

SASメタデータのバッチ登録

迫田 英之

株式会社タクミインフォメーションテクノロジー
システム開発推進部

Registering SAS Metatadata Objects in batch execution

Hideyuki Sakoda
System Development Dept.,
Takumi Information Technology Inc.

要旨：

SAS BI Server , SAS Data Integration Server などのSASプラットフォーム製品は、ユーザー・テーブル・ストアプロセスといった管理対象オブジェクトをSAS Metadata Serverに登録することで初めて利用可能となる。SAS Metadata Serverへのオブジェクト登録は、通常SAS 管理コンソールによる手動操作で実行するが、本論文では大量登録や自動連携の仕組みを構築する際に必要となるバッチ(一括)実行方法を紹介する。

キーワード:プラットフォーム SASシステム

はじめに

SAS Data Integration Server , SAS Enterprise BI Server 環境下では、一般的には、

1. SAS管理コンソールで、ユーザーとグループを登録
2. SAS Data Integration Studio でテキストデータ、外部データベース、キューブ、SASデータセットをテーブル登録
3. SAS Enterprise Guide でストアドプロセスを作成・登録
4. SAS Information Map Studio でインフォメーションマップを作成・登録
5. SAS Web Report Studioでレポートを作成・登録

というようにGUIツールを用いて、人、データ、処理を定義し、SASメタデータとして登録される。(4.,5.はWeb環境用)

この度、ユーザー情報を複数システム間で自動的に連携する仕組み(当社はSASサーバー部分のみ)の構築を受託することになり、バッチ登録が必要となった。その時に調査、検証したことを紹介する。

バッチ登録方法の分類

以下にメタデータオブジェクト種類毎に可能なバッチ登録方法を示す。

オブジェクト種類	ユーザー・グループ	テーブル	キューブ	インフォメーションマップ	ストアドプロセス
User Import Macros	○				
Proc METADATA	△	△	△	△	△
メタデータ用DATA ステップ関数	△	△	△	△	△
Proc METALIB		○			
Proc OLAP			○		
Proc INFOMAPS				○	

※ △:可能だが容易ではない。
別途、メタデータのエクスポート/インポートツールあり。

ユーザー・グループ登録方法の検討

ユーザー・グループ登録方法として、まず User Import Macrosを検討した。

受託したシステムの要件)

- 差分情報のみ連携される

User Import Macrosの仕様)

- 新規のユーザー・グループ登録が可能
- ユーザー・グループのマスター情報は全件読めることが必要

以上から、User Import Macrosの利用をあきらめた。User Import Macrosは
!SASROOT¥SAS¥SASFoundation¥9.2¥core¥sasmacro
に格納されており閲覧可能。主に proc METADATAを使用していたことがわかった。
従って、proc METADATA使用の検討に入った。

User Import Macrosの詳細情報は以下に記載。

「SAS(R) 9.2 Intelligence Platform: Security Administration Guide」

<http://support.sas.com/documentation/cdl/en/bisecag/61133/HTML/default/viewer.htm#a003175544.htm>

proc METADATAのinput/output は何か

構文) proc METADATA IN=ファイル参照名1 out=ファイル参照名2 header=full;
run;

SASメタデータはSAS Metadata Modelと呼ばれる階層構造でできているが Proc MetadataのinputにはIOMI メソッドを、outputファイルはSAS Metadata Modelそのもののデータ構造をXMLで受け取ることがわかった。そのためまずSAS Metadata Modelを理解することにした。

proc METADATAの詳細情報は以下に記載。

「SAS(R) 9.2 Language Interfaces to Metadata」

http://support.sas.com/documentation/cdl/en/lrmeta/60739/HTML/default/viewer.htm#proc_metadata.htm

SAS Metadata Modelの詳細情報は以下に記載。

「SAS(R) 9.2 Metadata Model: Reference」

<http://support.sas.com/documentation/cdl/en/omamodref/61849/HTML/default/viewer.htm#titlepage.htm>

IOMI メソッドの詳細情報は以下に記載。

「SAS(R) 9.2 Open Metadata Interface: Reference and Usage」

<http://support.sas.com/documentation/cdl/en/omaref/59983/HTML/default/viewer.htm#a002969912.htm>

SAS Metadata Modelの構造概要

SAS Metadata Model は様々なTypeに分類される。

(例: Person:人、PhysicalTable:テーブル)

オブジェクトはIDとPropertyを持つ。Propertyは、AttributeとAssociationからなる。

Associationは階層構造上、子に位置するTypeである。

ユーザーやグループに関するType(主要なもの)

Identity

Person (ユーザー)

IdentityGroup (グループ)

Login (ユーザーID,パスワード)

AuthenticationDomain (認証ドメイン)

Person のAttribute

Name , PublicType , UsageVersion , MetadataCreated , MetadataUpdated

Person のAssociation

ExternalIdentities (Type=ExternalIdentity)

IdentityGroups (Type=IdentityGroup)

Logins (Type>Login)

<= ユーザーとグループの紐付け

<= ユーザーとユーザーIDの紐付け

metabrowseコマンド (SASウィンドウコマンド)

The screenshot shows the SAS Metadata Browser window. The left pane displays a tree view of the metadata environment, with the 'Person' object selected. The right pane shows the properties and associations for the selected 'Person' object.

名前	値
UsageVersion	1000000
Id	A5FST44Y.AN00000Q
MetadataCreated(GMT)	2011年07月02日 10時...
MetadataUpdated(GMT)	2011年07月02日 10時...
Name	パーソン1
PublicType	User
ExternalIdentities	Association
IdentityGroups	Association
Logins	Association

SAS Metadata Model (XML形式表記)

以下はPerson「パーソン1」をXML形式表記したものである(前ページと同じオブジェクト)。

```
<Person Id="A5FST44Y.AN00000Q" Name="パーソン1" Desc="" Title="" DisplayName="">
  <ExternalIdentities>
    <ExternalIdentity Id="A5FST44Y.BF00000W" Context="IdentityImport" Identifier="P001" />
  </ExternalIdentities>
  <IdentityGroups>
    <IdentityGroup Id="A5FST44Y.A300000Z" Name="グループ1" />
  </IdentityGroups>
  <Logins>
    <Login Id="A5FST44Y.AQ00000K" Name="D7CP88BX¥testuser1" Desc="パーソン1 login"
      UserID="D7CP88BX¥testuser1" Password="*****">
      <Domain>
        <AuthenticationDomain Id="A5FST44Y.AP000001" Name="DefaultAuth">
          <ExternalIdentities />
        </AuthenticationDomain>
      </Domain>
    </Login>
  </Logins>
  <EmailAddresses />
  <Locations />
  <PhoneNumbers />
</Person>
```

メタデータのID形式
全てのオブジェクトはユニークなIDを持ち
その形式は、リポジトリID.オブジェクトID

Proc METADATAの指定方法と準備

```
options metaport=8561
      metaserver="localhost"
      metauser="sasadm@saspw"
      metapass="xxxxx"
      metarepository="Foundation"
;

filename request "D:¥temp¥req.xml";
filename response "D:¥temp¥res.xml" lrecl=1048576;
proc METADATA in=request out=response header=full;
run;
```

出力XMLは改行なしのため、
大きい値を指定

出力XMLに以下のヘッダ一行を付与。
(日本語版の場合は以下の通り)
<?xml version="1.0" encoding="shift_jis"?>

Proc METADATAのinput (IOMIメソッド)

以下はType:Person を全て表示(ただしAttributeはId ,Name)させる IOMIメソッドである。

```
<GetMetadataObjects>  
  <Reposid>$METAREPOSITORY</Reposid>  
  <Type>Person</Type>  
  <Objects/>  
  <NS>SAS</NS>  
  <Flags>0</Flags>  
  <Options/>  
</GetMetadataObjects>
```

システムオプション
METAREPOSITORYで指定され
た値が使用

表示のさせ方やオプション指定を制御

Proc METADATAのoutput

前ページを入力に、proc METADATA実行後の結果例は以下の通りである。

```
<?xml version="1.0" encoding="shift_jis" ?>
<GetMetadataObjects>
<Reposid>A0000001.A5FST44Y</Reposid>
<Type>Person</Type>
```

メタデータのID形式に変換される

```
<Objects>
  <Person Id="A5FST44Y.AN000001" Name="sasadm" />
  <Person Id="A5FST44Y.AN000002" Name="sastrust" />
  <Person Id="A5FST44Y.AN000003" Name="sasdemo" />
  <Person Id="A5FST44Y.AN00000Q" Name="パーソン1" />
  <Person Id="A5FST44Y.AN00000R" Name="パーソン2" />
  <Person Id="A5FST44Y.AN000018" Name="パーソン3" />
  <Person Id="A5FST44Y.AN0000S4" Name="パーソン4" />
</Objects>
```

GetMetadataObjectメソッドの結果が返る部分

```
</GetMetadataObjects>
<NS>SAS</NS>
<Flags>0</Flags>
<Options />
```

Proc METADATAのinput (IOMIメソッド) (2)

以下はType:Person を全て表示(PersonのAttributeは全て表示し、AssociationはId, Nameのみ)させる IOMIメソッドである。

```
<GetMetadataObjects>  
  <Reposid>$METAREPOSITORY</Reposid>  
  <Type>Person</Type>  
  <Objects/>  
  <NS>SAS</NS>  
  <Flags>257</Flags>  
  <Options/>  
</GetMetadataObjects>
```

システムオプション
METAREPOSITORYで指定され
た値が使用

表示のさせ方やオプション指定を制御。
フラグを足しこんだ値を指定。
今回は、
OMI_ALL(1) + OMI_GET_METADATA (256)

IOMI フラグの詳細情報は以下に記載。

「SAS(R) 9.2 Open Metadata Interface: Reference and Usage」

http://support.sas.com/documentation/cdl/en/omaref/59983/HTML/default/viewer.htm#flag_table.htm

Proc METADATAのoutput (2)

前ページを入力に、proc METADATA実行後の結果例は以下の通りである。

```
<?xml version="1.0" encoding="shift_jis" ?>
<GetMetadataObjects>
<Reposid>A0000001.A5FST44Y</Reposid>
<Type>Person</Type>
<Objects>
  <Person Id="A5FST44Y.AN000001" Name="sasadm" ChangeState="" Desc=""
    DisplayName="SAS Administrator" LockedBy="" MetadataCreated="05Dec2010:13:40:18"
    MetadataUpdated="05Dec2010:13:40:18" PublicType="User" Title=""
    UsageVersion="1000000">
    <AccessControlEntries>
      <AccessControlEntry Id="A5FST44Y.A500006X" Name="AJ00001Y:A500006X" Desc="" />
    </AccessControlEntries>
    <AccessControls />
    <AssociatedHomeFolder>
      <Tree Id="A5FST44Y.AJ00001X" Name="sasadm" Desc="" />
    </AssociatedHomeFolder>
```

... 略 ...

Proc METADATAのinput (IOMIメソッド) (3)

IOMIフラグを用いて、出力項目を制御(Template)したり、フィルタ(XMLSelect)をかけることができる。

```
<GetMetadataObjects>
  <Reposid>$METAREPOSITORY</Reposid>
  <Type>Person</Type>
  <Objects/>
  <NS>SAS</NS>
  <Flags>388</Flags> <!-- OMI_XMLSELECT(128) + OMI_GET_METADATA(256)
                       + OMI_TEMPLATE(4) -->
```

```
<Options>
```

```
<XMLSelect search="Person[@Name = 'パーソン1']"/>
```

フィルタ条件指定

```
<Templates>
```

```
<Person Id="" Name="" Desc="" title="" DisplayName="">
```

```
<ExternalIdentities />
```

```
<Logins />
```

```
<EmailAddresses />
```

```
<Locations />
```

```
<PhoneNumbers />
```

```
</Person>
```

```
<ExternalIdentity Context="" Identifier="" />
```

```
<Login Id="" Name="" Desc="" UserId="" Password="" >
```

```
<Domain />
```

```
</Login>
```

以下、略

Proc METADATAのinputをどのように作成するか

Proc METADATAのinputはテキストXMLである。様々な方法が考えられるが、以下はDATAステップで作成する例である。(パラメータをセットする方法にも対応しやすい)

```
filename request temp;
data _null_;
  file request;
  put '<GetMetadataObjects>';
  put ' <Reposid>$METAREPOSITORY</Reposid>';
  put ' <Type>Person</Type>';
  put ' <Objects/>';
  put ' <NS>SAS</NS>';
  put ' <Flags>257</Flags>';
  put ' <Options/>';
  put '</GetMetadataObjects>';
run;
```


Proc METADATAのoutputをどのように扱うか

Proc METADATAのoutputはXMLである。列情報はSAS Metadata Modelに従う。XMLエンジンの Libnameが利用できる。MAPファイルが必要となる。以下はDATAステップでMAPファイルを作成する例である。

```
filename usermap temp;
data _null_;
  file usermap ;
  put '<?xml version="1.0" encoding="shift_jis" ?>';
  put ' ';
  put '<SXLEMAP version="1.2" name="UserInfoMap">';
  put ' ';
  put ' <!-- ##### -->';
  put ' <TABLE name="Person">';
  put '   <TABLE-PATH syntax="XPath">/GetMetadataObjects/Objects/Person</TABLE-PATH>';
  put ' ';
  put '   <COLUMN name="Id">';
  put '     <PATH syntax="XPath">/GetMetadataObjects/Objects/Person/@Id</PATH>';
  put '     <TYPE>character</TYPE>';
  put '     <DATATYPE>string</DATATYPE>';
  put '     <LENGTH>17</LENGTH>';
  put '   </COLUMN>';
```

～略～

Proc METADATAのoutputをどのように扱うか (2)

以下はXMLエンジンのLIBNAME使用例である。後続処理としてDATAステップ等で加工処理の入力としてXMLエンジンのライブラリは使用できないので、別のライブラリ（以下の例ではWORK）にコピーしておく。

```
Filename response temp; /* proc metadataの out */  
Filename usermap temp; /* XML MAPファイルのファイル参照名 */
```

～MAPファイル作成処理(略)～

```
LIBNAME metpinf XML xmlfileref=response xmlmap=usermap access=READONLY;
```

```
proc copy in=metpinf out=work;  
  select person ;  
run;
```

更新系処理のバリエーションとIOMIメソッド

今回そもそも本論文作成のきっかけとなったバッチ処理要件は、

- ユーザーとユーザーに対するグループの紐付けの差分情報が連携される。
- メタデータの権限はグループに設定されるが、ユーザーには設定されない。
- ユーザーに対するグループとの紐付け情報は連携されるが、グループそのものの追加、削除、変更は連携処理としては実施しない(SAS管理コンソール使用)

→以上から権限設定情報をバッチ処理に組み入れ不要。

また環境として認証ドメインはDefaultAuthを使用し、UNIXホスト認証であった。

他の認証ドメインは存在しない。この場合、メタデータリポジトリ内の各ユーザー毎のパスワードは使用されないためパスワードを無視してよい。

このような条件と、IOMIメソッドが出来ることを組み合わせると、以下の処理バリエーションが出来上がった。

- ユーザー登録 → AddMetadata メソッド
- ユーザーに対するグループの紐付け(既存のグループ紐付け削除 と 新規のグループ紐付け追加 の2種類が必要) → UpdateMetadata メソッド
- ユーザー削除 → DeleteMetadata メソッド

※ 実行前に必ずメタデータリポジトリのバックアップを取得すること

メタデータの追加 (AddMetadata)

1ユーザーを追加する例を示す

```
<AddMetadata>
<Metadata>
  <Person Id="$Persons" Name="パーソン4" DisplayName=" " Desc=" " Title=" "
    PublicType="User" UsageVersion="1000000" >
  </Person>
  <Login Id="$Logins" Name="パーソン4" Desc=" " UserID="D7CP88BX¥testuser4" >
    <Domain>
      <AuthenticationDomain ObjRef="A5FST44Y.AP000001" />
    </Domain>
    <AssociatedIdentity>
      <Person ObjRef="$Persons" />
    </AssociatedIdentity>
  </Login>
</Metadata>
<Reposid>$METAREPOSITORY</Reposid>
<NS>SAS</NS>
<Flags>268435456</Flags>
<Options/>
</AddMetadata>
```

AuthenticationDomainのオブジェクトIDは
事前に調べておく

\$が付く場合、同一名と同じ値を意味し、
AddMetadataの時のIDは新規に付与されること
を意味する。

OMI_TRUSTED_CLIENT
(268435456) flag
更新系に当フラグは必須

メタデータの更新 (UpdateMetadata) (1)

1ユーザーを更新(グループ紐付けを削除)する例を示す。
同一ユーザー(Person)に対して一度にグループ紐付けの削除と追加をすることは不可。

```
<UpdateMetadata>
<Metadata>
<Person Id="A5FST44Y.AN0000S2">
  <IdentityGroups Function="Remove">
    <IdentityGroup ObjRef="A5FST44Y.A30000SC"/>
  </IdentityGroups>
</Person>
</Metadata>
<NS>SAS</NS>
<!-- OMI_TRUSTED_CLIENT Flag -->
<Flags>268435456</Flags>
<Options/>
</UpdateMetadata>
```

更新対象PersonのオブジェクトIDは事前に調べておく

当該Personの既存グループ (IdentityGroup)のオブジェクトIDは事前に調べておく

OMI_TRUSTED_CLIENT
(268435456) flag
更新系に当フラグは必須

メタデータの更新 (UpdateMetadata) (2)

1ユーザーを更新(グループ紐付けを追加)する例を示す。

同一ユーザー(Person)に対して一度にグループ紐付けの削除と追加をすることは不可。

```
<UpdateMetadata>
<Metadata>
<Person Id="A5FST44Y.AN0000S3">
  <IdentityGroups Function="Append">
    <IdentityGroup ObjRef="A5FST44Y.A300000Z"/>
  </IdentityGroups>
</Person>
</Metadata>
<NS>SAS</NS>
<!-- OMI_TRUSTED_CLIENT Flag -->
<Flags>268435456</Flags>
<Options/>
</UpdateMetadata>
```

更新対象PersonのオブジェクトIDは事前に調べておく

当該Personの既存グループ (IdentityGroup)のオブジェクトIDは事前に調べておく

OMI_TRUSTED_CLIENT
(268435456) flag
更新系に当フラグは必須

メタデータの削除 (DeleteMetadata)

1ユーザーを削除し、同時に削除したいAssociationを指定する方法を示す。

```
<DeleteMetadata>
<Metadata>
<Person Id="A5FST44Y.AN0000S3" TemplateName="myassns"/>
</Metadata>
<NS>SAS</NS>
<!--OMI_TEMPLATE + OMI_TRUSTED_CLIENT + OMI_RETURN_LIST -->
<Flags>268436484</Flags>
<Options>
  <Templates>
    <Template TemplateName="myassns">
      <Person>
        <ExternalIdentities/>
        <Logins/>
        <EmailAddresses/>
        <Locations/>
        <PhoneNumbers/>
      </Person>
    </Template>
  </Templates>
</Options>
</DeleteMetadata>
```

削除対象PersonのオブジェクトIDは事前に調べておく

Associationを含め、削除したオブジェクトIDをproc METADATAのout=で指定したXMLファイルに出力

Person削除と同時に削除したいAssociationを指定

おわりに

ユーザー・グループ情報をバッチ登録する場合、User Import Macrosが要件を満たせなかったのは残念だったが、proc METADATAによる方法で、メタデータの情報取得、登録とも、要件を満たす処理を作ることが出来た。

このproc METADATAによる方法は、原理的にどのようなオブジェクトTypeでも対応できるものだが、SAS Metadata Modelというデータ構造を理解し、実行方法やオプションを理解していくところで、マニュアルが英語のみということもあり、時間がかかった。

GetMetadataObjects 等を使った参照系処理は、メタデータに登録されている情報からレポートを作成することにも使えるので、部品としてプログラムを作っていくと管理作業が楽になると思われる。

SASフォルダで参照可能なフォルダ構造は、Type:Tree として参照可能であり、今後Treeを使用した参照処理を作りたいと思う。

- 問い合わせ先
株式会社タクミインフォメーションテクノロジー
システム開発推進部
迫田 英之(さこだ ひでゆき)
E-mail : hideyuki.sakoda@takumi-it.co.jp
Tel: 03-5979-7381

