PhUSE 2010

Paper AD08

Using web service technologies for incremental, real-time data transfers from EDC to SAS

Andrew Newbigging, Medidata Solutions Worldwide, London, UK

ABSTRACT

Data collected in EDC systems is typically transferred to SAS via batch files, daily or less frequently. These files are typically cumulative containing all collected data, even when a small fraction has changed since the last transfer. This paper will describe how web services technologies can be used to provide incremental feeds of data in CDISC Operational Data Model (ODM) format to populate the SAS environment in near real-time. The use of the CDISC ODM standard provides a common, human-readable format that supports transactional updates, where only new, changed or removed data is transmitted. This drastically reduces the volume of data transmission and facilitates frequent transfers. Web services provide a mechanism for direct communication between the SAS environment and the EDC system, without intermediate files being required.

INTRODUCTION

Clinical research data is increasingly collected in Electronic Data Capture systems (EDC), which provide a web-based user interface for investigative site staff to enter clinical data onto electronic Case Report Forms (eCRF). EDC systems typically provide features to apply edit checks to the data which can automatically verify if the entered data meets the required parameters and if not, discrepancies are raised for subsequent correction or clarification by the site.

Once data has been cleaned, the EDC system must provide a mechanism for transferring the data into a statistical environment, such as SAS®. Such transfers are usually:

- cumulative; all data is included in each transfer,
- file-based; the EDC system exports the data to a file, which is transported to a location from where it can be read into the SAS environment,
- in a batch mode; at the end of a study, at intermediate points in the study, or sometimes on a daily basis,
- formatted in SAS transport format, or SAS dataset format.

This paper describes how transfers from EDC to SAS Clinical Data Integration (CDI) may be:

- incremental; only data that is new or updated needs to be transferred,
- web-service based; the EDC system presents a web service API that can be called directly from the SAS CDI environment,
- near real-time; as soon as data is available in the EDC system it can be transferred to SAS,
- formatted in the CDISC Operational Data Model (ODM) standard.

DATA TRANSFER

INCREMENTAL

The primary advantage of incremental data transfers over cumulative data transfers is efficiency. As a clinical research study progresses new data is collected and updates are made to existing values. As the total amount of data collected in the study grows, the volume of unchanged data rapidly dwarfs the volume of changed data during each transfer period. This is particularly true for long-running studies.

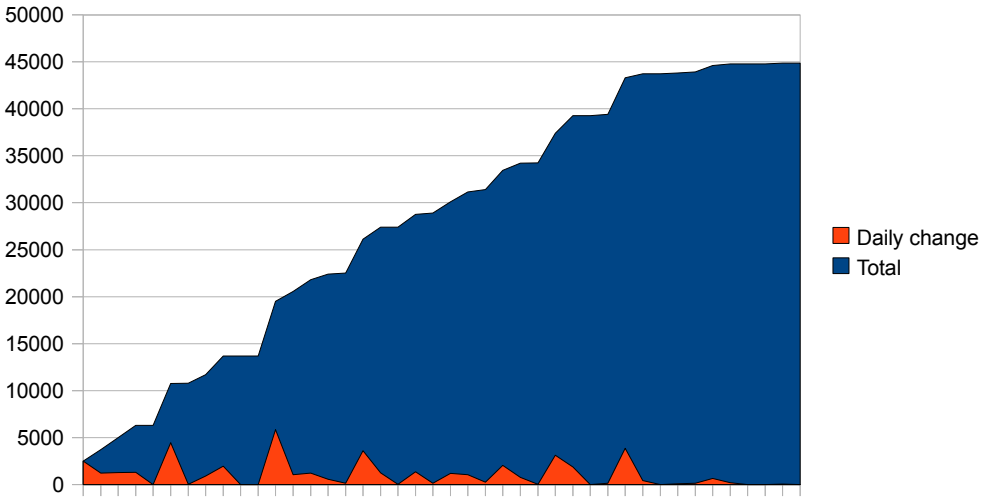


Illustration 1: Comparison of data changed per day against total data collected for a single study

An analysis of 269 EDC databases, containing 2.2 billion datapoints, shows that over a 1 year period, the proportion of data changed each day averages 2.8 million datapoints, which is 0.12% of the total. Excluding completed, inactive studies, the daily percentage change of the total is 1.8%. If cumulative data transfers are used, almost all of the effort in retrieving data from the EDC database, formattting it and writing it to a file, transferring the file and then loading the file into SAS, is wasted since the data values are largely unchanged.

Incremental transfers are clearly more efficient, and are necessary to support near real-time transfer; transferring and processing cumulative datasets cannot be quick enough to achieve real-time transfers.

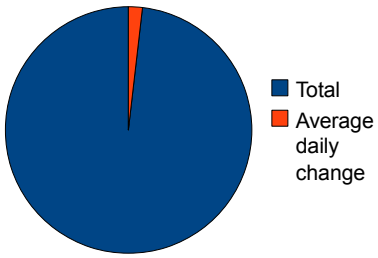


Illustration 2: Average daily change against total data volume

WEB SERVICES

Current technologies for data transfer from EDC to SAS are normally based on file transfers:

- on a scheduled basis, or on user request ('on-demand') the EDC system extracts data from its database and writes the data to a file
- the file is moved to a location such as a secure File Transfer Protocol (sFTP) server
- a second scheduled process detects that a new file has been received
- the new file is loaded into the SAS environment

File transfers are a traditional, well-known, method, but have some drawbacks:

- to achieve near real-time data transfers, a file transfer method becomes problematic. As the frequency of transfer increases it becomes likely that the disconnected file transfer model, with delays in each step of the process will fail to keep pace with the desired frequency,
- there are several components in the process, with consequent higher risk of failure should any one component be unavailable,
- files are prone to data corruption, either through incomplete files being generated or received, or through file content corruption. Additional steps, including crytographic hash functions such as md5 and sha512 [1], can be used to verify and guarantee file integrity, but these are rarely used in clinical research studies.

Web services are commonly used to connect modern software applications, and can be effectively applied to clinical data transfers. The term 'web services' is used loosely to apply to a variety of technologies, including protocols such as the Simple Object Access Protocol (SOAP)[2]. We do not use SOAP, preferring the style referred to as Representational State Transfer (REST)[3] whereby the available datasets are accessed through Uniform Resource Identifiers (URI) [4], familiar to all users of web browsers.

For example, the Adverse Event dataset for a clinical study called Mediflex would be located at an address such as the following:

https://innovate.mdsol.com/RaveWebServices/studies/Mediflex/datasets/regular/AE

The dataset can be retrieved by issuing an HTTP GET request – the HTTP 'GET' method is a request to return the contents of the resource identified by the URL. When using a web browser (Internet Explorer, Firefox, etc) for normal access to web pages, the web browser issues HTTP GET requests to display a web page and its associated content such as images.

The web service API will reply with an HTTP status code [5] indicating whether the request has been successful. Common status codes are:

HTTP status code	Description
200	OK
401	Unauthorized
404	Not Found

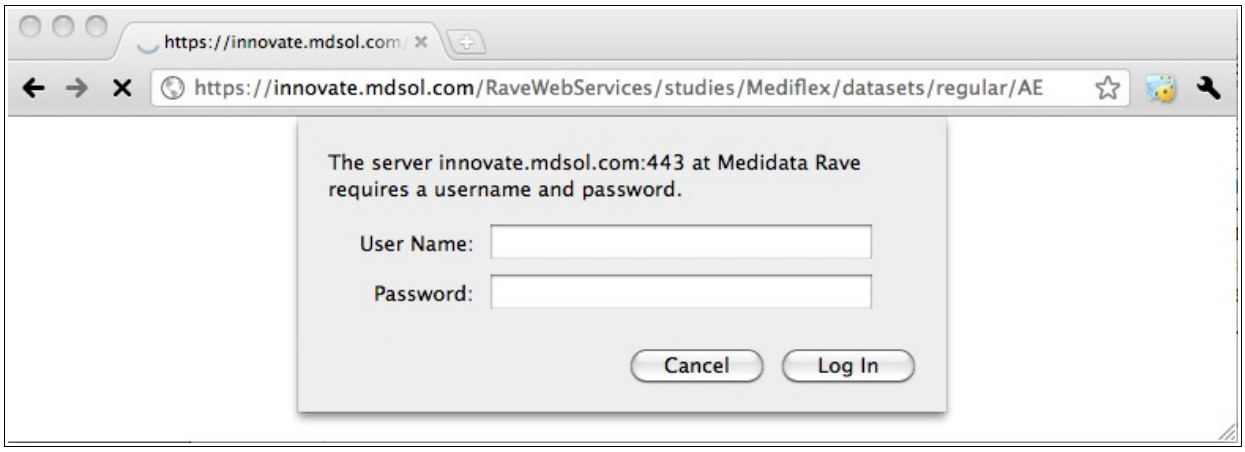
A request to a non-existent dataset or study will return a '404 Not Found' status code.

Access to the resource, in this case the clinical dataset, is restricted by requiring authentication details to be provided with the HTTP GET request. One of the simplest methods is HTTP Basic Authentication [6], where a username and password is supplied and used to authenticate the request before providing the clinical dataset. HTTP Basic Authentication should only be used with encrypted requests; typically Secure Socket Layer (SSL) encryption is used.

Access to the web service API can be seen through a transcript of a web browser session:

1. The URL is entered into the browser navigation bar:

2. The web services API responds with a request for authentication details and the browser displays a dialog window for entry of the username and password:



3. On entry of valid credentials, the web service API responds with a '200 OK' status code, and the dataset contents:

Details of the HTTP request/response messages between the client application and the web service API can also be seen using the cURL application[7] (a useful utility for testing and working with web service APIs of all kinds):

curl -u username:password -H 'Content-Type:text/xml' -v https://innovate.mdsol.com/RaveWebServices/studies/Mediflex/datasets/regular/AE	Initial request
* About to connect() to innovate.mdsol.com port 443 (#0) * Trying 70.42.99.224... connected * Connected to innovate.mdsol.com (70.42.99.224) port 443 (#0) * SSLv3, TLS handshake, Client hello (1): * SSLv3, TLS handshake, Server hello (2): * SSLv3, TLS handshake, CERT (11): * SSLv3, TLS handshake, Server finished (14): * SSLv3, TLS handshake, Client key exchange (16): * SSLv3, TLS change cipher, Client hello (1): * SSLv3, TLS handshake, Finished (20): * SSLv3, TLS change cipher, Client hello (1): * SSLv3, TLS handshake, Finished (20): * SSL connection using RC4-MD5 * Server certificate: * subject: O=*.mdsol.com; OU=Domain Control Validated; CN=*.mdsol.com * start date: 2007-03-28 17:49:39 GMT * expire date: 2017-04-03 14:34:46 GMT * subjectAltName: innovate.mdsol.com matched * issuer: C=US; ST=Arizona; L=Scottsdale; O=GoDaddy.com, Inc.; OU=http://certificates.godaddy.com/repository; CN=Go Daddy Secure Certification Authority; serialNumber=07969287 * SSL certificate verify ok.	SSL connection initiated SSL certificates verified (ie. The server has been verified as genuine)
* Server auth using Basic with user 'username' > GET /RaveWebServices/studies/Mediflex/datasets/regular/AE HTTP/1.1 > Authorization: Basic ********** > User-Agent: curl/7.19.7 (universal-apple-darwin10.0) libcurl/7.19.7 OpenSSL/0.9.8l zlib/1.2.3 > Host: innovate.mdsol.com > Accept: */* > Content-Type:text/xml >	HTTP Basic Authentication (private information has been obscured in this transcript)
< HTTP/1.1 200 OK < Date: Tue, 14 Sep 2010 01:18:05 GMT < Content-Type: text/xml <	200 OK response from web service API

<?xml version="1.0" encoding="utf-8"?>
<ODM
  FileType="Snapshot"
  FileOID="96741552-97f4-4035-aad3-e9f12459ca20"
  CreationDateTime="2010-09-14T01:18:05.255-00:00"
  ODMVersion="1.3"
  xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns="http://www.cdisc.org/ns/odm/v1.3"
>
	Dataset content

… … …	Further content omitted from this transcript

cURL example of 404 Not Found response:

`curl -u username:password -H 'Content-Type:text/xml' -v https://innovate.mdsol.com/RaveWebServices/studies/Mediflex/datasets/regular/A`	Initial request

`* About to connect() to innovate.mdsol.com port 443 (#0)`
`*   Trying 70.42.99.224... connected`
`* Connected to innovate.mdsol.com (70.42.99.224) port 443 (#0)`
`* SSLv3, TLS handshake, Client hello (1):`
`* SSLv3, TLS handshake, Server hello (2):`
`* SSLv3, TLS handshake, CERT (11):`
`* SSLv3, TLS handshake, Server finished (14):`
`* SSLv3, TLS handshake, Client key exchange (16):`
`* SSLv3, TLS change cipher, Client hello (1):`
`* SSLv3, TLS handshake, Finished (20):`
`* SSLv3, TLS change cipher, Client hello (1):`
`* SSLv3, TLS handshake, Finished (20):`
`* SSL connection using RC4-MD5`
	SSL connection initiated

`* Server certificate:`
`*  subject: O=*.mdsol.com; OU=Domain Control Validated; CN=*.mdsol.com`
`*  start date: 2007-03-28 17:49:39 GMT`
`*  expire date: 2017-04-03 14:34:46 GMT`
`*  subjectAltName: innovate.mdsol.com matched`
`*  issuer: C=US; ST=Arizona; L=Scottsdale; O=GoDaddy.com, Inc.; OU=http://certificates.godaddy.com/repository; CN=Go Daddy Secure Certification Authority; serialNumber=07969287`
`*  SSL certificate verify ok.`
	SSL certificates verified (ie. The server has been verified as genuine)

`* Server auth using Basic with user 'username'`
`> GET /RaveWebServices/studies/Mediflex/datasets/regular/A HTTP/1.1`
`> Authorization: Basic **********`
`> User-Agent: curl/7.19.7 (universal-apple-darwin10.0) libcurl/7.19.7 OpenSSL/0.9.8l zlib/1.2.3`
`> Host: innovate.mdsol.com`
`> Accept: */*`
`> Content-Type:text/xml`
`>`
	HTTP Basic Authentication

`< HTTP/1.1 404 Not Found`
`< Date: Tue, 14 Sep 2010 01:35:10 GMT`
`< Content-Type: text/xml; charset=utf-8`
`< Content-Length: 377`
`<`
	404 Not Found response from web service API. Dataset 'A' does not exist in this study.

`<?xml version="1.0" encoding="utf-8"?>`
`<ODM`
`  FileType="Snapshot"`
`  FileOID="ec794e12-7e57-405b-bc2a-813df3fefe34"`
`  CreationDateTime="2010-09-14T01:35:10.480-00:00"`
`  ODMVersion="1.3"`
`  xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata"`
`  xmlns:xlink="http://www.w3.org/1999/xlink"`
`  mdsol:ErrorDescription="Dataset does not exist [RWS00134]"`
`  xmlns="http://www.cdisc.org/ns/odm/v1.3"`
`/>`
	Body contains details of error. Note use of CDISC ODM vendor extension to provide detailed error message

NEAR REAL-TIME

To achieve near real-time data transfer, requests are made to the web service API on a regular, frequent schedule – for example once every 5 minutes. The API responds with those datapoints that have been added, updated, or removed (soft-deleted) since the previous request. A filter parameter is added to the URI to specify a timepoint from which changes should be measured:

https://innovate.mdsol.com/RaveWebServices/studies/Mediflex/datasets/regular/AE?start=2010-09-01T15:00:00

The above example will return changes occurring on or after 15:00 hours on the 1st September 2010. The timestamp is formatted in ISO 8601 format[8], and is in Co-ordinated Universal Time (UTC)[9]. For simplicity and clarity only UTC is supported – no other timezones may be specified. Any data value that was created after this timestamp will be returned as an 'Insert' transaction in the dataset. Any data value that was created before the timestamp and modified after will be returned as an 'Update'. Any data value that was created before the timestamp and then soft-deleted after will be returned as a 'Remove'.

The next request to the API will be at the next timestamp, in this case 5 minutes later:

https://innovate.mdsol.com/RaveWebServices/studies/Mediflex/datasets/regular/AE?start=2010-09-01T15:05:00

If any request does not succeed, for example because of a network failure, then the same timepoint should be used in the next request until a successful result is returned.

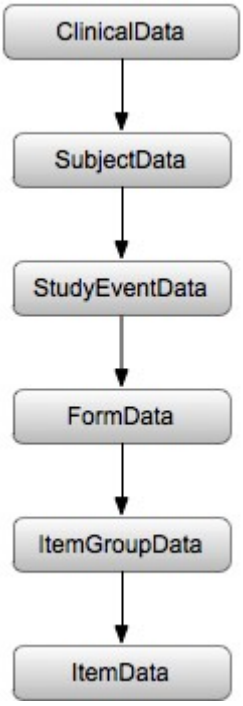
CDISC ODM

The format for the dataset content should meet the following requirements:

1. it should be flexible enough to contain datasets for any study design and form design commonly used in clinical studies,
2. it should be human-readable and self-describing, i.e. the contents of a dataset should be self-contained and not reliant on 'magic' unspecified knowledge about the format to decode the contents
3. it should support incremental transfers, i.e. by identifying inserted, updated or removed data,
4. it should not be a closed, proprietary format, and should be a standard if one is available.

SAS transport files[10], ASCII files and the CDISC Study Data Tabulation Model (SDTM)[11] all fail on one or more of these requirements. The CDISC Operational Data Model (ODM)[12] satisfies all of these requirements and is also extensible, so that future requirements can be added to the model if necessary through the use of the 'vendor extension' mechanism.

The ODM clinical data section has a consistent XML tree hierarchy to describe clinical data:



```
<ODM xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata"
     xmlns:xlink="http://www.w3.org/1999/xlink"
     xmlns="http://www.cdisc.org/ns/odm/v1.3"
     FileType="Snapshot"
     FileOID="c77ccb54-dc13-44d8-8966-e4e763453b73"
     CreationDateTime="2010-09-06T13:45:06.656-00:00"
     ODMVersion="1.3">
    <ClinicalData StudyOID="Mediflex" MetaDataVersionOID="23">
        <SubjectData SubjectKey="123 ATN">
            <SiteRef LocationOID="MDSOL"/>
            <StudyEventData StudyEventOID="SUBJECT">
                <FormData FormOID="AE" FormRepeatKey="1">
                    <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="1" TransactionType="Insert">
                        <ItemData ItemOID="AE.AEYN" Value="Y"/>
                        <ItemData ItemOID="AE.AETERM" Value="HEADACHE"/>
                        <ItemData ItemOID="AE.AESTDTC" Value="2010-01-01"/>
                        <ItemData ItemOID="AE.AEONG" Value="N"/>
                        <ItemData ItemOID="AE.AEENDTC" Value="2010-01-01"/>
                    </ItemGroupData>
                </FormData>
            </StudyEventData>
        </SubjectData>
    </ClinicalData>
</ODM>
```

Illustration 3: Example ODM clinical data

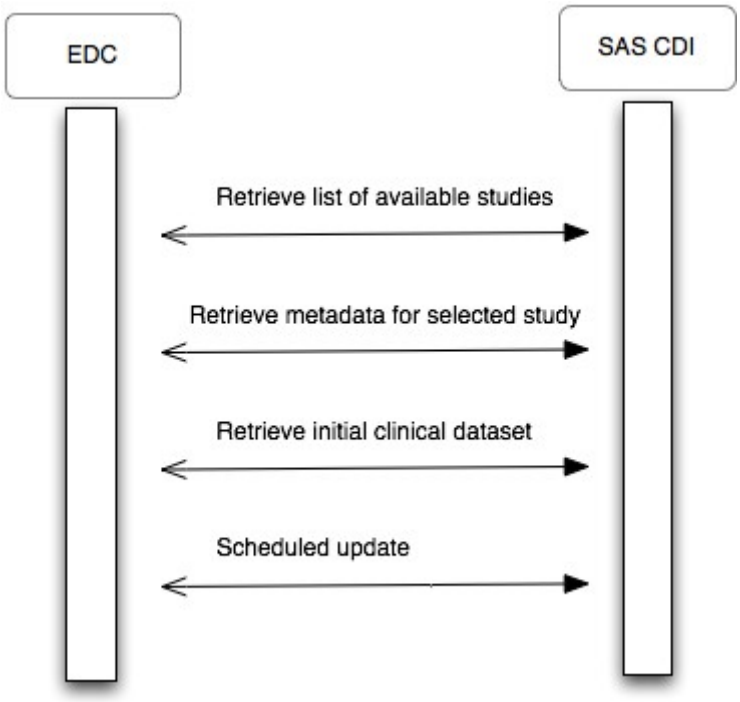
The ODM can also be used to describe the metadata associated with clinical data, ie how the forms and questions are structured and their attributes.

```
  <ItemDef OID="AE.AETERM" Name="AETERM" DataType="text" Length="200">
    <Question>
      <TranslatedText xml:lang="en">Event Description: </TranslatedTex
    </Question>
  </ItemDef>
  <ItemDef OID="AE.AESTDTC" Name="AESTDTC" DataType="date">
    <Question>
      <TranslatedText xml:lang="en">Start Date:</TranslatedText>
    </Question>
  </ItemDef>
  <ItemDef OID="AE.AEONG" Name="AEONG" DataType="text" Length="1">
    <Question>
      <TranslatedText xml:lang="en">Ongoing?</TranslatedText>
    </Question>
    <CodeListRef CodeListOID="YESNO" />
  </ItemDef>
  <ItemDef OID="AE.AEENDTC" Name="AEENDTC" DataType="date">
    <Question>
      <TranslatedText xml:lang="en">End Date:</TranslatedText>
    </Question>
  </ItemDef>
```

Illustration 4: Example ODM metadata

WORKFLOW

Using the data transfer characteristics defined in the preceding section, a typical workflow for transferring data from EDC to SAS using web services is:



The list of studies is available via a RESTful web service method, returning the list in ODM format:

https://innovate.mdsol.com/RaveWebServices/studies

The metadata for a study is also available via a RESTful web service method:

https://innovate.mdsol.com/RaveWebServices/metadata/studies/Mediflex/versions/1

CHALLENGES

CONSISTENCY

Incremental data transfers provide greater efficiency but there is a potential risk that data in the EDC system may not be entirely transferred to the SAS environment, particularly if there is an error during a transfer, such as a network failure. Recovery from errors is possible, by requesting increments from a known valid point in time, but the question remains as to how consistency can be verified. This problem remains under investigation – possible approaches include:

1. occasionally using a full data transfer to verify that all data has been received. This however negates the benefit of incremental transfers, and could not be performed frequently without the same drawbacks as using full transfers all the time.
2. Using an algorithmic hash function, such as md5, sha1 or sha512, to compare the data in each system, without requiring entire datasets to be transferred for the comparison.

METADATA VERSIONS

CDISC ODM provides for different versions of the study metadata to be associated with different subjects, so that, for example, subject A may be associated with metadata version 1 and subject B may be associated with metadata version 2. This flexibility is needed to cope with situations such as protocol amendments, which may be introduced to different investigative sites at different times, and modifications to the electronic Case Report Form during the course of the study.

There are no constraints on the differences between metadata versions in CDISC ODM; two versions of the same study may contain different forms and questions, and forms and questions with different definitions, such as field length, or differences in code lists. Differences in field lengths may be accommodated by taking a 'lowest common denominator' approach – if a field is defined as 'text' in one version and 'integer' in another, then the lowest common denominator is 'text', and the field must be treated as a text field for analysis. Differences in code lists, new or removed codes and changed descriptions are particularly problematic and still under investigation.

CONCLUSION

ODM-based web services provide a new method for transferring clinical data efficiently and frequently from EDC systems to SAS. The use of such web services enables incremental transfers, directly from SAS Clinical Data Integration, in near real-time.

CONTACT DETAILS

Andrew Newbigging
Medidata Solutions Worldwide
Harman House
1 George Street
Uxbridge
UB8 1QQ
UK

email: anewbigging@mdsol.com

REFERENCES

1 http://en.wikipedia.org/wiki/Cryptographic_hash_function
2 http://www.w3.org/TR/soap/
3 http://en.wikipedia.org/wiki/Representational_State_Transfer
4 http://en.wikipedia.org/wiki/Uniform_Resource_Identifier
5 http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html
6 http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.8
7 http://curl.haxx.se/
8 http://www.w3.org/TR/NOTE-datetime
9 http://en.wikipedia.org/wiki/Coordinated_Universal_Time
10 http://support.sas.com/techsup/technote/ts140.html
11 http://www.cdisc.org/sdtm
12 http://www.cdisc.org/odm