

Clinical Data Review on iPad® with DEFINE.XML

Sy Truong, Meta-Xceed, Inc. (MXI), Fremont, CA

ABSTRACT

Clinical data review on a tablet computer is a natural step in the evolution of automated information delivery. Clinical Research Associates and Medical Monitors require a unique method of accessing information during the conduct of a clinical trial and also at the pivotal conclusion, or submission of a trial. The challenge remains to overcome the technical complexity of metadata models such as DEFINE.XML. This paper describes these challenges as it elaborates upon methods of using SAS to deliver clinical data in a format that can be easily navigated on an iPad. Some of the methodologies described include:

- Capturing data with standardized metadata of DEFINE.XML
- Navigating to specific clinical domains with standardized data structures with multi-touch
- Searching for subject data such as Adverse Events on iPad
- Viewing clinical data while mobile and offline
- Zooming in on images such as CRF or graphs with pinch gestures



These technologies are new and will pave ways for tools in this "post-PC World". However, data review of clinical information remain as old as the drug development process.

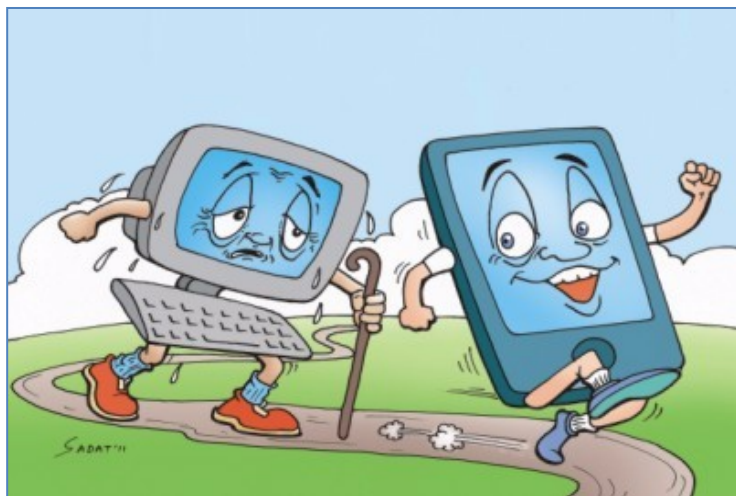
POST-PC ERA FOR CLINICAL

Just a few months ago when Steve Jobs announced that we are entering a "Post-PC" era at the iPad 2 launch event, I have been reflecting upon what that means for us working within SAS and clinical data. The iPad has since then grown in usage along with its related iOS devices including iPhone and iPod touch. Many other tablet devices from Google's Android operating system have also been introduced. This mobile environment with its multi-touch interface is also making inroads into laptops and desktop operating systems such as how Mac OSX is adopting the multi-touch gestures. The clinical data environment and its adoption of technologies is relatively conservative and slow to change due to its regulatory requirements and how it directly affect on the health of real patients. It therefore can take time before the latest innovations is applied. However, the ability to review clinical data in a portable mobile environment has been what clinical professionals have been requesting for a long time.

Forrester Research has evaluated the maturing market place of tablet computers and has come to some interesting conclusions. These findings explain the rapid adoption of tablets like the iPad and its effects on the traditional PC market. This also explains how it will change the way we use clinical data. Some of the points are described here:

- **Stationary to Ubiquitous** – The research found that the traditional stationary desktop PC that is used when the user is at their desk is transitioning to mobile computing and used at any time. A Clinical Research Associates and Clinicians are not always at their desk and are on the move requiring information to be mobile. The ability to access critical information at anytime and anywhere is one of the main ways how clinical data is accessed.
- **Abstract to Immediate** – The report also shows how the PC needed a mouse interface which is an abstraction to how users point to objects on a PC screen. It requires a boot up time forcing users to wait. This has been replaced by a touch screen which is closer to how users interact with objects in the physical world and it can startup instantly. This is a step towards making access to clinical data more immediate and approachable. Clinical professionals by training are not computer scientists, thus are hindered by cumbersome computers and related database software technologies. This barrier is beginning to dissipate

as we enter the post-PC era where there are less abstractions and hindrances for a clinical scientist to get to his or her data via a tablet device.



The research is showing how mobile computing can deliver the form factor that is analogous to a clipboard or thin notebook which are commonly used in the clinical environment. This paper describes how tablets are used with clinical data as shown through a Clinical Reviewer App which takes advantage of these “Ubiquitous” and “Immediate” attributes to describe how the move towards “post-PC” era can significantly benefit how we use clinical data.

PREPARING MOBILE CLINICAL DATA

Clinical data is usually centrally managed on a server and can only be accessed by users who are logged onto the system. Although this provides a good layer of security, it can prevent some essential users, such as Medial


Monitors or Clinical Research Associates (CRAs), from viewing the data on a mobile device such as on an iPad. The data can be copied from the server and placed onto an iPad, but additional metadata needs to be captured before the information can be useful. The types of metadata captured for this process is similar to what is done when generating a DEFINE.XML. The data definition of DEFINE.XML was designed for clinical data to be placed within an electronic submission and used in a FDA submission. This type of information is also useful when delivering data on a tablet computer. Some of the key components of DEFINE.XML that makes it suitable for a mobile review application are listed below.

1. **CDISC Standard** – The metadata or information about the datasets that is going to be placed on the iPad needs to be in a standard format. This allows a single algorithm to be used to port the data from a server to an iPad. CDISC is the logic choice since it is the premier industry standard for clinical data.
2. **Organization Metadata** – Information that is normally documented in a clinical protocol containing the name of the study or protocol, drug molecule number and sponsor or company name. If there are many clinical studies loaded onto the iPad, this information will help navigate the user to the specific protocol of interest.
3. **Dataset Metadata** – All the data within a clinical protocol contains a name and label. If CDISC standards were applied, the short would be a two letter name such as AE for adverse event.
4. **Variable Metadata** – Each variable within a dataset will have attributes detailing the type of variable, length and descriptive label. This is similar to what you would see in a PROC CONTENTS output.
5. **Controlled Terminology** – If values of each variable are coded such as “M” for “Male” and “F” for “Female”, a list of these codes, referred to as controlled terms, are captured and referenced when viewing on an iPad.
6. **Value Level Metadata** – The controlled terminology is one example of a value level metadata providing more information about the values stored in a particular variable. Optionally, there could be additional information explaining the meaning of each variable value.

The goal is to capture all the metadata and store them in a separate SAS dataset before the clinical data can be ported to the iPad. This process of capturing the metadata is similar to what is done before generating a DEFINE.XML. The following steps describe the process and how it is accomplished. This is done on the server prior to having the data copied to the iPad. Some steps are automated with a graphical user interface, while also accompanied by SAS macros as an alternative way of performing the tasks in batch mode.

STEP 1 – Capture Organization Metadata

Organization information such as the protocol name and number needs to be captured and inserted in a SAS dataset. The following graphical user interface screen is designed to acquire this.


General Information

Output Library:

Company Name:

Product Name:

Protocol:

XPT Link Location: (*)
(i.e. ../xptfile or ../)

CRF Link Location: File Folder (*)

Split Char:

Layout

Left Title: Center Title: Right Title:

Left Footnote: Center Footnote: Right Footnote:

* NOTE: This is relative to the output path such as: C:\temp.

In addition to names of the protocol and drug information, relative path locations to related files are also captured. All files captured will be relative to the data definition root path. If it is stored in the same location, the path “.” is specified. This makes it more portable so that when it is transferred to the iPad, no explicit full paths to the server are referenced. The layout information is optional for reports generated. For some reports, information such as the protocol name will be inserted at the top title or bottom footnote of the report.

STEP 2 – Capture Data and Variable Metadata

After we know where the data is going to be used, the next step is to capture all the dataset and variable level metadata. Some of this information already exists within the data and can be captured through the use of PROC CONTENTS. The following code segment illustrates how this is captured.

```

*** Capture the metadata of specified dataset ***;
proc contents data = &libdat out= work.&shortcutname noprint;
run;

proc sort data = work.&shortcutname;
  by varnum;
run;

*** Add additional metadata used for DEFINE.XML ***;
data work.&shortcutname(keep= variable label length format
  fmtcode origins role key sortedby comment typen path labname libname datname
  location keys numvar numobs struct purpose class datcom valuelev mdatory);
  attrib variable label = "Variable Name";
  attrib label label = "Variable Label";
  attrib length label = "Length";
  attrib format label = "Format" length=$6000;
  attrib formatcode label = "Decode Formats" length=$6000;
  attrib origins label = "Origins" length=$1024;

```

```

attrib role label = "Role" length=$1024;
attrib key label = "Key" length=$1024;
attrib comment label = "Comment" length=$5000;

attrib path label="Library Path" length=$1024;
attrib labname label="Dataset Label" length = $1024;
attrib libname label="Library Name" length = $200;
attrib datname label="Dataset Name" length = $1024;
attrib struct label="Structure" length=$1024;
attrib purpose label="Purpose" length=$1024;
attrib class label="Class" length=$1024;
attrib location label="Location" length = $1024;
attrib keys label="Keys" length = $1024;
attrib numvar label="Number of Variables" length=8;
attrib numobs label="Number of Records" length=8;
attrib datcom label = "Dataset Level Comment" length=$5000;
attrib valuelev label = "Value Level Metadata" length=$200;
attrib mdatory label = "Mandatory" length = $50;

set work.&shortcutname;
name = upcase(name);
purpose="Tabulation";
struct="";
class="";

rename name = variable;
rename formatcode=fmtcode;
rename type = typen;

*** Insert General Organization Information ***;
path = "&curlibpath";
labname = "&datlabel";
libname = "&curlib";
datname = "&curdat";

location = "&curlocation";
keys = "&keys";
numvar = &numvar;
numobs = &numobs;
datcom = "%nrbquote(&datlabel)";
valuelev="";
mdatory="";
run;

```

In addition to information such as variable name, label and length, information about the structure of the data or variable requirement is also captured. The additional information is appended to the initial set of metadata captured by PROC CONTENTS. The combined set of information is stored in a new dataset so that it is easily manipulated and reports can be generated suitable for viewing on an iPad.

STEP 3 – Capture Controlled Terms and Value Level Metadata

If the data comes from a relational database such as Oracle, it would be exported into SAS since there are many tools within SAS that are useful for this process. An example is how SAS handles controlled terminologies. SAS datasets are designed to store related controlled terms in a separate format catalog. The goal of presenting the data on an iPad in a meaningful way is to process the format catalog and decode any controlled terms that are associated with each variable. This is accomplished by transforming the terms stored in the catalog into a separate SAS dataset to make reporting of the values much easier.

```

*** Capture the formats from the catalog
dataset ***;
options fmtsearch = (&fmtlib);
proc format library=&fmtlib
    cntlout=work.tempfmt;
run;

```

This is efficiently applied through a PROC FORMAT which converts information from the format catalog referenced in the macro variable FMTLIB and creates an output WORK.TMPFMT dataset. This works well but it also generates extra information that may not be useful for this particular use.

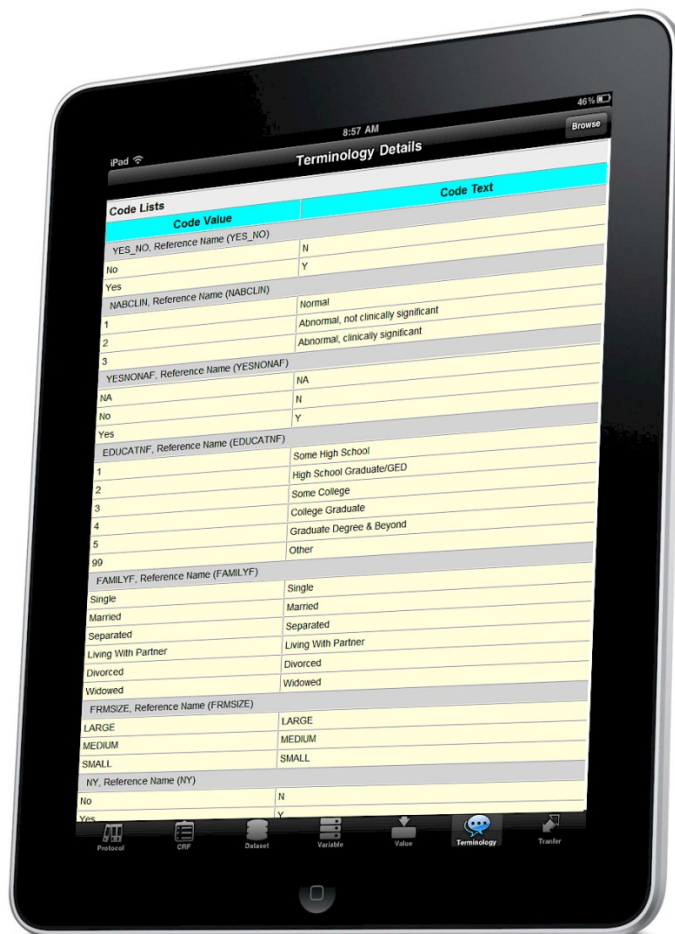
```

*** Hold on to only the variables
needed for iPad Conversion ***;
data work.tempfmt1(keep = start end
label type hlo);
    set work.tempfmt;
    if lowercase(fmtname) = "&temp";

    ...
    *** Combine the coded and
    decoded values ***;
    if (lowercase(start) eq
lowercase(end)) then do;
        code=start || "=" || label;
    end;
    else do ;
        code=start || "-" || end ||
"=" || label;
    end;
    controlterms = controlterms ||
' ' || code;

run;

```



In this case, each format is processed within a dataset with columns such as START and END representing the coded and decoded values. This is combined into a final variable named CONTROLTERMS which is used in the final documentation on the iPad.

PROCESSING CASE REPORT FORMS

The processing of metadata upon SAS datasets can be accomplished entirely within SAS. However, when it comes to manipulating actual images of the Case Report Forms (CRF), this requires tools that are external to SAS. In this case, ImageMagick® was used due to its scripting capabilities. The goal is to be able to have SAS perform a call through an X command or a SYSTASK to the operating system to perform these tasks. Since SAS has access to all the metadata captured, this is an efficient way to have SAS programs generate the scripts used to manipulate the CRFs before porting to the iPad. A standard file format that CRFs are stored in is a PDF file. PDF is a good format for desktop computers. It is based upon Postscript which is page oriented language designed for printers. Because of this, it is not optimal for mobile devices. The iPad can view PDF files, but in the case where there are hundreds of pages, the PDF become large and opening them become clunky and slow. Thus, it is recommended that the PDF be converted to a more suitable format such as PNG. The following command is an example of how the PDF is converted to PNG file format.

Subject Number	Subject Initials	Visit	Page
Site Number Rand. Number	F M L	<u>1</u>	1

DEMOGRAPHICS

Date of Birth: ___/___/___
 DD MMM YYYY

Gender: Male
 Female

Ethnicity: Not Hispanic or Latino
 Hispanic or Latino

Race: American Indian or Alaska Native Caucasian or White
 Asian Native Hawaiian or Other Pacific Islander
 African American or Black Other, specify: _____

CONSENT

Date Informed Consent Signed: ___/___/___
 DD MMM YYYY

Screening Number: S _____

PREGNANCY TEST

Not Applicable (Male)

If Female:

Is the subject a female of child bearing potential? No Yes*

*If yes:

Is the subject practicing a medically acceptable method of contraception? No Yes

Result of pregnancy test: Positive Negative Not Done

Date of pregnancy test: ___/___/___
 DD MMM YYYY

07JUNE2011

```
*** Convert the CRF in PDF to PNG
format ***;
systask command 'convert "&pdfcrf"
"&pngcrf"' WAIT;
```

The command “convert” is provided by ImageMagick and can be invoked directly within a SAS program though a SYSTASK command. This process will convert each PDF page into a separate PNG file so it can be optimally viewed on the iPad. In addition to having the ability to perform a straight conversion, there are many options to control the size and quality of the resulting PNG image file. This is crucial in optimizing the PNG file for display on the iPad screen. Here is an updated command to create a thumbnail view of the same CRF.

```
*** Convert the CRF in PDF to a
thumbnail view PNG format ***;
systask command 'convert -density
400 "&pdfcrf" -resize 25% "&pngcrf"'
WAIT;
```

In this example, two additional options are applied controlling the quality of the final image including the “density” option along with the “resize”, which shrinks it down to the correct size for the thumbnail view. These thumbnails are then used in the navigational list-view as shown on the left panel of the landscape view.

In this example, the conversion from the original PDF file to separate PNG files has been applied multiple times. The initial conversion transforms the PDF to individual PNG files which are displayed on the right panel when the iPad is viewed in landscape view. The user can pinch to zoom in on this image to view a specific area within the CRF. The navigational panel on the left presents both a thumbnail view along with the name of the CRF, so the user can efficiently flip to the specific CRF of interest. The layout of a CRF can differ significantly. For example, the CRF for demographic information is in portrait and captures one record for each subject. On the other hand, the adverse event case report form is in landscape and captures multiple observations per subject creating a distinctive table grid layout. Thus, the use of thumbnails as a way of visually identifying each CRF for the purpose of navigation can be more effective than just using a list of names.



DISPLAY CLINICAL DATA ON IPAD

The preparation of the clinical data on the server is primarily accomplished using SAS. However, once the data is transferred to the iPad, the software used to view the information is developed in Objective C programming language provided from Apple. The following steps describe how the data is process in the App running on the iPad.

STEP 1 – Transfer Data to iPad

After information is processed on the SAS server, the next step is to copy this over to the iPad. The iOS operating system used with the iPad supports two main methods for transferring files from the desktop or server to the iPad.

1. **Wireless Cloud** – The iPad can be connected to the Internet (the cloud) using either Wi-Fi or 3G data service. You can install an App from the AppStore such as DropBox to allow easy file drag and drop from your desktop computer to the iPad. Apple is also going to release its own service named iCloud which will have these features. This process is convenience in that you do not need to be physically connected to any machine and the files can be updated to the iPad even if you are traveling at a clinical site and are not near your desktop.

2. **iTunes Transfer** – When the iPad is connected to your desktop computer and running iTunes, you can navigate to the file sharing folder pertaining to your application. Once this folder has been selected, you can drag and drop files to the Clinical Reviewer App folder within iTunes. The files will then be transferred when the iPad is synced between the iTunes library and the mobile device. This is useful when you do not have network connection and can be more secure since the transfer is performed directly from the desktop computer to the iPad through a physical USB cable.



In both methods, the steps of transferring data from the server onto the iPad need to be performed repeatedly since the clinical data is periodically updated as the study is being conducted. Once the data is downloaded to the iPad, you do not need any connection to the server, or even to the Internet. The review of the clinical data will be local to the iPad. Being able to access files locally within memory of the iPad is much more efficient compared to having to download the data on the server upon each request. It also extends the ability to review data when you are at a location where there is no 3G signal or Wi-Fi connection. Even if the device is constantly connected to the network, transferring a large PDF file with thousands for CRF images can hinder an interactive viewing experience.

STEP 2 – Selecting Study Protocol

The logic code on the server is SAS but once the data is transferred over to the iPad, the code used to manage the study protocol information is applied using Objective C. This is based on the C programming language with libraries defined for software development on iOS devices such as the iPad. Before the user can select a particular CRF or dataset within a study, they must first select a specific protocol study. The following code segment creates this table view to manage protocol selection.

```
// Generate Table for Study Protocol
(UITableViewCell *) tableView:(UITableView *) tableView
cellForRowAtIndexpath:(NSIndexPath *) indexPath
{
    static NSString *kCellID = @"cellID";
```

```

UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:kCellID];
cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleSubtitle
reuseIdentifier:kCellID autorelease];
cell.autoresizesSubviews = YES;

if (tableView == self.searchDisplayController.searchResultsTableView) {
    //cell.textLabel.text = [[self.filteredItemList objectAtIndex:indexPath.row]
    objectForKey:@"Name"];
    //cell.detailTextLabel.text = [[self.filteredItemList
    objectAtIndex:indexPath.row] objectForKey:@"StudyProtocol"];
    cell.detailTextLabel.textColor = [UIColor blackColor];
    cell.textLabel.text=@"Protocol ABC-123";
} else {
    cell.textLabel.text = [[[itemList objectAtIndex:indexPath.section]
    objectAtIndex:indexPath.row] objectForKey:@"Name"];
    //cell.detailTextLabel.text = [[[itemList objectAtIndex:indexPath.section]
    objectAtIndex:indexPath.row] objectForKey:@"StudyProtocol"];
    cell.detailTextLabel.textColor = [UIColor blackColor];
}

if ([cell.textLabel.text isEqualToString:@"Protocol ABC-123"])
    cell.imageView.image = [UIImage imageNamed:@"ABC123_page1.png"];
else
    cell.imageView.image = [UIImage imageNamed:@"ABC123_icon.png"];
return cell;
};

```

The iOS SDK from Apple provides access to standard graphical user elements on the iPad which can be accessed through the program. In this example, the TableView object and all its related attributes including text and color which can be manipulated programmatically. The object oriented programming syntax of Objective C refers to the attributes with the use of dot notation. For example, if the "CELL" is assigned to TableView object, The label text can be assigned to the object referenced by the three level dot notation as shown here:

```
cell.textLabel.text
```

This is analogous to how SAS uses dot notation to reference a datasets from a specified library such as:

```
sashelp.vstable
work.mydata
```

The BASE SAS programming language is designed to work with data so the dot notation is geared towards referencing datasets. The Objective C used for developing iPad Apps are designed to interact with graphical user elements so all of its objects are manipulated using this dot notation method.

STEP 3 – List-View of Data Items

The table view is useful to display textual information such as information related to a study protocol. Another useful object on the iPad is the use of the list-view which uses an array to store its values. Similar to SAS, the advantage of using an array is that it manages the data in memory and thus very fast in processing.

```

@implementation ClinDataManager
+ (NSMutableArray *) generateDataList
{
    NSMutableArray *tmpArr = [[NSMutableArray alloc] init];
    NSMutableDictionary *tmpLst;

    tmpLst = [NSMutableDictionary alloc] init];
    [tmpLst setObject:@"DEMOG - Demographics " forKey:@"Name"];
    [tmpLst setObject:@"0" forKey:@"Coordinate"];
    [tmpArr addObject:tmpLst];
    [tmpLst release];
}

```



```

tmpLst = [[NSMutableDictionary alloc] init];
[tmpLst setObject:@"MEDHIS - Medical History" forKey:@"Name"];
[tmpLst setObject:@"1" forKey:@"Coordinate"];
[tmpArr addObject:tmpLst];
[tmpLst release];

tmpLst = [[NSMutableDictionary alloc] init];
[tmpLst setObject:@"ENROLL - Enrollment" forKey:@"Name"];
[tmpLst setObject:@"2" forKey:@"Coordinate"];
[tmpArr addObject:tmpLst];
[tmpLst release];

tmpLst = [[NSMutableDictionary alloc] init];
[tmpLst setObject:@"SURGERY - Surgery" forKey:@"Name"];
[tmpLst setObject:@"3" forKey:@"Coordinate"];
[tmpArr addObject:tmpLst];
[tmpLst release];
}

```



This example code segment illustrates how the array can be used to manage names of clinical datasets. This is used in the selection menus which are displayed on the left panel of the iPad landscape view or a popup selection list of the portrait view.

The array can be used for many lists such as metadata which go beyond just the names of datasets. It can also be used for metadata such as variable names, or controlled terms that was described in step 3 of the process during the information capture on the server with SAS. The difference is that the iPad needs to have a way of accessing this information in real time including the ability to instantly search and then display the resulting information. Even though the disk drive on the iPad is solid state, it is not as fast as flash memory. Thus, clinical data that needs to be searched and displayed on the iPad, the use of arrays is an optimal method.

CONCLUSION

In the "post-PC" era where information is accessed instantly and ubiquitously, the ability to deliver clinical data to an iPad is essential for conducting clinical trials efficiently. Power users are no longer tethered to their desktop computers since Clinical Research Associates and Medical Reviewers need to be at the clinical site during the administration of drug treatment or medical device close to the patients. The PC has evolved from cryptic commands to the use of graphical user interfaces using a mouse.

This however is still an abstraction from how users interact with real world objects. The use of the iPad brings us one step forward towards an optimized clinical data review. The iPad provides the ability to zoom with the use of multi-touch gestures to view a case report form, or code-list value pertaining to patient information. This paper presents a set of technologies including a portion on the server which uses SAS to capture and transform the information in preparation for the Clinical Reviewer App on the iPad. The iPad then organizes and displays the information in a way that is easy for non-technical users to navigate

to the clinical information of interest. This can make the difference in the handling of a serious adverse event of a subject resulting in the success of a drug approval or medical device submission.

CONTACT INFORMATION

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

Sy Truong
MetaXceed, Inc. (MXI)
42978 Osgood Rd
Fremont, CA 94539-5627
tel: 510.979.9333
fax: 510.440.8301
E-mail: sy.truong@meta-x.com
Web: www.meta-x.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.

iPad, iTunes and iOS are trademarks of Apple Computers.
Other brand and product names are trademarks of their respective companies.

Clinical Reviewer are trademarks of Meta-Xceed, Inc.

Sarah Rotman Epps "The "Post-PC" Era: It's Real, But It Doesn't Mean What You Think It Does", May 17, 2011