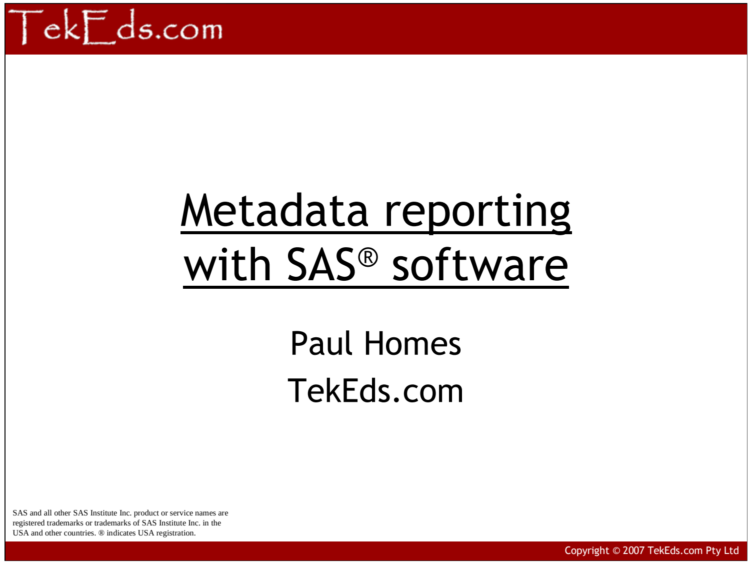


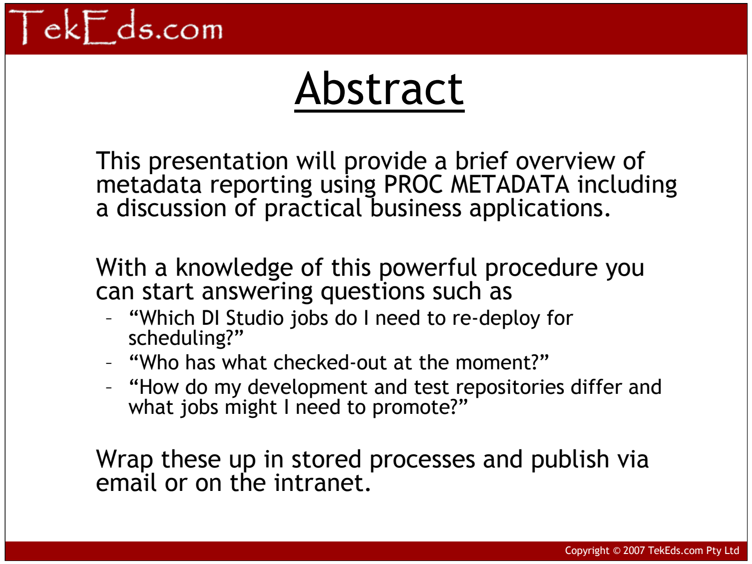
TekEds.com

Metadata reporting with SAS® software

Paul Homes

TekEds.com

This presentation will provide a brief overview of metadata reporting using Base SAS and PROC METADATA. It is a technical presentation involving some SAS code samples. I will avoid going into too much detail due to time constraints but the presentation and the sample code will be available for you to review after this presentation.



TekEds.com

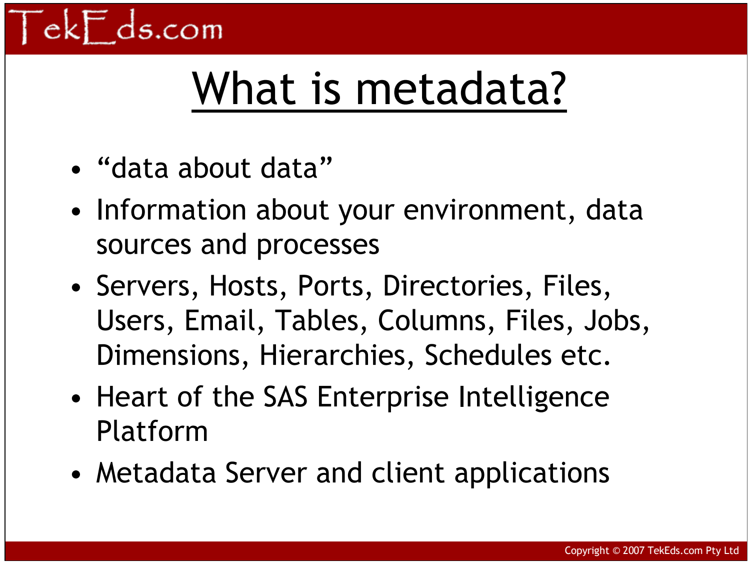
Abstract

This presentation will provide a brief overview of metadata reporting using PROC METADATA including a discussion of practical business applications.

With a knowledge of this powerful procedure you can start answering questions such as

- “Which DI Studio jobs do I need to re-deploy for scheduling?”
- “Who has what checked-out at the moment?”
- “How do my development and test repositories differ and what jobs might I need to promote?”

Wrap these up in stored processes and publish via email or on the intranet.

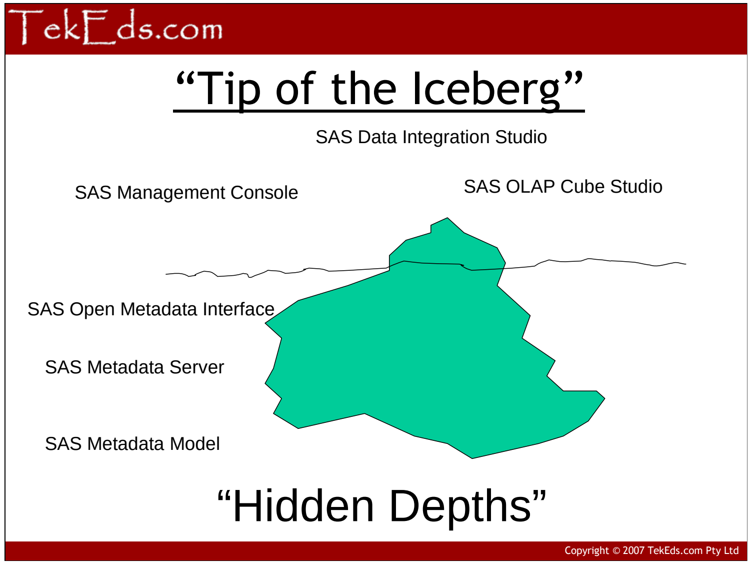


TekEds.com

What is metadata?

- “data about data”
- Information about your environment, data sources and processes
- Servers, Hosts, Ports, Directories, Files, Users, Email, Tables, Columns, Files, Jobs, Dimensions, Hierarchies, Schedules etc.
- Heart of the SAS Enterprise Intelligence Platform
- Metadata Server and client applications

A very quick overview of metadata. Metadata is often described as “data about data”. Other than the specific data that you use for your day to day business there is a great deal of other data (metadata) that describes your environment such as servers, configuration details, people, roles, data and navigation structures, data transformation processes, schedules etc. This information can be maintained and exploited by software applications. Metadata is are the very heart of the SAS enterprise intelligence platform. The metadata server is a dedicated component that all of the other SAS server and client application use to store and find the metadata that they need.

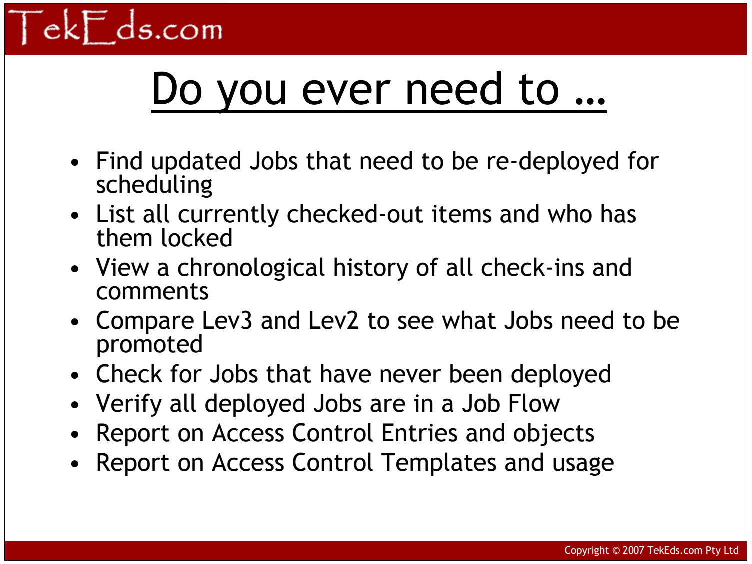


TekEds.com

“Tip of the Iceberg”

“Hidden Depths”

Client applications such as SAS Data Integration Studio, SAS OLAP Cube Studio and the SAS Management Console present specific perspectives of SAS metadata relevant to the task at hand. This is the still only the “tip of the iceberg” The SAS Metadata Model is rich and very detailed. Using the SAS Open Metadata Interface we can access these “hidden depths” and fulfil additional requirements. The SAS metadata model is accessible from Java, .NET and even Base SAS. The rest of this presentation will concentrate on metadata reporting from Base SAS via PROC METADATA.



TekEds.com

Do you ever need to …

- Find updated Jobs that need to be re-deployed for scheduling
- List all currently checked-out items and who has them locked
- View a chronological history of all check-ins and comments
- Compare Lev3 and Lev2 to see what Jobs need to be promoted
- Check for Jobs that have never been deployed
- Verify all deployed Jobs are in a Job Flow
- Report on Access Control Entries and objects
- Report on Access Control Templates and usage

These are examples of useful business applications of metadata reporting with PROC METADATA.

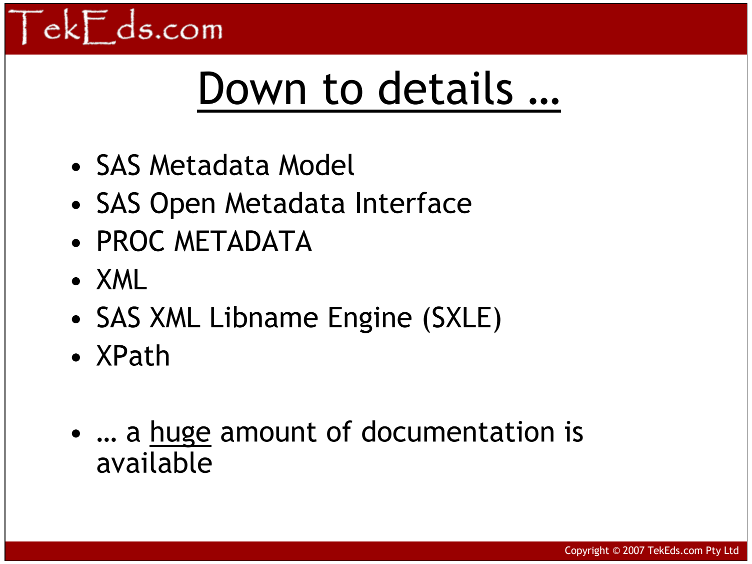
Re-deploying all jobs can be a lengthy process, so it can be more productive to only re-deploy those that need it.

Sometimes you want to make sure nothing is currently checked out.

When promoting metadata from development to test it helps to view a list of check-ins and who to ask questions of.

Comparing development and test repositories can streamline metadata promotion.

Metadata reporting can also be used for audit checks.



TekEds.com

Down to details …

- SAS Metadata Model
- SAS Open Metadata Interface
- PROC METADATA
- XML
- SAS XML Libname Engine (SXLE)
- XPath
- … a huge amount of documentation is available

An understanding of the SAS Metadata Model will help you find the underlying objects and follow the associations to related objects.

PROC METADATA is used to send XML requests to the SAS Metadata Server and received the XML response

The SAS XML Libname Engine (SXLE) will help you turn the XML response into SAS tables – you then have access to all the power of SAS

A knowledge of XML will help you construct valid XML requests and SXLE maps.

SAS Metadata Model

- SAS OnlineDoc®: SAS Open Metadata Architecture, SAS Open Metadata Interface: Reference, SAS Metadata Model
- Object Types: e.g.
  - Person, IdentityGroup
  - SASLibrary, PhysicalTable, Column
  - Job, JFJob
- Object Attributes: e.g.
  - Id, Name, Desc, Libref,
- Object Associations: e.g.
  - EmailAddresses, IdentityGroups, Columns,

The SAS Metadata Model documents the available objects and their possible attributes and associations.

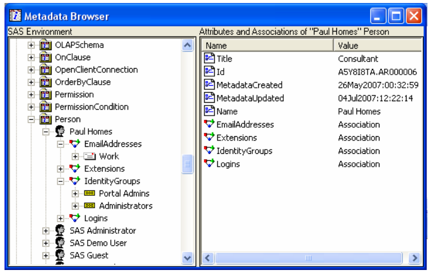
There is a lot of documentation available in the SAS OnlineDoc. Follow the path: SAS Open Metadata Architecture, SAS Open Metadata Interface: Reference, SAS Metadata Model

Objects often represent obvious things you are familiar with: people, security groups, SAS libraries, tables and columns, jobs and jobs flows. Some are not so obvious like TransformationActivity and SyncStep and you will probably have to look them up.

Objects are linked together by associations: for example a person (Person) is associated with their security groups (IdentityGroup) via the IdentityGroups association.

Metadata Browser

- An essential tool for metadata exploration …

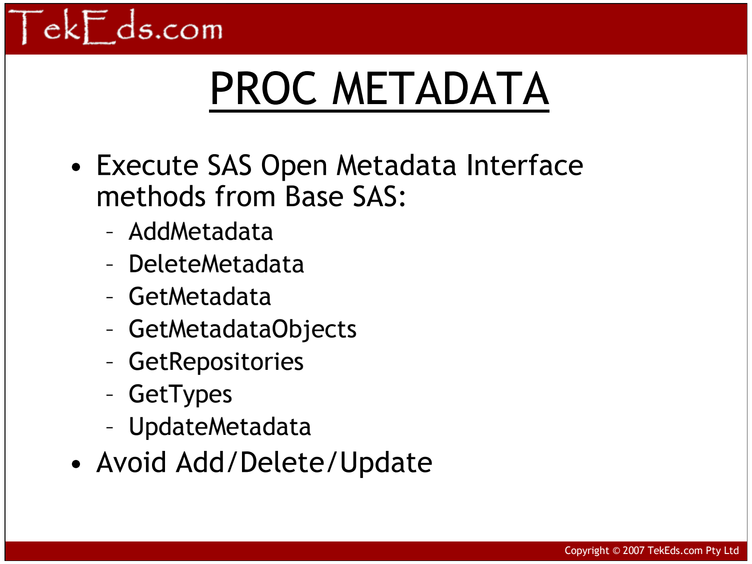


Attributes and Associations of "Paul Homes" Person

Name	Value
Title	Consultant
Id	A5Y8I8TA.AR000006
MetadataCreated	26May2007:00:32:59
MetadataUpdated	04Jul2007:12:22:14
Name	Paul Homes
EmailAddresses	Association
Extensions	Association
IdentityGroups	Association
Logins	Association

One of the essential tools for exploring SAS metadata and developing PROC METADATA queries is a metadata browser available from the desktop SAS interface. It can be accessed using the *metabrowse* command or from the menu using *Solutions > Accessories > Metadata Browser.*

This screenshot demonstrates the display of object types, objects, object attributes and object associations



TekEds.com

PROC METADATA

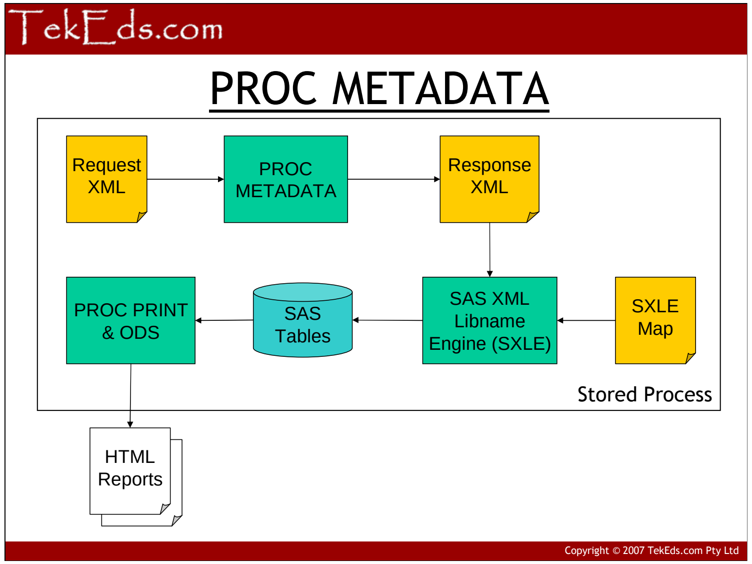
- Execute SAS Open Metadata Interface methods from Base SAS:
  - AddMetadata
  - DeleteMetadata
  - GetMetadata
  - GetMetadataObjects
  - GetRepositories
  - GetTypes
  - UpdateMetadata
- Avoid Add/Delete/Update

SAS OnlineDoc paths:

For PROC METADATA: SAS Open Metadata Architecture > SAS Open Metadata Interface: Reference > SAS Language Metadata Interfaces > Procedures > METADATA Procedure

For available methods: SAS Open Metadata Architecture > SAS Open Metadata Interface: Reference > Method Classes

The Get* methods are read-only and safe. Unless you really know what you are doing avoid the add, delete and update methods – they are dangerous and may get you in trouble ☺ If you absolutely have to use them, test thoroughly on a private metadata server and repository.



TekEds.com

PROC METADATA

Typical use of PROC METADATA query involves:

•provide metadata server connection details: host, port, credentials, repository

•construct an XML request containing a method call

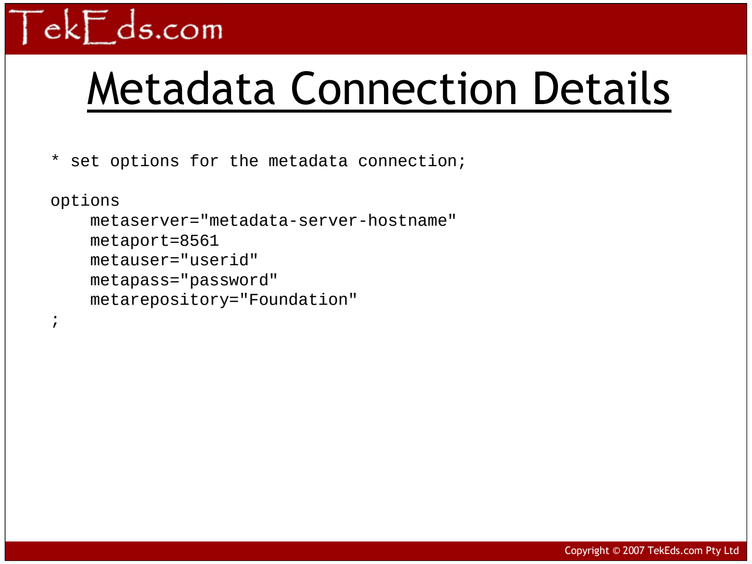
•execute proc metadata using the request XML and get the XML response containing the metadata

•review the XML response

•construct a SAS XML Libname Engine (SXLE) map document that can converts the XML response into SAS tables

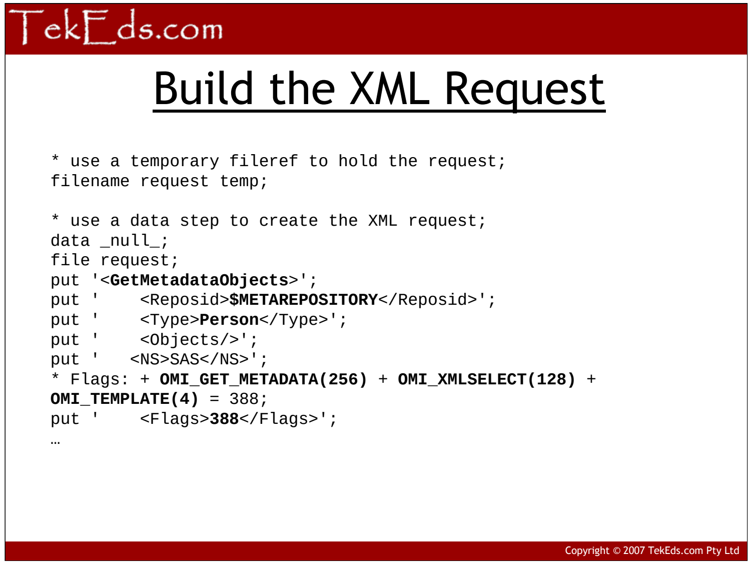
•generate HTML reports from SAS tables using SAS procedures and ODS as normal

•wrap the whole thing up in a stored process



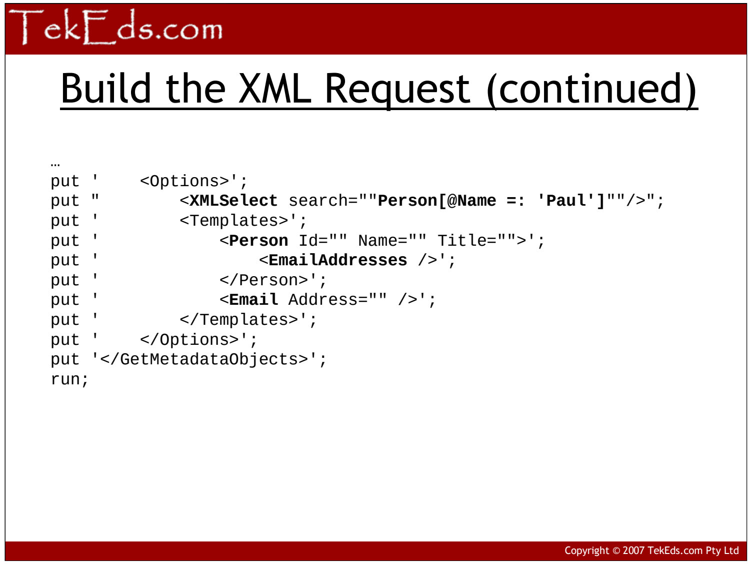
Metadata Connection Details

```
* set options for the metadata connection;

options
    metaserver="metadata-server-hostname"
    metaport=8561
    metauser="userid"
    metapass="password"
    metarepository="Foundation"
;
```



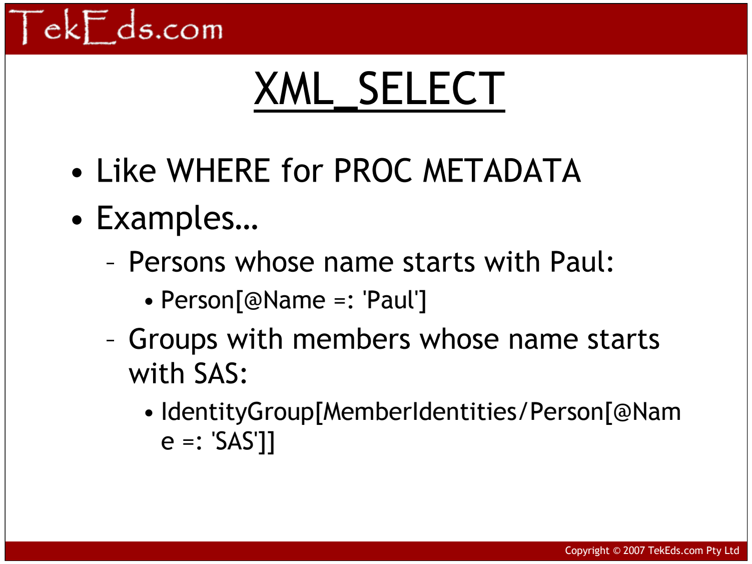
Build the XML Request

```
* use a temporary fileref to hold the request;
filename request temp;

* use a data step to create the XML request;
data _null_;
file request;
put '<GetMetadataObjects>';
put '    <Reposid>$METAREPOSITORY</Reposid>';
put '    <Type>Person</Type>';
put '    <Objects/>';
put '   <NS>SAS</NS>';
* Flags: + OMI_GET_METADATA(256) + OMI_XMLSELECT(128) +
OMI_TEMPLATE(4) = 388;
put '    <Flags>388</Flags>';
…
```



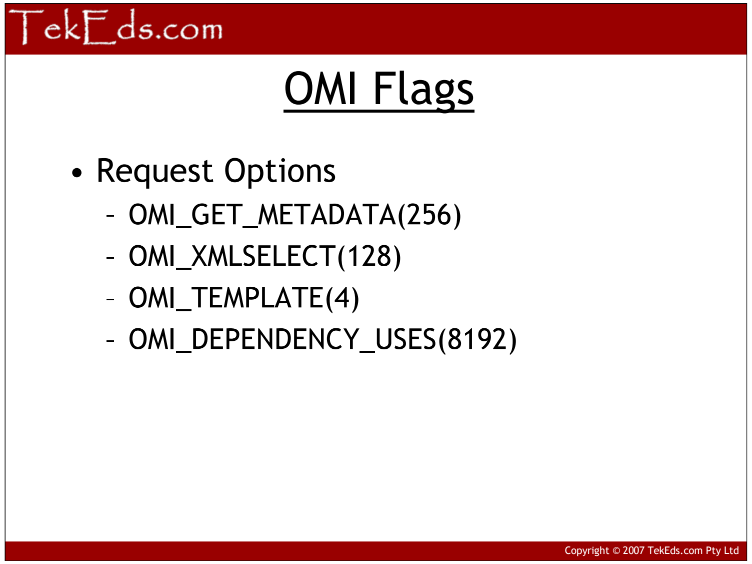
Build the XML Request (continued)

```
…
put '    <Options>';
put "        <XMLSelect search=""Person[@Name =: 'Paul']""/>";
put '        <Templates>';
put '            <Person Id="" Name="" Title="">';
put '                <EmailAddresses />';
put '            </Person>';
put '            <Email Address="" />';
put '        </Templates>';
put '    </Options>';
put '</GetMetadataObjects>';
run;
```



XML_SELECT

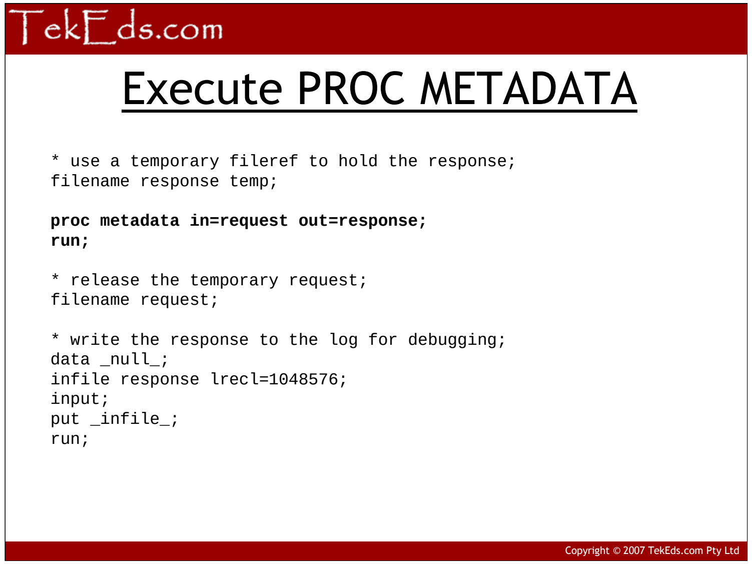
- Like WHERE for PROC METADATA
- Examples…
  - Persons whose name starts with Paul:
    - Person[@Name =: 'Paul']
  - Groups with members whose name starts with SAS:
    - IdentityGroup[MemberIdentities/Person[@Name =: 'SAS']]



TekEds.com

OMI Flags

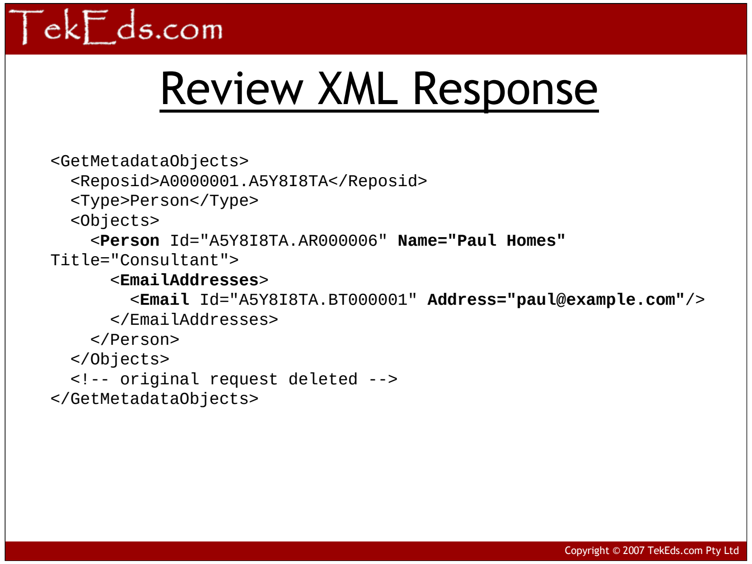
- Request Options
  - OMI_GET_METADATA(256)
  - OMI_XMLSELECT(128)
  - OMI_TEMPLATE(4)
  - OMI_DEPENDENCY_USES(8192)



TekEds.com

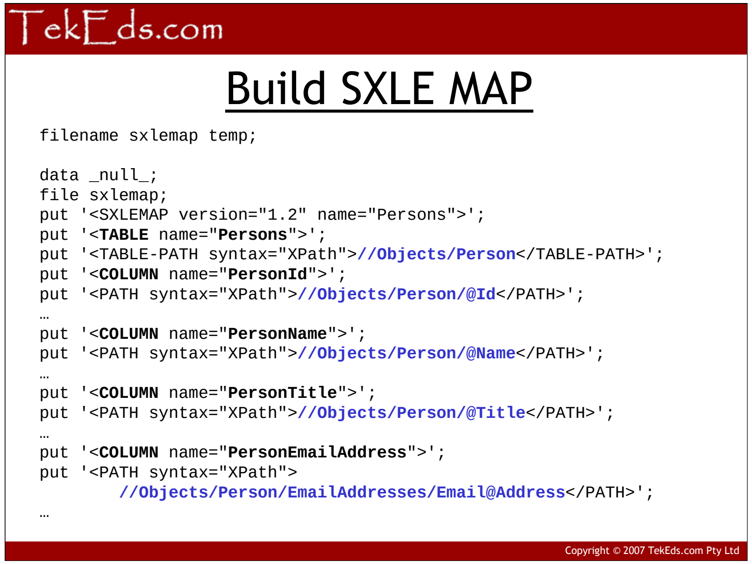
Execute PROC METADATA

```
* use a temporary fileref to hold the response;
filename response temp;

proc metadata in=request out=response;
run;

* release the temporary request;
filename request;

* write the response to the log for debugging;
data _null_;
infile response lrecl=1048576;
input;
put _infile_;
run;
```



Review XML Response

```
<GetMetadataObjects>
  <Reposid>A0000001.A5Y8I8TA</Reposid>
  <Type>Person</Type>
  <Objects>
    <Person Id="A5Y8I8TA.AR000006" Name="Paul Homes"
Title="Consultant">
      <EmailAddresses>
        <Email Id="A5Y8I8TA.BT000001" Address="paul@example.com"/>
      </EmailAddresses>
    </Person>
  </Objects>
  <!-- original request deleted -->
</GetMetadataObjects>
```



Build SXLE MAP

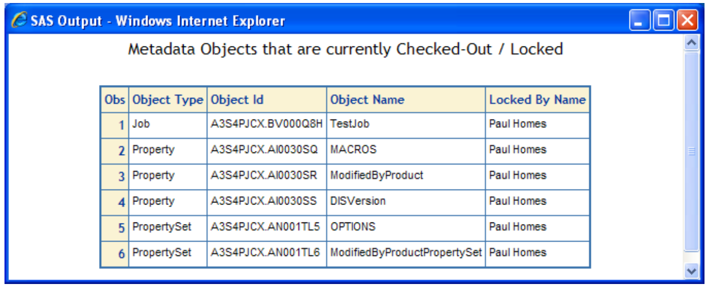
```
filename sxlemap temp;

data _null_;
file sxlemap;
put '<SXLEMAP version="1.2" name="Persons">';
put '<TABLE name="Persons">';
put '<TABLE-PATH syntax="XPath">//Objects/Person</TABLE-PATH>';
put '<COLUMN name="PersonId">';
put '<PATH syntax="XPath">//Objects/Person/@Id</PATH>';
…
put '<COLUMN name="PersonName">';
put '<PATH syntax="XPath">//Objects/Person/@Name</PATH>';
…
put '<COLUMN name="PersonTitle">';
put '<PATH syntax="XPath">//Objects/Person/@Title</PATH>';
…
put '<COLUMN name="PersonEmailAddress">';
put '<PATH syntax="XPath">
        //Objects/Person/EmailAddresses/Email@Address</PATH>';
…
```

TekEds.com

Use SXLE to Generate Report

```
* assign the SXLE library using the XML response and the SXLE map;
libname resplib xml xmlfileref=response xmlmap=sxlemap;

* generate a simple report;
proc print data=resplib.Persons;
run;

* clear the SXLE library;
libname resplib;

* release the temporary SXLE map;
filename sxlemap;

* release the temporary response;
filename response;
```

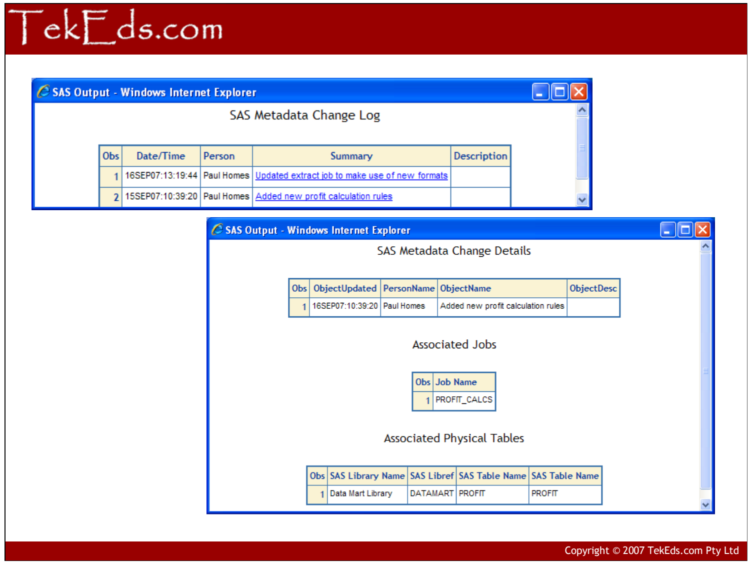
TekEds.com



Metadata Objects that are currently Checked-Out / Locked

Obs	Object Type	Object Id	Object Name	Locked By Name
1	Job	A3S4PJCX.BV000Q8H	TestJob	Paul Homes
2	Property	A3S4PJCX.AI0030SQ	MACROS	Paul Homes
3	Property	A3S4PJCX.AI0030SR	ModifiedByProduct	Paul Homes
4	Property	A3S4PJCX.AI0030SS	DISVersion	Paul Homes
5	PropertySet	A3S4PJCX.AN001TL5	OPTIONS	Paul Homes
6	PropertySet	A3S4PJCX.AN001TL6	ModifiedByProductPropertySet	Paul Homes

This is an example metadata report shows a listing of objects that are currently checked out and by whom. It could be consulted prior to metadata promotion between repositories to ensure all expected changes are complete and checked-in.



TekEds.com

SAS Output - Windows Internet Explorer

SAS Metadata Change Log

Obs	Date/Time	Person	Summary	Description
1	16SEP07:13:19:44	Paul Homes	Updated extract job to make use of new formats	
2	15SEP07:10:39:20	Paul Homes	Added new profit calculation rules	

SAS Output - Windows Internet Explorer

SAS Metadata Change Details

Obs	ObjectUpdated	PersonName	ObjectName	ObjectDesc
1	16SEP07:10:39:20	Paul Homes	Added new profit calculation rules	

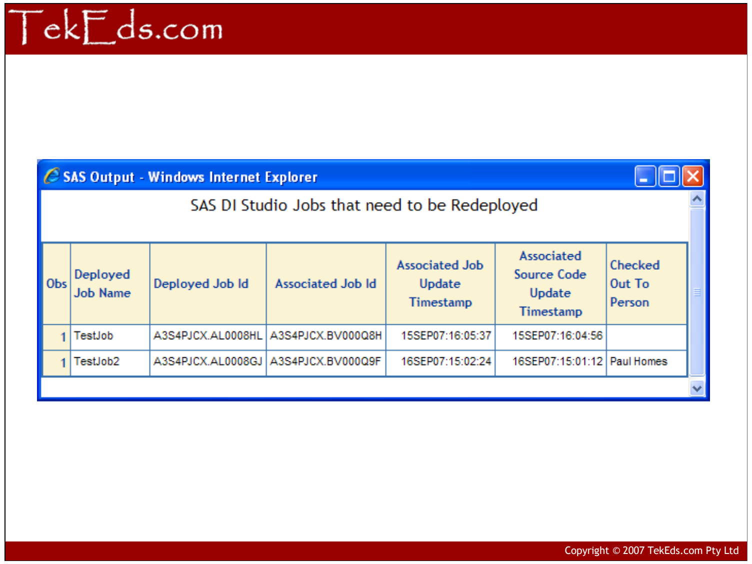
Associated Jobs

Obs	Job Name
1	PROFIT_CALCS

Associated Physical Tables

Obs	SAS Library Name	SAS Libref	SAS Table Name	SAS Table Name
1	Data Mart Library	DATAMART	PROFIT	PROFIT

This is an example metadata report that shows a chronological list of metadata changes (via change control commit). Clicking on an individual change displays more detail including associated jobs and physical tables. This report can be used to identify changes and associated objects for partial promotion between repositories.



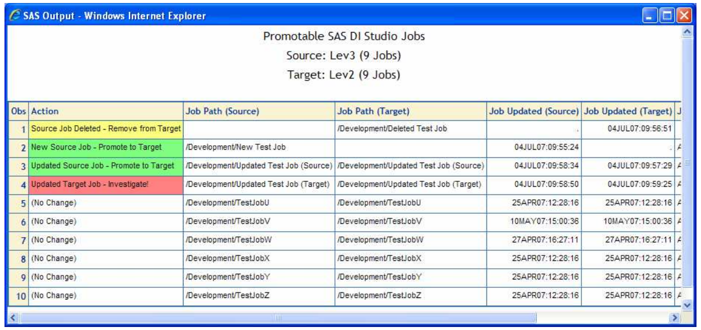
TekEds.com

SAS Output - Windows Internet Explorer

SAS DI Studio Jobs that need to be Redeployed

Obs	Deployed Job Name	Deployed Job Id	Associated Job Id	Associated Job Update Timestamp	Associated Source Code Update Timestamp	Checked Out To Person
1	TestJob	A3S4PJCX.AL0008HL	A3S4PJCX.BV000Q8H	15SEP07:16:05:37	15SEP07:16:04:56	
1	TestJob2	A3S4PJCX.AL0008GJ	A3S4PJCX.BV000Q9F	16SEP07:15:02:24	16SEP07:15:01:12	Paul Homes

This is an example metadata report that shows SAS Data Integration Studio jobs that may need to be re-deployed for scheduling. The job metadata has a newer timestamp that the associated deployed source code. If the job is not re-deployed the old source code may be run by the scheduler. Deploying all jobs for scheduling can take a significant amount of time for large repositories and this report helps to narrow the list down to only those that need to be re-deployed.



Promotable SAS DI Studio Jobs
Source: Lev3 (9 Jobs)
Target: Lev2 (9 Jobs)

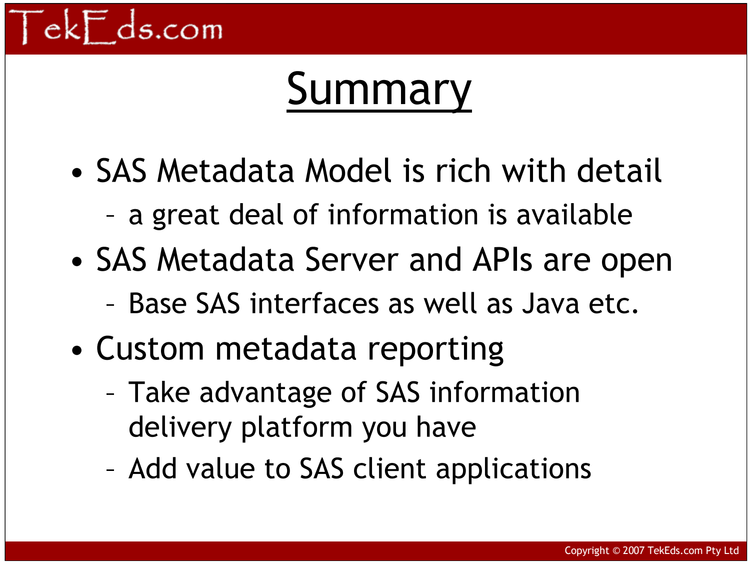
Obs	Action	Job Path (Source)	Job Path (Target)	Job Updated (Source)	Job Updated (Target)	J
1	Source Job Deleted - Remove from Target		/Development/Deleted Test Job	.	04JUL07:09:56:51	
2	New Source Job - Promote to Target	/Development/New Test Job		04JUL07:09:55:24	.	A
3	Updated Source Job - Promote to Target	/Development/Updated Test Job (Source)	/Development/Updated Test Job (Source)	04JUL07:09:58:34	04JUL07:09:57:29	A
4	Updated Target Job - Investigate!	/Development/Updated Test Job (Target)	/Development/Updated Test Job (Target)	04JUL07:09:58:50	04JUL07:09:59:25	A
5	(No Change)	/Development/TestJobU	/Development/TestJobU	25APR07:12:28:16	25APR07:12:28:16	A
6	(No Change)	/Development/TestJobV	/Development/TestJobV	10MAY07:15:00:36	10MAY07:15:00:36	A
7	(No Change)	/Development/TestJobW	/Development/TestJobW	27APR07:16:27:11	27APR07:16:27:11	A
8	(No Change)	/Development/TestJobX	/Development/TestJobX	25APR07:12:28:16	25APR07:12:28:16	A
9	(No Change)	/Development/TestJobY	/Development/TestJobY	25APR07:12:28:16	25APR07:12:28:16	A
10	(No Change)	/Development/TestJobZ	/Development/TestJobZ	25APR07:12:28:16	25APR07:12:28:16	A

This is an example metadata report that shows a comparison of jobs and their metadata update timestamps for a development (Lev3) and test (Lev2) environment. It helps to identify jobs that may need to be promoted. The differences are colour coded. The green jobs need to be promoted from the source to the target environment. The yellow jobs need to be deleted from the target environment. The red jobs are newer in the target environment and may indicate a problem with change control procedures.

What else?

- METABROWSE
  - Unhide All
  - Find…
- SAS Management Console
  - Tools > Metadata Utility
- SAS Data Integration Studio
  - Advanced Tab
- Data step metadata_* functions

A few other items to mention if time allows.



TekEds.com

Summary

- SAS Metadata Model is rich with detail
  - a great deal of information is available
- SAS Metadata Server and APIs are open
  - Base SAS interfaces as well as Java etc.
- Custom metadata reporting
  - Take advantage of SAS information delivery platform you have
  - Add value to SAS client applications

TekEds.com

Questions?