

Speeding Up the Validation Process of Clinical Trials Data

Ali Dootson

PAREXEL International Limited

PAREXEL International

PAREXEL is one of the largest contract pharmaceutical outsourcing organisations in the world.

With a commitment to providing solutions that expedite time-to-market and peak market penetration, PAREXEL has developed significant expertise in clinical trials management, data management and biostatistical analysis amongst others

Headquartered near Boston, MA, PAREXEL operates in 53 locations throughout 33 countries around the world, and has approximately 4,600 employees

PAREXEL International Clinical I.T. Department

Clinical I.T. provides a high level of support to the CRS group - including Data management and Statistics amongst others

We develop applications and tools using relevant software that help these departments function more efficiently and to reach their goals effectively

Clinical Trials Management

◆ Process

- Database Set Up (QC'd and Validated)
- **Validation Programs Written and QC'd (Written in SAS)**
- First CRF In house
- Data Entry
- Data Validated
- QC Data then Export back to Client

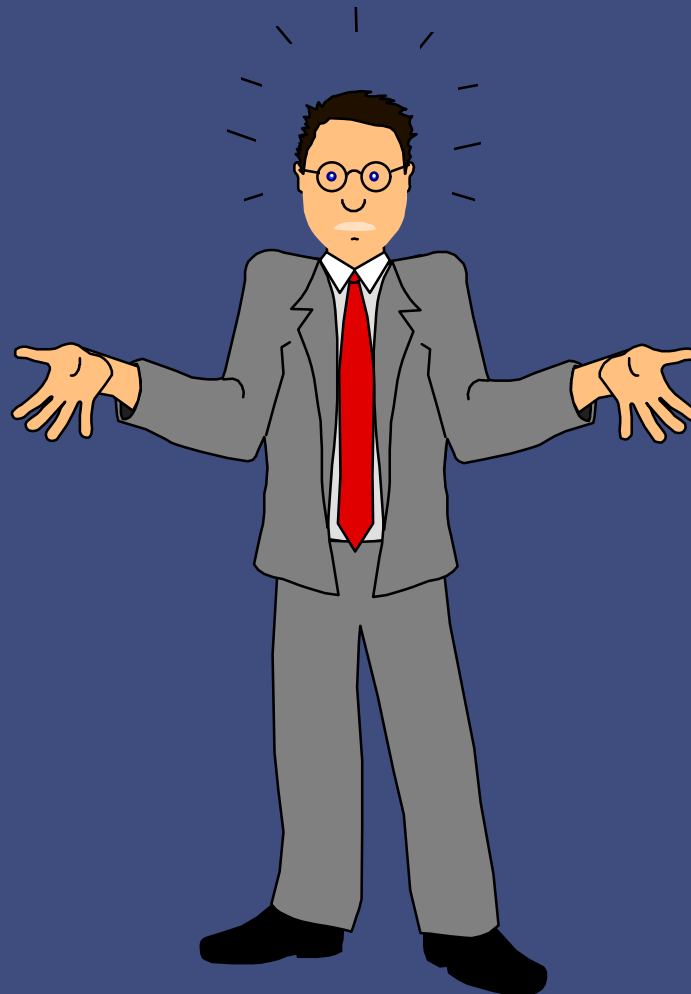
◆ Main Bottlenecks

- **Validation Programs**

What Does A SAS Programmer Bring to Validation Programming?

- ◆ Expensive
- ◆ Doesn't Understand the Specifications
- ◆ Doesn't Know the Database
- ◆ SAS Programming Skills

Problem



How WAS Validation Programming Done?

- ◆ SAS Programmer Given the Specification
 - Doesn't really understand it completely
 - Doesn't really know the database
- ◆ Copies Other Validation Programs
 - Why reinvent the wheel?
 - Cut and Paste, Cut and Paste
- ◆ Error Prone
 - May miss something

How Could the Programming be Done?

- ◆ Why not get the people that know the database to do the Validation Programs?

The Data Managers

- Know the database very well
 - Limited or No SAS Skills
- ◆ How do we balance between the two?

Solution



Validation Program Generator (VPG)

- ◆ The Validation Program Generator provides a simple data driven interface to allow **Data Managers** to produce textual validation specifications and equivalent SAS validation programs for their studies.
- ◆ No specific SAS programming knowledge is required but the Data Manager is required to **understand data validation principles and their own database properties**.

VPG Overview

- ✓ The system uses code and text fragments (templates) to construct the specification and program documents.
- ✓ Additional templates can be added to the system by suitably trained programmers.
- ✓ The system is written using SAS/AF and runs on OpenVMS.
- ✓ The system is modular allowing different levels of user access only to the parts they require.
- ✓ The generated programs are consistent with the requirements of the PAREXEL internal query management system, which creates queries from the errors datasets produced by the validation program.

What to Know

- ◆ The Database
- ◆ Minimal SAS - Enough to write an IF Statement
- ◆ What is Required by the Specifications
- ◆ What Outcomes the Test-Data Should Produce

VPG - How Does It Work?

- ◆ Check and Derivation Templates are stored globally
- ◆ User Input:
 - Dataset on which to Validate
 - When to perform the Check (Subset data)
 - What Kind of Check is Required
 - The Variable to Check
 - Any References

Other User Inputs

- ◆ Cross Panel Merges
- ◆ Deal With Changes to Database
- ◆ Disable unwanted sections (When Live)
- ◆ Overwrite Query Text
- ◆ Lock the Program

Support Personnel maintain templates
and support the users

Looking Forward



Client Focus

- ◆ Many different protocols from a client.
 - Set up each one?
 - Copy and change?
- ◆ No.
- ◆ Protocol Distribution
 - All Checks for a series of protocols (e.g. A client) are programmed
 - Using EXCEL interface - relevant checks are picked for each study.

Validation of Validation

- ◆ PASS/FAIL Templates
 - Ensure that there is test data for every check.
 - At least 1 record must pass and 1 must fail every check
- ◆ Know that the checks are validating the data correctly.
- ◆ Any new templates or updates are checked and documented by the support team

Timing

- ◆ Depends on the complexity of DVS and Database set up
- ◆ Panel and Protocol Copy was to help set up time. Now enables multiple users
- ◆ Warehousing (Protocol Distribution) will mean minimal set up time for suites of programs.
- ◆ Final DVS for Client can be in EXCEL format

VPG in Action

Validation Program generator

Protocol: TRNPROT9

Data Source: DATASETS

- < Specify Validations for this protocol >
- < Specify Data Listings >
- < Generate Program >
- < Generate DVS >
- < View outputs >
- < End >

□&d0DData Validation Specification□&d0

Protocol:

Created: 12MAR2001

(n.b. Sections printed in italics have been removed from the program)

□&d0D1.Global definitions□&d0

The following Globals are defined for use in the checks:-

End of Global Definitions.

□&d0D2.Data Validations□&d0

Data validation checks are coded as follows:-

2.1. Checks for table DEMOG: |

2.1.2. The following checks are always performed .

2.1.2.1. The Value present check which ensures that a variable is populated is performed on the following variables:-

Code	Variable
2.1.2.1.1.	DOB1W (Date of Birth)
2.1.2.1.2.	RAC1C (Race)
2.1.2.1.3.	SEX1C (Sex)

End of checks always performed.

End of checks for table DEMOG.

End of data validation checks.

End of Data Validation Specification for protocol

VALIDATE.SAS

PAREXEL
Clinically thinking.™

```

/***** Validation Suite *****/

/* Code generated on: 12MAR2001
   by: ALI DOOTSON
   for protocol: DOOTSON
   under SAS version: 6.12 */

libname DATABASE 'XXX:[XXX.XXX.DATASETS]';
libname ERRORS 'XXX:[XXX.XXX.VP.ERRORS]';
libname LIBRARY 'XXX:[XXX.XXX.LIBRARY]';

proc DATASETS lib=ERRORS kill;
quit;

/* 1. Global definitions */
/* end of Globals */
/* Validation checks */

/* Checks for panel DEMOG */
data ERRORS.DEMOG;
  length QRYTXT1-QRYTXT4 PRBMSG1 $200 PRBCOD1A $30 VLDTP1A $40 PNL1A $8
         TGT1C $3 SUGACT1C $4;
  retain PNL1A 'DEMOG' ;
  set DATABASE.DEMOG( keep=SEX1C RAC1C DOB1W CT_RECID FRMNAM1A MOD1A VIS1A
                     PAGSUF1N PAG1A PID1A );

  VLDTP1A="Value present";

  /* 2.1.2.1.1. Check Date of Birth is present */

  if DOB1W=. then do;
    PRBMSG1="DOB1W is missing.";
    QRYTXT1="Date of Birth" || " is blank";
    QRYTXT2=".";
    QRYTXT4=" Please provide the missing information.";
    QRYTXT3="";
    PRBCOD1A="2.1.2.1.1.";
    SUGACT1C="Q";
    TGT1C="S";
    output;
  end;

  if 0 then output; /* prevent output when all checks disabled */

  keep PNL1A PID1A PAG1A PAGSUF1N VIS1A MOD1A FRMNAM1A CT_RECID VLDTP1A
      PRBCOD1A PRBMSG1 SUGACT1C TGT1C QRYTXT1-QRYTXT4;

run;

/* end of checks for panel DEMOG */
/* end of validation checks */

/***** end of Validation suite *****/
```

VPG Deployment

PAREXEL
Clinically thinking.™

Use of VPG Within PAREXEL



Active Demonstrations

- ◆ Unfortunately not available here
- ◆ We are happy to arrange demonstrations in our offices
- ◆ Contacts
 - Sheffield - tel. 0114 225 1000
 - Oliver Rees
 - Sandy Meek
 - Uxbridge - tel. 01895 23 8000
 - Alistair Dootson
 - Siddick Mohamoodally
 - Nick Darwall-Smith