salary&outgraphs=1

File Edit View Favorites Tools Help

SAS Output

**The REG Procedure**  
**Model: MODEL1**  
**Dependent Variable: logSalary Log10(salary)**

Number of Observations Read	322
Number of Observations Used	263
Number of Observations with Missing Values	59

**Analysis of Variance**

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	6	19.88079	3.31346	44.24	<.0001
Error	256	19.17344	0.07490		
Corrected Total	262	39.05423			

Root MSE	0.27367	R-Square	0.5091
Dependent Mean	2.57424	Adj R-Sq	0.4975
Coeff Var	10.63115		

**Parameter Estimates**

Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr >  t
Intercept	Intercept	1	1.91903	0.06176	31.07	<.0001

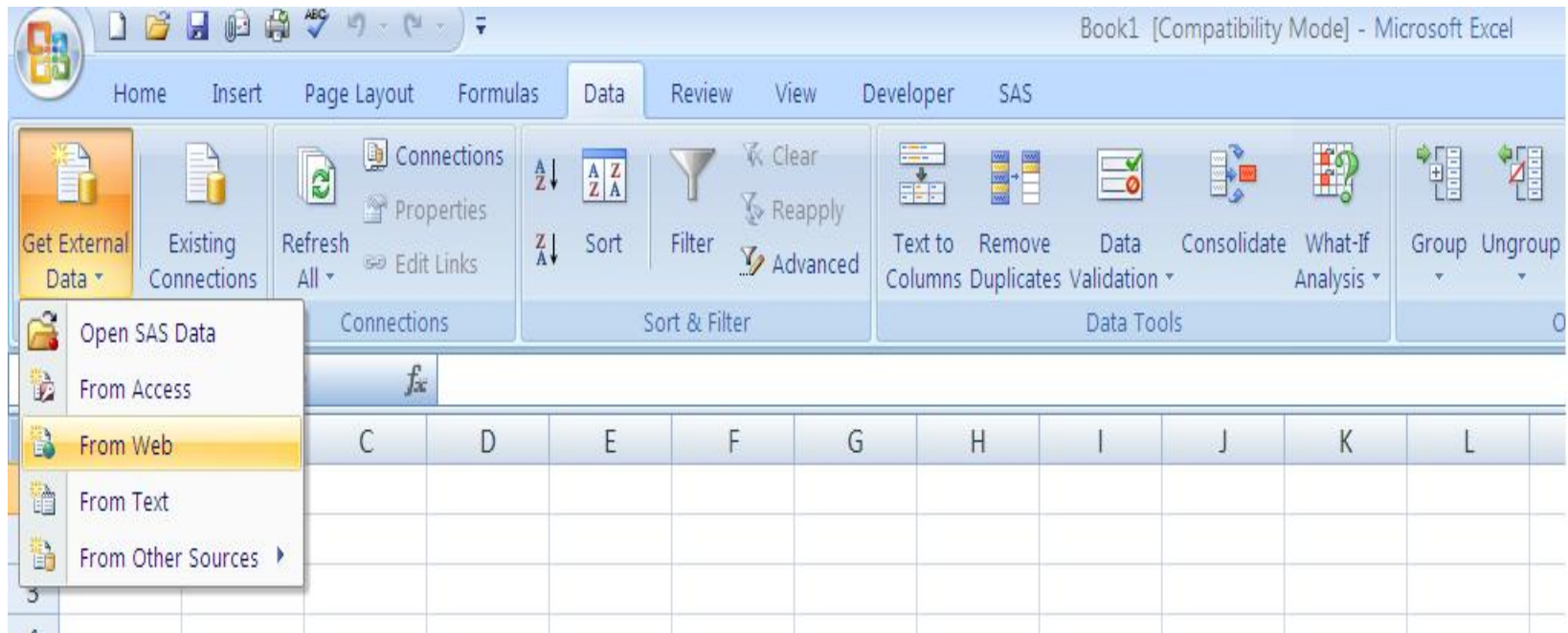
Done

Local intranet 100%

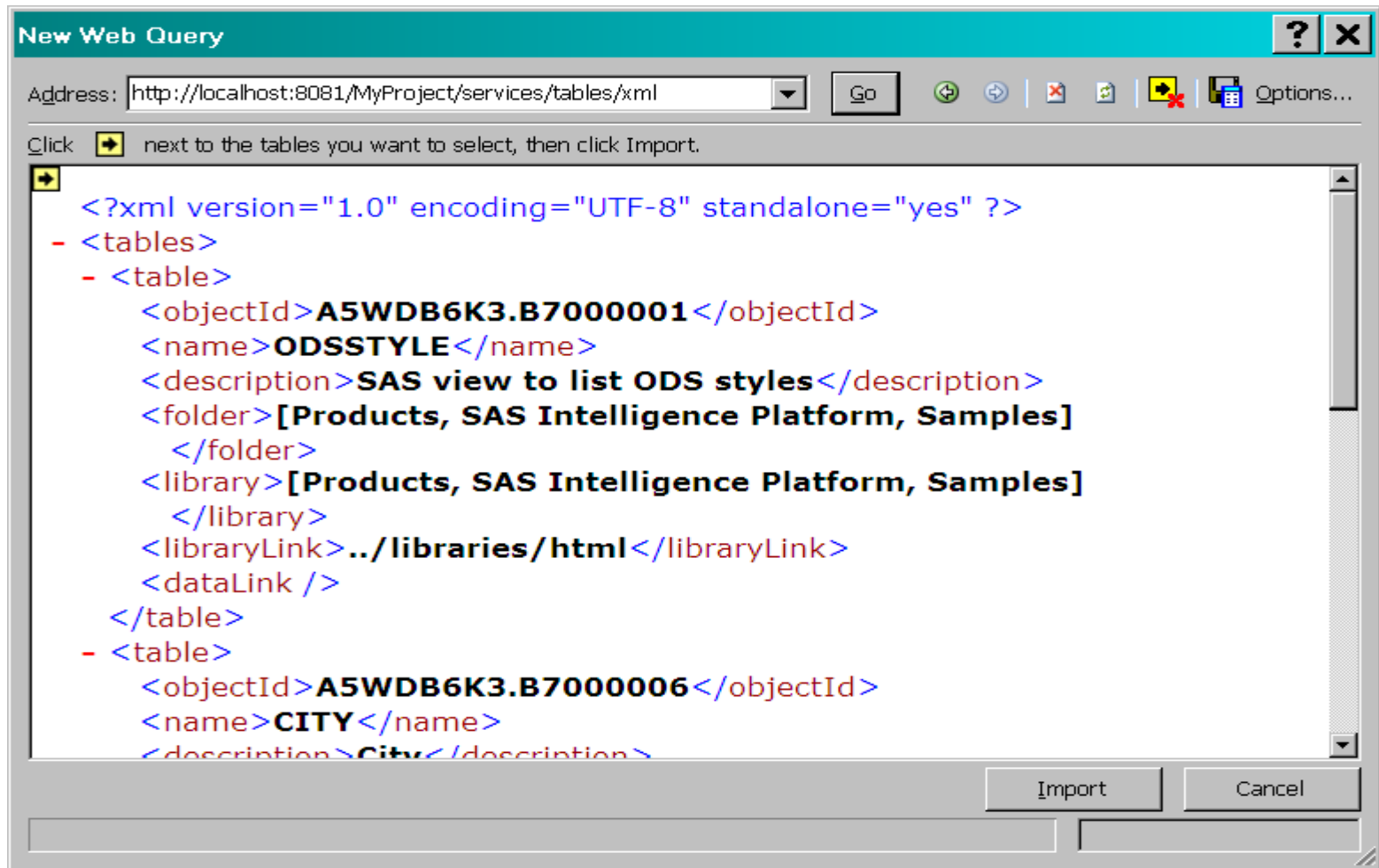


# Example: MS-Excel

Excel allows the retrieval of Web data



# Example: MS-Excel



# Example: MS-Excel

The screenshot shows a Microsoft Excel window with a table named 'Table1'. The table has the following data:

objectid	name	description	folder	library	libraryLink	dataLink
A5WDB6K3.B7000001	ODSSTYLE	SAS view to list ODS styles	[Products, SAS Intelligence Platform, Samples]	[Products, SAS Intelligence Platform, Samples]	../libraries/html	
A5WDB6K3.B7000006	CITY	City	[Shared Data, Orion Detail]	[Shared Data, Orion Detail]	../libraries/html	
A5WDB6K3.B7000007	CONTINENT	Continent	[Shared Data, Orion Detail]	[Shared Data, Orion Detail]	../libraries/html	
A5WDB6K3.B7000008	COUNTRY	Country	[Shared Data, Orion Detail]	[Shared Data, Orion Detail]	../libraries/html	



# Example: ... or mobile devices

AT&T 9:22 AM 55%

Untitled

192.168.1.101:8081/M... Google

[Sample: European Demographic Data](#)  
[Sample: European Demographic Data Detail](#)  
[Sample: Frequency Analysis of Municipalities](#)  
[Sample: Hello World](#)  
[Sample: MEANS Procedure Web Service](#)  
[Sample: Multiple Output Formats](#)  
[Sample: Server Test](#)  
[Sample: Shoe Sales by Region](#)  
[Sample: Shoe Sales Graphics](#)  
[Sample: Stored Process Macro Variables](#)  
[Sample: Year to Date Budget](#)  
[mySample](#)  
[mySample2](#)  
[Regression Demo](#)  
[Analytical Model 3](#)  
[Analytical Model 4](#)  
[Analytical Model 5](#)  
[Analytical Model 6](#)  
[Analytical Model 7](#)  
[Analytical Model 7](#)

Navigation icons: back, forward, home, search, refresh

AT&T 9:22 AM 55%

192.168.1.101:8081/M... Google

Transform   
 Analysis Variable   
 Input Variables   
 Analyses   
 Format   
 Style

Navigation icons: back, forward, home, search, refresh

AT&T 9:26 AM 54%

The REG Procedure  
Model: MODEL1  
Dependent Variable: logSalary Log10(salary)

Number of Observations Read	322
Number of Observations Used	263
Number of Observations with Missing Values	59

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	6	32.88076	3.31346	44.24	<.0001
Error	256	38.17244	0.07490		
Corrected Total	262	39.05423			

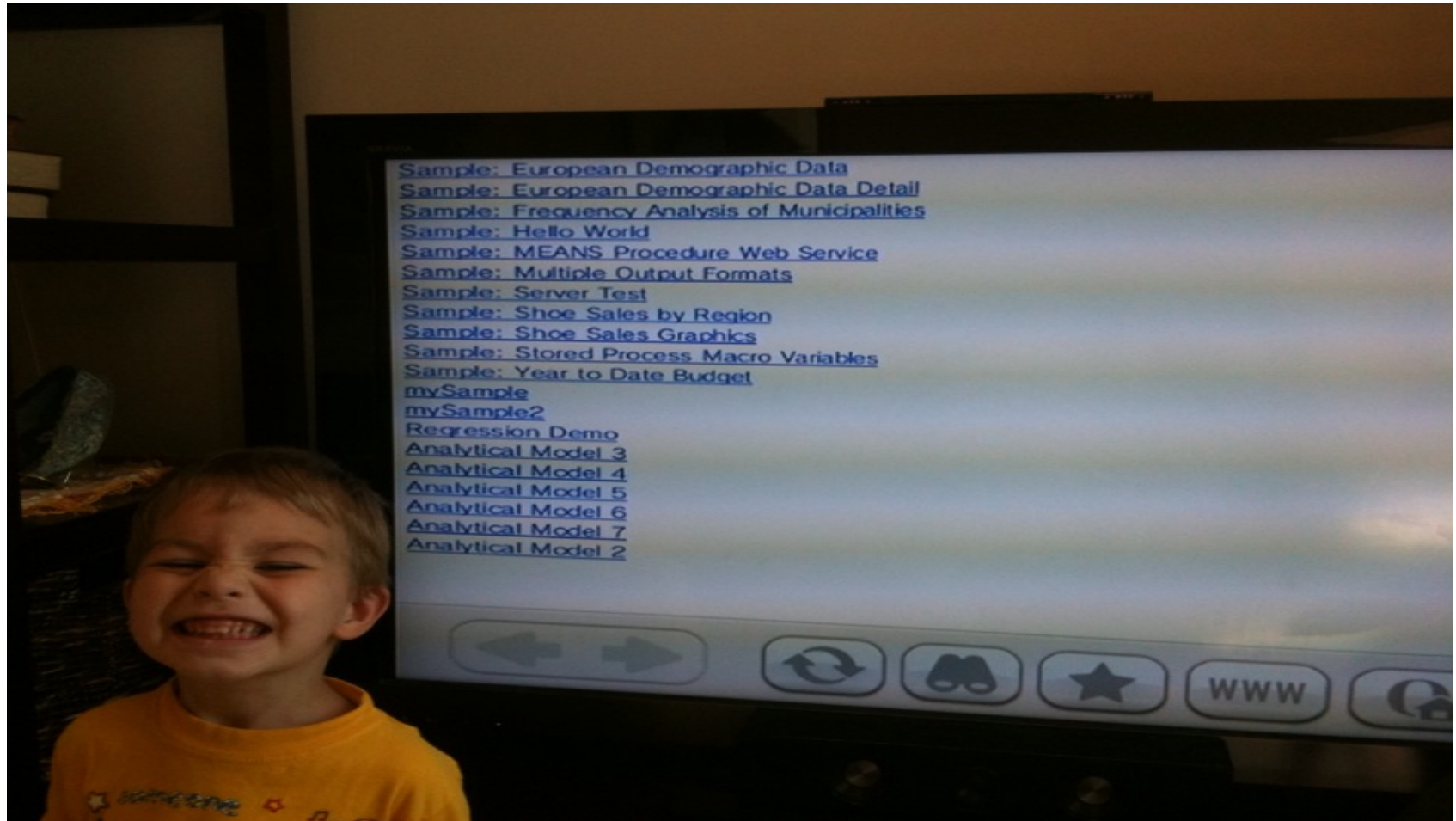
Root MSE	0.27367	R-Square	0.5091
Dependent Mean	2.57424	Adj R-Sq	0.4975
Coeff Var	10.63115		

Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr >  t
Intercept	Intercept	1	1.91903	0.06176	31.07	<.0001
hits	Hits	1	0.09209	0.00092331	2.10	0.0369
runs	Runs	1	0.00049780	0.00189	0.26	0.7928
rbi	Runs Batted In	1	0.00053240	0.00112	0.47	0.6395
walks	Walks	1	0.09246	0.00113	2.17	0.0338
years	Years in the Major Leagues	1	0.02224	0.00901	2.47	0.0142
hitsc	Career Hits	1	0.00015512	0.00005847	2.27	0.0243

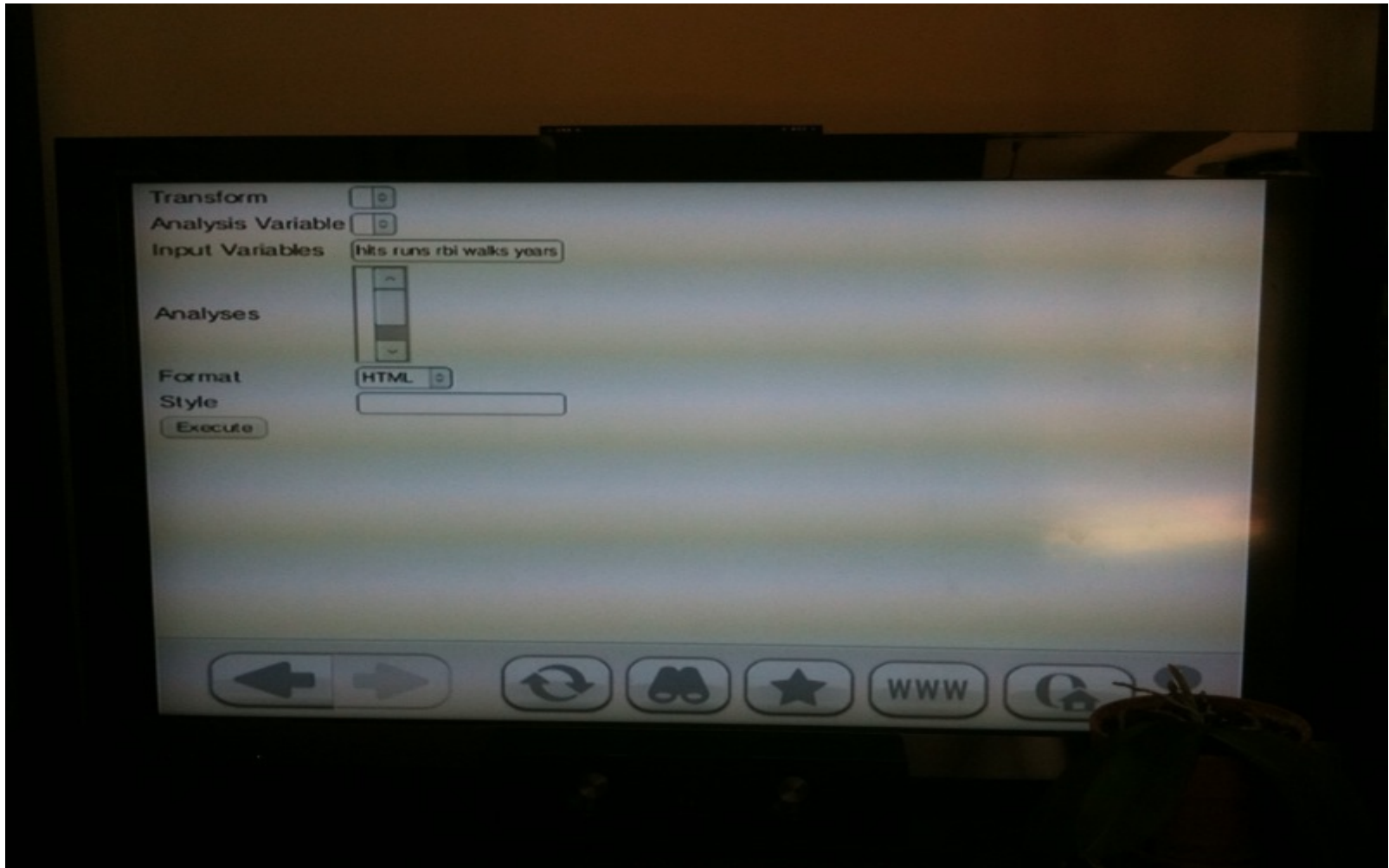
Navigation icons: back, forward, home, search, refresh



Example: ... or any device → The Wii



Example: ... or any device → The Wii



Example: ... or any device → The Wii

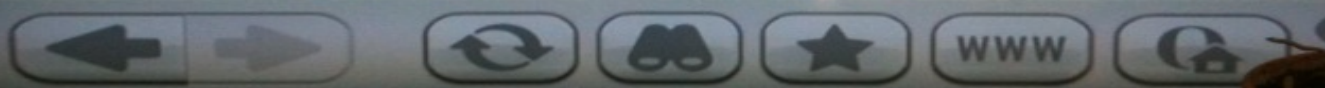
The REG Procedure  
Model: MODEL1  
Dependent Variable: logSalary Log10(salary)

Number of Observations Read	322
Number of Observations Used	263
Number of Observations with Missing Values	59

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	6	19.88079	3.31346	44.24	<.0001
Error	256	19.17344	0.07490		
Corrected Total	262	39.05423			

Root MSE 0.27367 R-Square 0.5091



# Time to take a REST...

## Thank you!!!

Questions, feedback &  
comments are welcome.

[mike.vanderlinden@experis.com](mailto:mike.vanderlinden@experis.com)

