

From InForm EDC Data to SAS Datasets using SAS Clinical Data Integration – Real Time

Jitesh Nagaria, SAS Institute, Cary, NC

ABSTRACT

In the pharmaceutical industry, getting the real time data from EDC systems into SAS Datasets is still the biggest challenge. Companies spend significant amount of time and resources in synchronizing data from EDC systems to SAS datasets. As the industry becomes more and more competitive, having the right data in hand for analysis as soon as it has been entered is crucial and challenging.

This paper talks about how to integrate and synchronize data from InForm EDC system through SAS Clinical Data Integration and convert it into SAS Datasets. It will describe the process flow and technical details required to set up the connection between SAS Clinical Data Integration and InForm EDC trial. It will also discuss how to fetch the metadata, clinical data in the form of ODM XML, parse the ODM XML and produce the SAS Datasets for each clinical domain. The integration involves technologies like SOAP, XML, XSL, SAS programs and SAS Clinical Standard Toolkit.

INTRODUCTION

For any pharmaceutical company, the need and frequency of having the clinical data ready for analysis is a driving force for their clinical trials and data management. InForm EDC software is being used in the industry by many companies for capturing clinical trials data. The major challenge is to convert the data coming out of InForm into SAS Datasets, and also making sure they are in sync. Some of the challenges are:

- When data is fetched from InForm EDC, it is in the form of Operational Data Model (ODM)
- Operational Data Model (ODM) is a complex XML file
- Due to high complexity of the ODM XML file, traditional XML SAS engine does not serve the purpose
- Synchronizing the metadata updates and clinical data updates between InForm EDC and SAS Datasets
- Ability to get Incremental Transactional data

The methodology described in this paper provides the solution to these challenges.

Software versions used in the paper are:

- SAS® Clinical Data Integration Version 2.2
- SAS® 9.2 M3
- InForm EDC Version 4.6 SP2a
- InForm Adapter Version 1.3 SP4

BACKGROUND

It is important to understand the background of InForm Adaptor SOAP based web services and how it can be used in SAS Clinical Data Integration, using SAS 9.2 PROC SOAP procedure.

INFORM ADAPTOR

The InForm Adapter provides interfaces to Web services that support the secure transfer of data between an InForm database and third-party products and custom applications. InForm Adapter interfaces use published Web service interfaces to allow programmatic access to applications.

The InForm Adapter 1.3 SP4 release includes the interfaces shown in Table 1 that are available for programmatic access.

Interface Name	Description
Admin	This interface provides access to information about the location of InForm servers and InForm trials.
Discrepancy	This interface transmits InForm queries (discrepancies) to and from other systems or modules.
Eventing	This interface publishes a particular event or set of events when they occur within the InForm application.
ODM Export	This interface provides access to receive InForm clinical data and metadata in CDISC ODM standard format.

Table 1. InForm Adaptor Interfaces

The ODM Export interface is a unidirectional interface that provides a web service for any outside application that needs to retrieve data from an InForm database in CDISC ODM standard format. Both clinical data and metadata are available. The ODM Export interface connects directly to the InForm database.

The scope of this paper is restricted to ODM Export interface; however, the framework can be applied to other interfaces as well.

SAS CLINICAL DATA INTEGRATION

SAS Clinical Data Integration supports the pharmaceutical industry needs for transforming, managing, and verifying the creation of industry-mandated data standards such as those created by Clinical Data Interchange Standards Consortium (CDISC). SAS Clinical Data Integration leverages the SAS Clinical Standards Toolkit to provide validation, conformance checking and also parsing the ODM XML files and converting them into SAS datasets.

One of the important features of SAS Clinical Data Integration is ability to create custom transformation. As part of this paper, a set of user-written custom transformations are developed, which allow a connection to the InForm EDC system to extract the data and convert them to data standards.

HIGH LEVEL SYSTEM OVERVIEW

Below, Figure 1 shows the high level system overview. A SOAP request is being made to the InForm Adaptor server from SAS Clinical Data Integration. In response, the clinical data or metadata is returned in the form of ODM XML, which is processed to convert into SAS datasets. The final output SAS datasets can be stored into a file system, database or SAS Drug Development using WebDAV libname or SAS Drug Development Remote API.

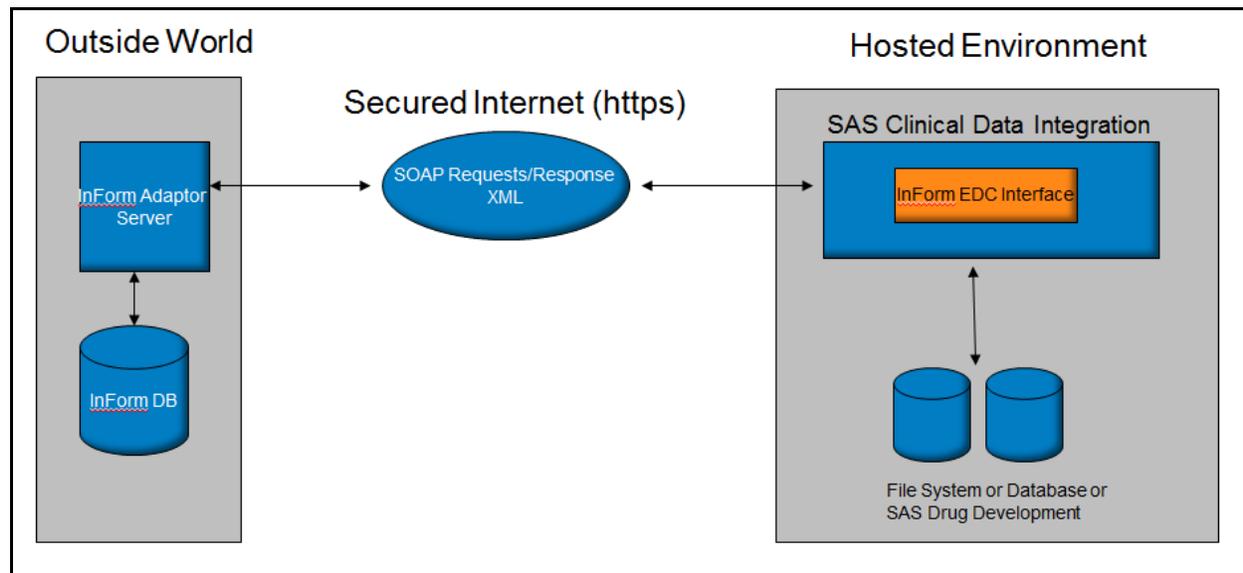


Figure 1. High Level System Overview

SOAP requests are XML files with required information in the header and body that is understood by InForm Adaptor web services. Table 2 shows the important SOAP request methods in the InForm ODM Export Interface.

SOAP Method Name	Description
DownloadMetadaODM	This method returns the definitions of all the metadata (for example, study versions, visits, and forms) in a study.
DownloadAdminDataODM	This method returns lists of all users, sites, and signature definitions in a study.
GetTransactionStatus	This method returns the status of transactions that have been processed.
GetTransactions	This method returns a list of all the subject-related transactions for a specified study.

Table 2. ODM Export SOAP Methods

STEP BY STEP DESIGN

SAS CLINICAL DATA INTEGRATION CONFIGURATIONS

The following are required steps to be applied on SAS Clinical Data Integration Application Servers in order to allow PROC SOAP to connect to the InForm Adaptor server.

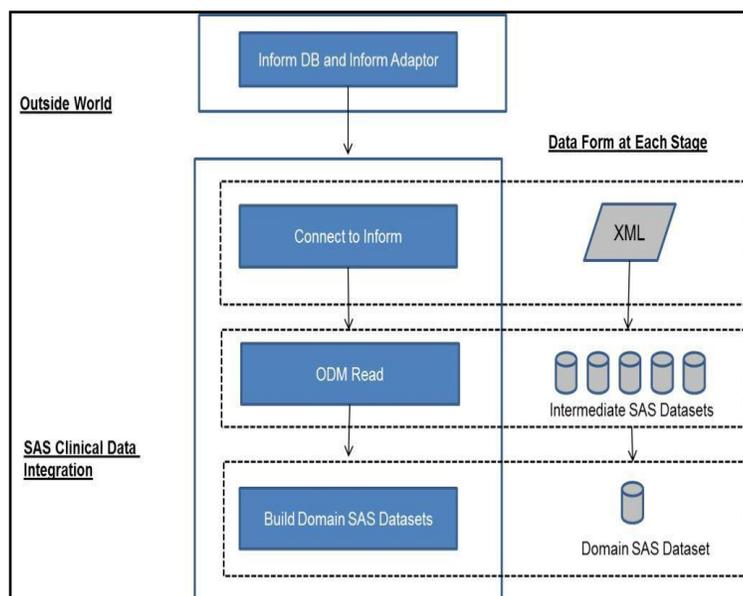
- The security certificate, which is provided by the vendor who is hosting InForm, must be installed on the SAS Application Server of SAS Clinical Data Integration.
- The name of the certificate must match the URL that is being accessed to connect to the InForm Adapter. For example, if the name of the certificate is *.company1.com.*, then the URL should also contain *.company1.com.*
- The certificate must be exported in a DER encoded binary file format (.cer) (For example: D:\temp\cert_der.cer). Using the keytool command, this certificate is registered in the Java KeyStore. For example, use the command:

```
C:\Program Files\Java\jre1.5.0_12\bin>keytool -import -trustcacerts -keystore
C:\Program Files\java\jre1.5.0_12\lib\security\cacerts -storepass changeit -file
D:\temp\cert_der.cer -alias Company1
```

PROCESS FLOW

Figure 2 shows the process flow of extracting data from InForm to creating SAS datasets. It also shows the corresponding data flow at each process. Each process is described in detail in the following sections.

Figure 2 Process Flow



- **Connect to InForm:** This is the first step in the process flow, which allows making a connection to the InForm Adaptor server. Required parameters are: URL of the InForm Adaptor, TrialName, WebService Name (one of the SOAP methods described in Table 2). Using these parameters, the SOAP request XML file is built and PROC SOAP is executed. The form of data at this stage is XML in ODM standard.
- **ODM Read:** This is the most important step of the entire system. The ODM Read module is part of SAS Clinical Standard Toolkit, which is installed in SAS Clinical Data Integration. SAS Clinical Standard Toolkit provides a set of pre-compiled SAS macros to allow reading and parsing of the ODM XML file. In output it produces normalized SAS datasets representing the same information as the ODM XML file.

- **Build Domain SAS Datasets:** At this step, the normalized SAS datasets are joined together to produce a single Domain SAS Dataset. Output of this step is one SAS dataset per domain (DM, AE, MH etc). This step uses the standard SAS Data Integration capability to perform various SQL joins, Extract, Sort, Append transformations. In addition user-written SAS code is also used to achieve the desired output.

CONNECT TO INFORM

There are two categories of information required to make a connection to InForm. One is required for PROC SOAP to execute and process the connection, and the other one is for InForm web services to execute the SOAP request query and provide the response accordingly.

PARAMETERS FOR PROC SOAP

This category defines all the information required for PROC SOAP to make a connection to the InForm Adaptor. Table 3 describes the required parameters for PROC SOAP:

PROC SOAP Parameters	Description
IN	This is SOAP Request XML filename reference
OUT	This is SOAP Response XML filename reference
ENVELOPE	Specifies that a SOAP envelope is to be included in the response
URL	This is WSDL URL for InForm Adaptor
SOAPACTION	This is SOAP Method described in Table 2 for InForm
WSSUSERNAME	This is InForm username. This parameter allows WS-Security and a UsernameToken is sent with the Web service request for user authentication, security, and encryption
WSSPASSWORD	This is InForm password. This parameter is the WS-Security password for WSSUSERNAME
AXIS2CONFIGFILE	For WS-Security the Axis2 configuration XML file is required. This parameter specifies the complete path to axis2.xml file

Table 3. Parameters for PROC SOAP

Here is the example code for GetTransactions SOAP method which uses all the above mentioned parameters:

```

/* GetTransactions */
filename resq "&target_Loc.\DownloadMetadataODM_request.xml";
filename resp "&target_Loc.\DownloadMetadataODM_response.xml";

PROC SOAP in = resq out = resp
envelope
url = "&url./InFormAdapter/ODM/InFormODM.asmx?wsdl"
soapaction = "http://www.phaseforward.com/InFormAdapter/ODM/1.0/&soapaction."
wssusername = "&username"
wsspassword = "&password"
axis2configfile="&target_Loc.\axis2.xml";
RUN;

```

In SAS Clinical Data Integration, a custom transformation is developed to allow the user to select the values from a user interface. These are referred to as prompts in SAS Clinical Data Integration terminologies. These prompts can be a text box, drop down selection list or directory selection. One can create a static list or a dynamic list with values coming from SAS datasets. The values of these prompts are stored in a defined SAS macro variable, which can be used in the underlying PROC SOAP SAS code.

In the above example code, SAS macro variables &target_Loc, &url, &soapaction, &username and &password are converted into prompts in the SAS Clinical Data Integration transformation for the user to select/enter the values during run time.

Figure 3 shows an example transformation and user interface prompts for accepting these PROC SOAP parameters.

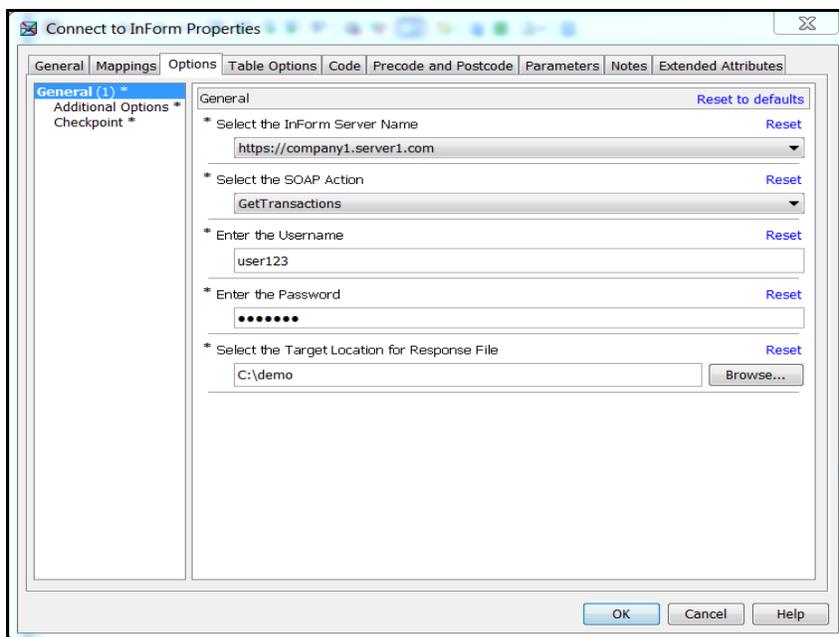


Figure 3. SAS Clinical Data Integration Transformation Parameters for PROC SOAP

PARAMETERS FOR SOAP REQUEST XML

This category defines all the information that needs to be put into the SOAP Request XML file, which is understood by the InForm Adaptor web services. The following section describes how to build the SOAP request for the GetTransactions SOAP method. A similar approach can be used for all the other SOAP methods mentioned in Table 2 with the use of the InForm Adaptor User Guide.

Build SOAP request for GetTransactions SOAP Method

The GetTransactions method returns a list of all the subject-related transactions for a specified study. Along with the ODM document, the xml elements Status and a Bookmark are also returned. Status describes whether all the transactions have been downloaded. Bookmark returns the record after which processing should start the next time that GetTransactions is called. Table 4 shows some of the important input parameters for GetTransactions.

Input Attribute Parameters	Description
TRIAL	Name of the InForm Study from which to retrieve the transaction data
BOOKMARK	This is InForm transaction ID. If not specified then transactions are returned from the beginning
USERNAME	This is InForm username for the requested trial
INCLUDEEXTENSIONS	This is to specify whether to include Phase Forward ODM extensions in the response. The default is True

Table 4. Parameters for GetTransactions SOAP Method

Figure 4 shows the user-written transformation written in SAS Clinical Data Integration to allow the user to enter the values for the parameters listed above. Bookmark is an InForm system generated ID, and is passed into the SOAP request programmatically.

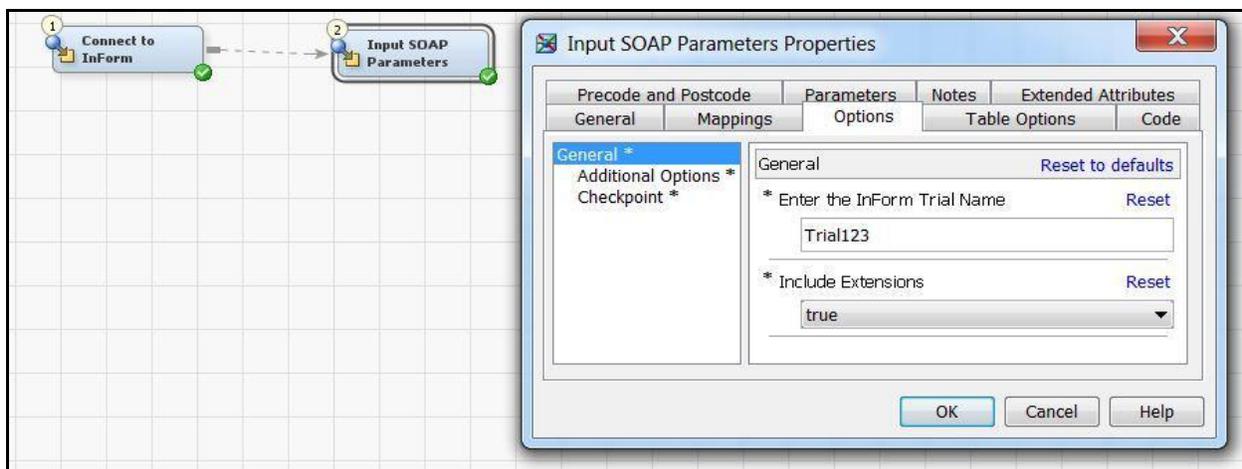


Figure 4. SAS Clinical Data Integration Transformation Parameters for GetTransactions

The values entered by the user in the prompts are stored in the SAS macro variables and can be used to dynamically build the SOAP request XML file. Since the majority of the content of the SOAP request file is static, one can store these prebuilt request XML file permanently and use simple SAS data _null_ steps to resolve and replace only macros variables. For example, Figure 5 shows the SAS code used to read the static request XML, resolve the SAS macro variables and write to another filename for PROC SOAP to run.

```

/* SOAP request xml with unresolved macros */
filename requ "&target_Loc.\request.xml";

/* SOAP request xml with resolved macros */
filename reqr "&target_Loc.\request_resolved.xml";

/* Read request xml, resolve macro variables and write to temp filename for proc
soap */

DATA _null_;
  length str $ 1024;
  infile requ;
  file reqr;
  input;
  str = resolve(_infile_);
  put str;
RUN;

```

Figure 5 SAS Code to write the dynamic SOAP request file

Once the above code is executed, the final SOAP request file looks like Figure 6 which is executed by PROC SOAP

```

<?xml version="1.0" encoding="UTF-8"?>
- <soap:Envelope xmlns:wssu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/03/addressing" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header/>
  - <soap:Body>
    - <node xmlns="http://www.phaseforward.com/InFormAdapter/ODM/1.0">
      <DownloadTransactionODM xmlns="http://www.phaseforward.com/InFormAdapter/ODM/2.0"
        userName="user123" bookmark="" trial="Trial123" includeExtensions="true">
      </DownloadTransactionODM>
    </node>
  </soap:Body>
</soap:Envelope>

```

Figure 6. SOAP Request for GetTransactions

ODM READ

The PROC SOAP execution from the above process returns an XML file with the SOAP response in ODM standard. This response ODM XML file is referenced using SAS Filename reference.

Figure 7 shows an example SOAP response file for GetTransactions.

```
<?xml version="1.0"?>
- <GetTransactionsResult xmlns="http://www.phaseforward.com/InFormAdapter/ODM/1.0">
- <ResponseODM xmlns="http://www.phaseforward.com/InFormAdapter/ODM/2.0"
  xmlns:pf="http://www.phaseforward.com/InFormAdapter/ODM/Extensions/2.0">
- <ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" pf:InFormAdapterVersion="1.0.3999.27031" pf:HierarchicalOIDs="Yes" pf:MappingVersion="V2"
  Originator="system" ODMVersion="1.3" PriorFileOID="{CB219A42-F227-43D8-93F7-DF91AED86EC2}" CreationDateTime="2010-12-
  13T20:38:11+00:00" FileOID="{DD36F77C-D15B-4CC6-905E-C3FE13028A76}" FileType="Transactional" Description="StudyDesign">
- <ClinicalData pf:Revision="253647973055083.000" pf:TransactionGuid="{AB7C93A6-C5F2-49D0-9282-C42A82ED5E14}"
  MetaDataVersionOID="Study Design 1.0.0" StudyOID="StudyDesign">
- <SubjectData pf:GUID="{44C8BF9B-C32B-4C92-9B4C-51A3C8A72D84}" TransactionType="Upsert" SubjectKey="14440">
- <SiteRef LocationOID="12121"/>
- <StudyEventData StudyEventOID="vstCORE8">
- <AuditRecord>
- <UserRef UserOID="mmeyer"/>
- <LocationRef LocationOID="Unknown"/>
- <DateTimeStamp>2009-01-14T22:46:13+00:00</DateTimeStamp>
- <ReasonForChange>Visit Started</ReasonForChange>
</AuditRecord>
- <FormData FormOID="frmDOV">
- <ItemGroupData ItemGroupOID="frmDOV.sctVisitDateandTime">
- <ItemData pf:FormattedDateValue="2007-09-11T12:25:NUL+00:00" Value="2007-09-11T12:25:-+00:00"
  ItemOID="frmDOV.sctVisitDateandTime.DateandTimeofVisit_DOV.DateandTimeofVisit_DOV">
- <AuditRecord>
- <UserRef UserOID="mmeyer"/>
- <LocationRef LocationOID="12121"/>
- <DateTimeStamp>2009-01-14T22:46:13+00:00</DateTimeStamp>
- <ReasonForChange>Initial Entry</ReasonForChange>
- <SourceID>rdinformq079</SourceID>
</AuditRecord>
</ItemData>
</ItemGroupData>
</FormData>
</StudyEventData>
</SubjectData>
</ClinicalData>
</ODM>
<Status>END</Status>
<Bookmark>{DD36F77C-D15B-4CC6-905E-C3FE13028A76}</Bookmark>
<TransactionsProcessed>28</TransactionsProcessed>
</ResponseODM>
</GetTransactionsResult>
```

Figure 7 SOAP Response for GetTransactions

There are two steps involved in parsing the ODM standard SOAP response file coming from InForm.

1. Pre-process the SOAP Response for GetTransactions

- This step is only valid for GetTransactions. For other SOAP methods this step is not required.
- As highlighted in Figure 7, the <Bookmark> and <Status> tags from ODM response file are extracted using standard SAS XML file in the Libname statement. These values are stored in a permanent SAS dataset. Below is the sample SAS code for the same:

```
filename bookmark "&target_Loc.\bookmark.map";
libname resp xml xmlmap = bookmak;

DATA lib1.bookmark;
  length bookmark $100 status $10;
  set resp.responseodm;
RUN;
```

- This bookmark value is passed on to the next GetTransactions SOAP request using the method described in the above sections to fetch the incremental clinical data for the study.

2. SAS Clinical Standard Toolkit

- SAS Clinical Standard Toolkit is part of the SAS Clinical Data Integration product and is available as a set of pre-compiled SAS macros and XSL (EXTensible Stylesheet Language) files.
- The important SAS macro is odm_read. The SOAP Response XML file, which has been pre-processed in the above step, is passed onto the odm_read macro.

- It produces 65 or more output SAS datasets based on the type (SOAP Methods) of an input ODM XML file.
- Each ODM XML element becomes a SAS dataset in the output library. Examples include:
 - odm.sas7bdat
 - ClinicalData.sas7bdat
 - ItemGroupDef.sas7bdat
 - ItemGroupData.sas7bdat
- If the Phase Forward extensions are required to be present in the SAS datasets, then the SAS Clinical Standard Toolkit needs to be modified. This step requires deep knowledge and understanding of SAS Clinical Standard Toolkit architecture.
- The customized toolkit can read the PF extensions and convert it into equivalent SAS datasets.
- There are two categories of PF extensions:
 - **PF attributes:** Within the standard ODM element, these attributes become available in that particular ODM element SAS data set. For example:
 <ClinicalData pf:Revision="311154614921048.000" ... >: After parsing this line, there will be a variable pf_revision in the ClinicalData.sas7bdat SAS dataset.
 - **PF Elements:** These are individual XML elements that are available as a SAS dataset. For example:
 <PF:InFormItemDef > becomes pf_InFormItemDef.sas7bdat
 <PF:Candidate> becomes pf_candidate.sas7bdat

Figure 8 shows an important part of the SAS driver program for the clinical standard toolkit odm_read() macro.

```

PROC SQL;
insert into work.sasreferences
values ( "CST-FRAMEWORK" "1.2" "messages" "" "messages" "libref" "" 1 "" "" )
)
values ( "CDISC-ODM" "1.3.0" "messages" "" "odmmsg" "libref" "" 2 "" "" )
values ( "CDISC-ODM" "1.3.0" "autocall" "" "odmcode" "fileref" "" 1 "" "" )
values ("CDISC-ODM" "1.3.0" "sourcedata" "" "srcdata" "libref"
"&studyRootPath/data" . "" "" )
values ("CDISC-ODM" "1.3.0" "sourcemetadadata" "table" "srcmeta" "libref"
"&studyRootPath/metadadata" . "source_tables.sas7bdat" "" )
values ("CDISC-ODM" "1.3.0" "sourcemetadadata" "column" "srcmeta" "libref"
"&studyRootPath/metadadata" . "source_columns.sas7bdat" "" )
values ("CDISC-ODM" "1.3.0" "referencemetadadata" "table" "refmeta" "libref" "" .
"" "" )
values ("CDISC-ODM" "1.3.0" "referencemetadadata" "column" "refmeta" "libref" ""
. "" "" )
values ("CDISC-ODM" "1.3.0" "externalxml" "xml" "odmxml" "fileref"
"&studyRootPath/sourcexml" 1 "GetTransactons_Response.xml" "" )
values ("CDISC-ODM" "1.3.0" "referencexml" "map" "odmmap" "fileref"
"&studyRootPath/referencexml" 1 "odm.map" "" )
;
QUIT;

/* Allocate the SASReferences */
%cstutil_allocatesasreferences;

/* Parse the ODM XML file */

%odm_read();

```

Figure 8 ODM_READ SAS Macro

OUTPUT OF ODM READ

The outputs of SAS Clinical Standard Toolkit macros are SAS datasets that represent the information of ODM response XML in a normalized structure. Below, Figure 9 shows the mapping of important ODM XML elements from Figure 7 to their equivalent SAS datasets. Similarly, each ODM XML element from other SOAP methods mentioned in Table 2 can be converted into equivalent SAS datasets.

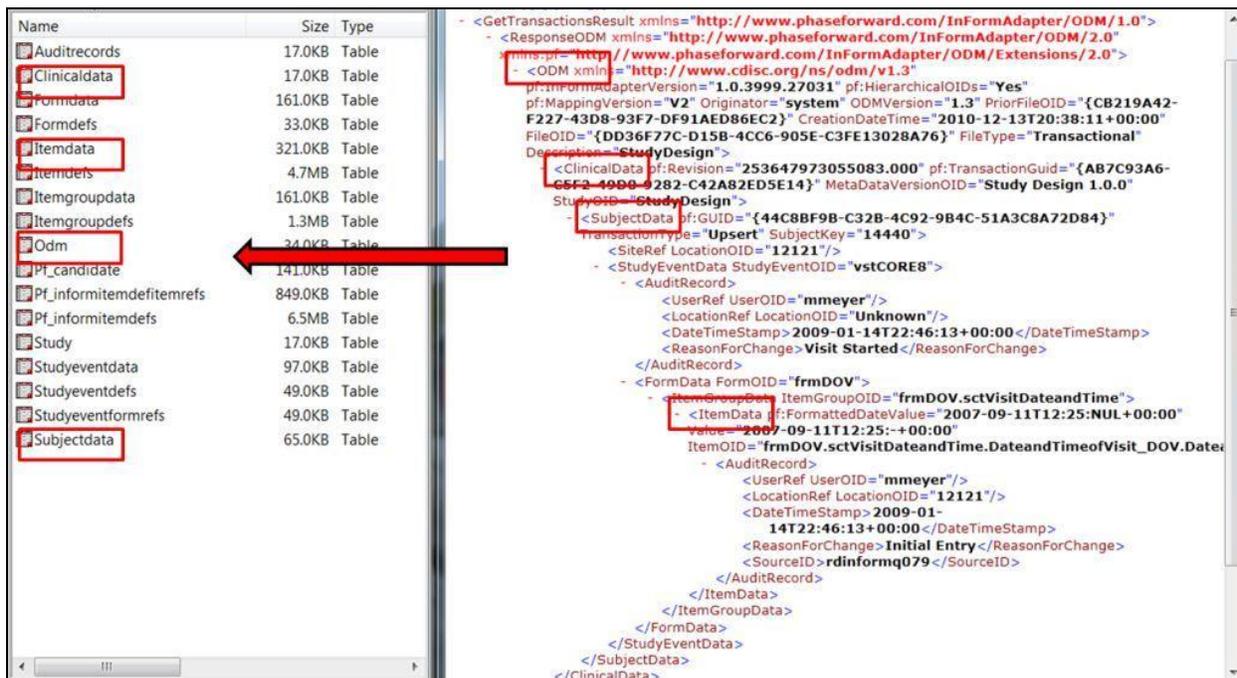


Figure 9 Output of ODM_READ

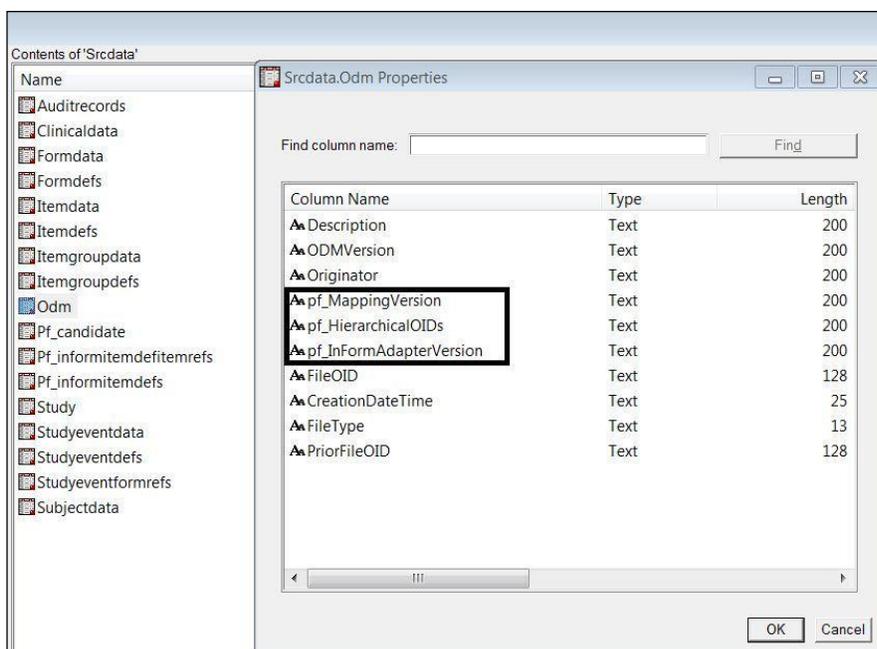


Figure 10 Columns of ODM SAS Dataset

After the SAS Clinical Standard Toolkit is modified, one can see the Phase Forward extensions present in ODM XML file in the SAS datasets.

Figure 10 shows the columns of an ODM SAS dataset that contains PF_ variables.

BUILD DOMAIN SAS DATASETS

The SAS datasets produced by the above ODM READ section are normalized in their structure. SAS Clinical Data Integration standard transformations are used to transform these normalized SAS datasets into domain datasets. Since all the required ODM information is present in the SAS datasets, one can also write customized SAS programs in SAS Clinical Data Integration to build their organization-specific data standard or CDISC CDASH standard.

The following check point requirements are considered for this paper before domain SAS datasets are created.

1. A call to SOAP method DownloadMetadaODM is made using the steps mentioned in the CONNECT TO INFORM section.
2. The output SAS datasets from ODM READ for DownloadMetadaODM are stored in a permanent SAS library (we will use “libname srcmeta” as an example).
3. A call to SOAP method GetTransactions is made using the steps mentioned in the CONNECT TO INFORM section.
4. The output SAS datasets from ODM READ for GetTransactions are stored in a permanent SAS library (we will use “libname srcdata” as an example).
5. As per ODM Standards: One SAS dataset for each <ItemGroupDefs>.
6. As per ODM Standards: All <ItemDefs> become the SAS variable name in respective SAS dataset.

Figure 11 is an example SAS code to create DM domain dataset.

```
/* Build DM Domain Dataset */

libname srcmeta "C:\demo\clinical_metadata";
libname srcdata "C:\demo\clinical_data";
libname domain "C:\demo\domain";

PROC SQL;
create table work.dm_1 as
select
d.name as sas_dataset_name,
b.name as sas_var_name,
b.datatype,
b.length,
a.value,
f.name as form_name,
h.name as study_event_name,
i.subjectkey,
i.transactiontype,
k.studyname,
k.studydescription,
k.protocolname,
l.description,
l.creationdatetime,
l.filetype

from srcdata.itemdata a
  inner join srcmeta.itemdefs b
    on a.itemoid = b.oid
  inner join srcdata.itemgroupdata c
    on a.fk_itemgroupdata = c.oid
  inner join srcmeta.itemgroupdefs d
    on c.itemgroupoid = d.oid
  inner join srcdata.formdata e
    on c.fk_formdata = e.oid
  inner join srcmeta.formdefs f
    on e.formoid = f.oid
  inner join srcdata.studyeventdata g
    on e.fk_studyeventdata = g.oid
  inner join srcmeta.studyeventdefs h
    on g.studyeventoid = h.oid
```

```

inner join srcdata.subjectdata i
    on g.fk_subjectdata = i.oid
inner join srcdata.clinicaldata j
    on i.fk_clinicaldata = j.oid
inner join srcmeta.study k
    on j.studyoid = k.oid
inner join srcdata.odm l
    on k.fk_odm = l.fileoid

where d.name = "DM";

QUIT;

PROC SORT data = work.dm_1;
    by sas_dataset_name description studyname protocolname subjectkey
transactiontype study_event_name form_name sas_var_name;
RUN;

PROC TRANSPOSE data = work.dm_1 out = domain.dm(drop=_name_ _label_
sas_dataset_name);
    by sas_dataset_name description studyname protocolname subjectkey
transactiontype study_event_name form_name;
    var value;
    id sas_var_name;
RUN;

```

Figure 11 SAS Code to create DM Domain SAS Dataset

The final output DM domain SAS dataset after executing above processes looks like Figure 12.

	StudyName	SubjectKey	TransactionType	study_event_name	form_name	AGE	AGEU	BRTHDAT	DMSUBJCD
1	Study123	20837	Upsert	Visit1	DM	28	YEARS	1984-01-23T---+00:00	M-L
4	Study123	20838	Upsert	Visit1	DM	34	YEARS	1978-04-17T---+00:00	A-B
6	Study123	20839	Upsert	Visit1	DM	31	YEARS	1980-07-13T---+00:00	DJ

	DMSUBJCD	ETHNIC	IEYN	RACE	SEX	SUBJID	VSORRES_WEIGHT	VSSTAT_HEIGHT	VSSTAT_WEIGHT
1	M-L	NOT REPORTED	Y	WHITE	M	121213	154.2	NOT DONE	DONE
4	A-B	NOT REPORTED	Y	WHITE	M	121213	154.2	NOT DONE	DONE
6	DJ	NOT REPORTED	Y	WHITE	M	121213	154.2	NOT DONE	DONE

Figure 12 DM Domain SAS Dataset

TransactionType variable from the above developed DM SAS dataset can be further used to program the Insert/Upsert/Deletion algorithm.

CONCLUSION

By using the framework and process described above, one can build the end-to-end integration modules in SAS Clinical Data Integration to transfer data from an InForm EDC system and convert it into SAS Datasets for analysis and reporting. This process can be automated by deploying and scheduling SAS Clinical Data Integration jobs at a required frequency to get the InForm data in real-time.

REFERENCES

- 1) www.cdisc.org for more information on the ODM standard
- 2) SAS Clinical Data Integration 2.2 User Guide from www.support.sas.com/documentation
- 3) PROC SOAP procedure from SAS 9.2 Help
- 4) InForm Adapter 1.3 SP4 Interfaces Guide from www.oracle.com
- 5) www.w3.org for more information on SOAP

ACKNOWLEDGEMENTS

I would like to thank the following individuals for their support and valuable input during the design and development of InForm integration with SAS Clinical Data Integration presented in this paper.

- Ben Bocchicchio and Sandeep Juneja from SAS Institute Inc, Cary, NC, USA
- Hans van Tricht and Gert Vant Hof from Novartis Vaccines and Diagnostics, Amsterdam, The Netherlands

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Jitesh Nagaria
Enterprise: SAS Institute Inc
Address: SAS Campus Drive,
City, State ZIP: Cary, North Carolina 27513
Work Phone: 919-531-5601
E-mail: Jitesh.nagaria@sas.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.