

"Effectively Utilizing the Power of the SAS Display Manager"
Richard M. Priem, The Priemere Consulting Group.

ABSTRACT

The SAS System can be invoked in a variety of different modes. One of the most powerful and efficient tools available is the Interactive Display Manager. With the appropriate techniques, applications can be developed, debugged, and maintained with great ease using the Dialog Manager.

This paper will present a quick overview of the SAS Display Manager, followed by an outline of effective procedures and "tricks" to maximize your productivity. Attention to these guidelines should lead to fast and efficient development of neat, clean, and orderly programs. The programs will follow a logical top-down flow and contain reasonable documentation - two of the major factors in producing low-maintenance code. Finally, similar techniques can be adapted for debugging existing software.

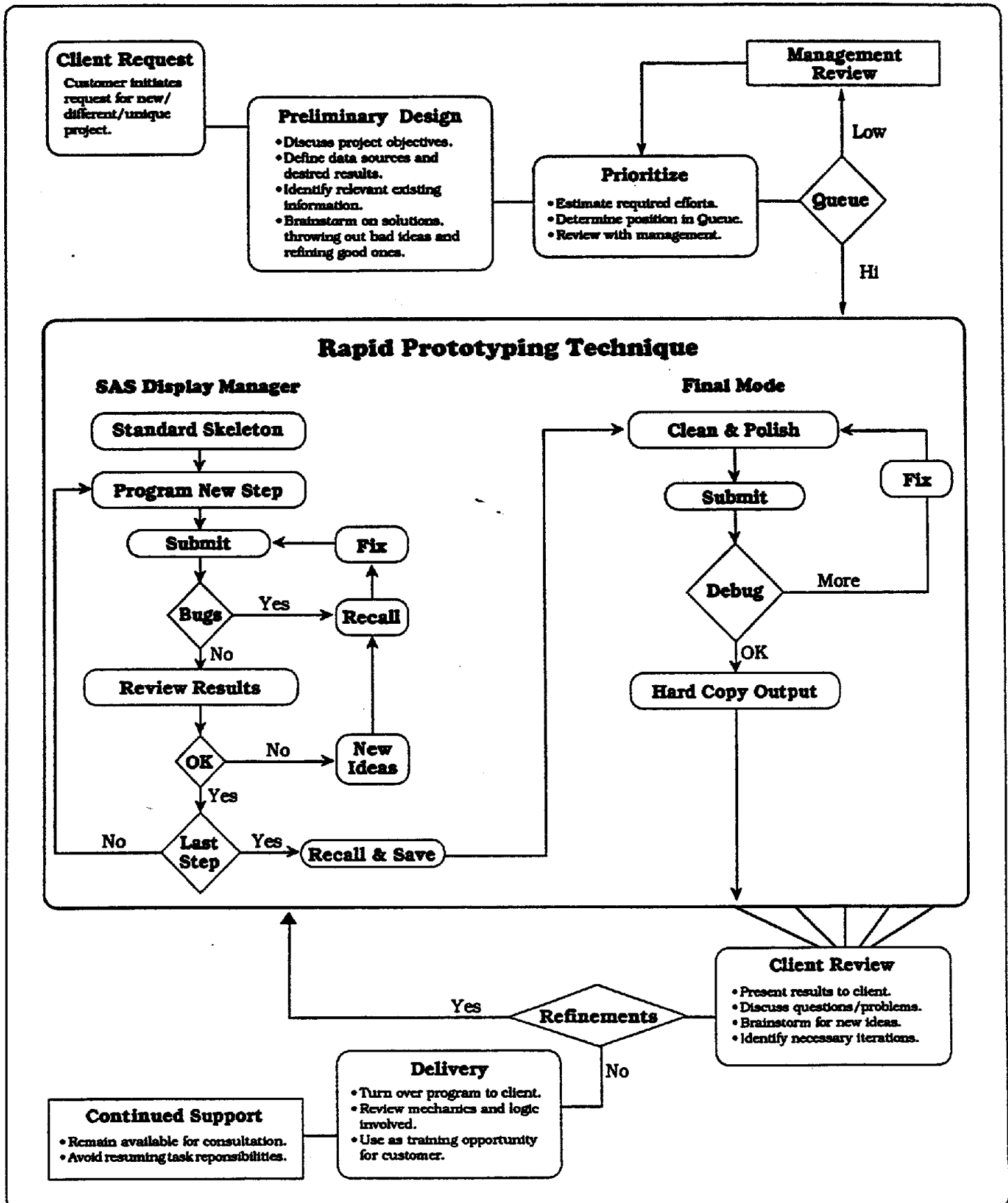
These procedures have been proven in their success. Although each situation is unique, the author has repeatedly demonstrated not only nickel-and-dime (5-10%) advantages in development time over competing alternatives, but often on the order of 200-400% and even more!

BIOGRAPHY

Rich is a Software Engineering Specialist with a dozen years of SAS programming experience. He graduated Magna Cum Laude with degrees in Engineering, Mathematics, and Computer Science in 1979, and then began his career with Exxon Production Research Company doing reservoir modeling and analysis. In 1986 he left EPR to become a computer consultant, and has since worked in the Chemical, Environmental, Transportation, and Petroleum industries. Rich is currently working for Exxon's Offshore/Alaska (OSAK) Exploration Department, managing a project for Arco's Southern Production District, and continuing to market his company's services to other potential clients. ☺

Premiere Projects by Prototyping

(Customer-Driven Project Management)



OBJECTIVES OF A SAS SESSION

- DEVELOP NEW APPLICATIONS
- MAINTAIN EXISTING PROGRAMS
- AD HOC QUERIES TO DATA BASE

AVAILABLE ENVIRONMENTS FOR SAS

- INTERACTIVE DISPLAY MANAGER
- FOREGROUND EXECUTION
- BACKGROUND BATCH JOB

DISPLAY MANAGER: PROS AND CONS

ADVANTAGES

- IMMEDIATE FEEDBACK
- EFFICIENT ITERATIONS

DISADVANTAGES

- MORE CONFUSING
- REQUIRES MORE SKILLED USER

DEMO PROGRAM

- TWO INDEPENDENT SOURCES OF DATA
- WANT TO SEE HOW THEY COMPARE
- POLYGON DESIGNATING GEOGRAPHIC AREAS FOR COMPARISON

```

PROC O TRACE
/*****
/**** DataSet Name = C100.Clist(SasRMP)
/**** Invoke SAS Display manager with appropriate allocations for RMP.
/****
IF &TRACE = TRACE THEN CONTROL NOFLUSH LIST CONLIST
ELSE CONTROL NOFLUSH NOMSG

BLANK

/****
/**** Allocate some files for convenience.
/****
Allocate Old ReUse F1(SasUser) DA(SasUser.Profile)
Allocate Old ReUse F1(Source) DA(Sas.Source)
Allocate Old ReUse F1(SasExec) DA(Sas.Source(AutoExec) )
Allocate Old ReUse F1(Output) DA(Batch.Output) )

/****
/**** Invoke SAS Driver from "MG.Spf.Clist"
/****
SAS WORK('100 50') +
Load(' 'C100.Sas.Devlib2' ' 'Prod.Sas.TestLib'') +
Autos(' 'C100.Sas.Macros' ' 'Osak.Sas.Macros' '')+
OPTIONS('DMS DEV=GDDM79 NOCAPS-Workinit Stimer MemRpt NoTso')

FreeAll

```

```

Output ;
  Cbanner Blue;
Log;
  Cbanner Cyan;
  Cursor 8 ;
  Split ;
Program;
  AutoRoll Log Off;
  Include Source(Zmaster);

```

```

*=====;
*== Sas.Source(Zmaster ): Set up for Frontiere Sub-Salt Project.
*=====;
*== 05/07/91 RMP: Set up this working environment.
*== 06/07/91 RMP: Enhanced setup of graphics options.
*=====;
Options Errors=1 Dquote;

Title H=1.5 C=Red "Template for Title Line";

FootNote H=0.8 F=Duplex
  J=L  " Dsn: C100.Sas.Source(Zmaster) "
  J=R  " &SysDate &SysTime "
  J=R  " By: " C=Red " Rich Priem ";

Goptions Ftitle=Xswiss Ctitle=Blue Cby=Cyan Hby=0.8 Border;
Goptions Colors=(White,Red,Green,Yellow,Cyan,Pink,Blue);

Options Dquote Mprint Mlogic MacroGen SymbolGen Merror Serror;
Options NoMprint NoMlogic NoMacroGen NoSymbolGen NoMerror NoSerror;

* %AllocTek(Dest=PS019);
* Goptions Colors=(Black,Red,Green,Yellow,Cyan,Pink,Blue);

Run;*****;
*** Sec.10: Allocate required external files.
*****;
Proc AllocDsn;
  * Alloc Shr ReUse Fi=Poly      Dsn='OSAK.Frnt.SubSalt.Poly' ;
  * Alloc Shr ReUse Fi=Grid      Dsn='OSAK.Frnt.SubSalt.Grid' ;
  * Alloc Shr ReUse Fi=Cont      Dsn='OSAK.Frnt.SubSalt.Cont' ;
  * Alloc Shr ReUse Fi=Fill      Dsn='OSAK.Frnt.SubSalt.Fill' ;

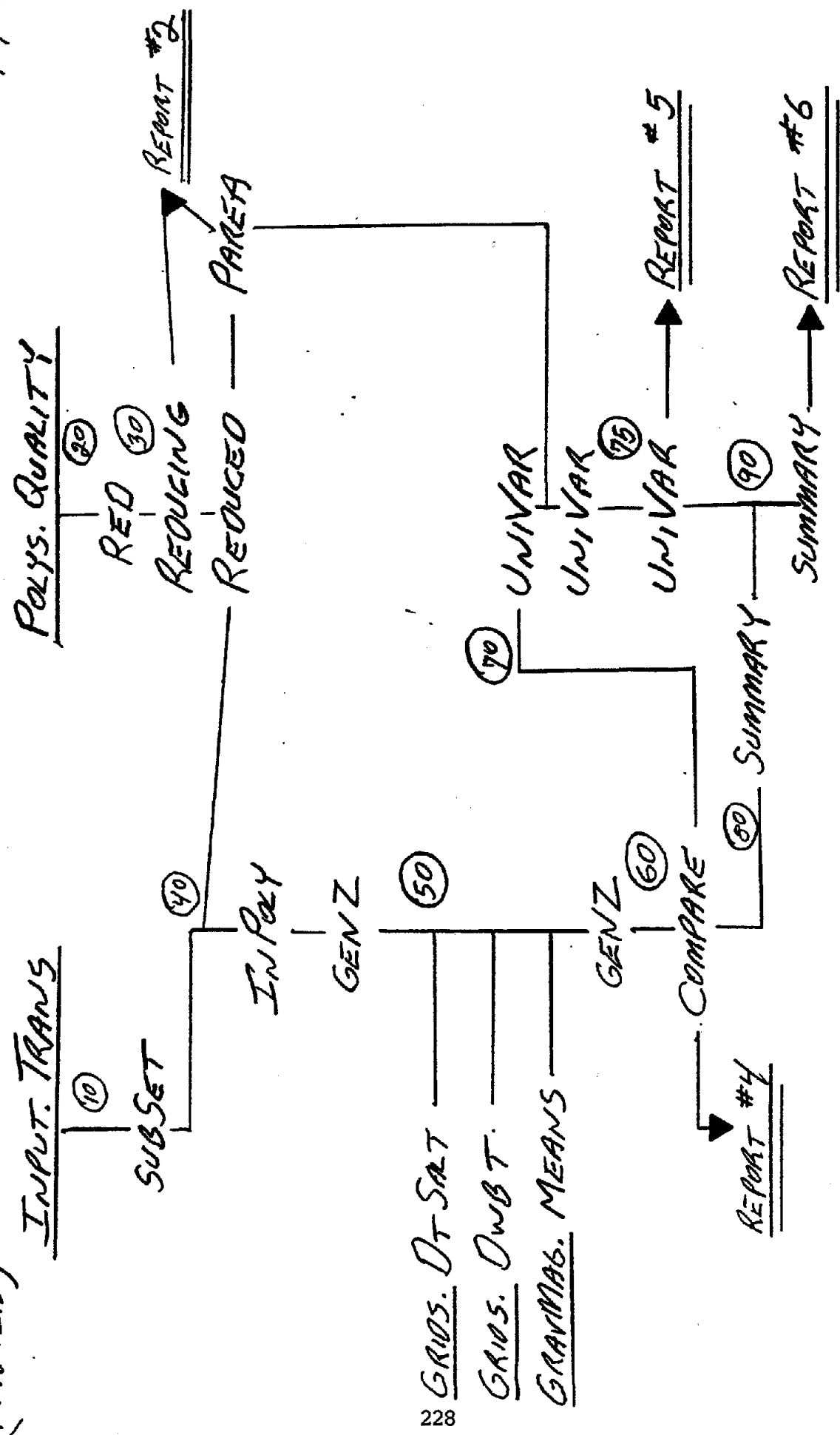
  * Alloc Shr ReUse Fi=Lease     Dsn='OSAK.CCE.LeaseMap.SasData';
  * Alloc Shr ReUse Fi=GomWell   Dsn='OSAK.CCE.GomWell.SasData';

  * Alloc Shr ReUse Fi=GeoPol    Dsn='EXPL.OffE.SasCarto.Gm0015.All';
  * Alloc Shr ReUse Fi=Maps      Dsn='C100.Sas.P.Maps';
  * Alloc Shr ReUse Fi=GravMag   Dsn='C100.GravMag.SasData';
  * Alloc Shr ReUse Fi=Ebreaks   Dsn='C100.Ebreaks.SasWork' ;
  * Alloc Shr ReUse Fi=Greplay   Dsn='C100.SAS.MapsPros.Greplay';

  * Alloc Shr ReUse Fi=Gravity   Dsn='B202.COM.Gravity.SasData';
  * Alloc Shr ReUse Fi=Bevan     Dsn='Osak.Dwrs.Bathymtr.Grids';

Run;*****;
*** Sec.99: The End, that's all Folks!
*****;

```



```

//C100DEMO JOB 'WT733U.FRONTIER.01','Priem: Demo Job',
//  MSGLEVEL=(1,1),COND=(8,LT),PRTY=13,
//  CLASS=Z,NOTIFY=C100,MSGCLASS=T,TIME=10
//*
//PRINT OUTPUT DEST=HOUOAP.LOCAL,DEFAULT=YES,JESDS=ALL
//PLOT OUTPUT DEST=N15R452 OSAK COLOR PLOTTER
//*
//*****
//*   DataSet Name = C100.Sas.Regional.Paper(Demo)
//*   Old Name = C100.Final.Jobs(Map2Seis)
//*-----
//*   Ground-truthing salt thickness from gravity.
//*   Using interpreted seismic horizons in high quality seismic areas.
//*-----
//* 10/22/91 RMP: First began working on this problem.
//* 10/26/91 RMP: Refined for presentation at team meeting.
//* 11/11/91 RMP: Cleaned up for presentation at SC-SUG Conference.
//*****
//MAP1 EXEC SASRMP
//*
//SASLIB DD DISP=SHR,DSN=C100.SAS.DEVLIB2
//SASAUTOS DD DISP=SHR,DSN=C100.SAS.MACROS
//*
//*-----
//POLYS DD DISP=SHR,DSN=OSAK.FRNT.SUBSALT.POLY
//GRIDS DD DISP=SHR,DSN=OSAK.FRNT.SUBSALT.GRID
//GRAVMAG DD DISP=SHR,DSN=C100.GRAVMAG.SASDATA
//*-----
//SYSPRINT DD SYSOUT=*
//SYSIN DD *

```

Options Errors=1 Dquote Parm='';

Title H=1.5 C=Red "Gravity-vs-Seismic Salt Base";
FootNote H=0.8 F=Duplex
J=L " Dsn: C100.Final.Jobs(Map3Seis)
J=R " &SysDate &SysTime "
J=R " By: " C=Red " Rich Priem ";

Goptions Ftitle=Xswiss Ctitle=Blue Cby=Cyan Hby=0.8 Border;
Goptions Colors=(White,Red,Green,Yellow,Cyan,Pink,Blue);

Options Dquote Mprint Mlogic MacroGen SymbolGen Merror Serror;
Options NoMprint NoMlogic NoMacroGen NoSymbolGen NoMerror NoSerror;

```

Run;*****;
*** Sec.01: Allocate required input files.
*****;
Proc AllocDsn;
  Alloc Shr ReUse Fi=Input DA='C100.GomSeis.SasData';

```

```

Proc Contents Data=Input. All ;
Title3 "0) Contents of Data File";

```

```

Run;*****;
*** Sec.05: Macro variables to define Area-of-Interest window.
*****;

```

```

%UserAoi(User=Sale139);

Run;*****;
*** Sec.10: Select a subset of the interpreted seismic data.
*****;
Title3 "3) SubSet the ShotPoint Data";
Data SubSet; Set Input.Trans;
    If &Xmin < X < &Xmax;
    If &Ymin < Y < &Ymax;
    Keep Alin Nlin SpId X Y;
    Keep Picker DatePick;
    Keep TsIt BsIt;

Run;*****;
*** Sec.20: Split Quality polygons into two groups.
*****;
Title3 "1) Identify Quality Polygon Areas";
Data Green Blue Red;
    Set Polys.Quality;
    Select (Color);
        When ('G') Output Green;
        When ('B') Output Red;
        When ('R') Output Blue;
        Otherwise Put "Debug: " Color=;
    End;
Proc Contents Position;

Run;*****;
*** Sec.30: Reduce density of data points.
*****;
Title3 "2) Reduce Density of Points on Polygons";
Proc Greduce Data=Red Out=Reducing
    E5= 500
    E4= 1000
    E3= 1500
    E2= 2000
    E1= 4000
    ;
    Id _Name_ ;
Proc Chart; Hbar Density / discrete;

Data Reduced; Set; If Density < 3 ;
    By PolyNo Notsorted;
    If First.PolyNo then PolyId+1;

Proc Parea Out=Parea; ID Color PolyNo _Name_ ;
Proc Print;          By Color;

Run;*****;
*** Sec.40: Check if Grid Point is in each Polygon.
*****;
Proc InPoly Data=SubSet
    Out =SubSet
    Poly=Reduced;
    IdPoly Color PolyNo;

Proc Contents Position;

```

```

Run;*****;
*** Sec.50: Generate Z Values from Grids.
*****;
Data GenZ; Set Subset;
      Salthick = Bslt - Tslt ;
      If Salthick ;
Proc Sort; By X Y;

Proc GenZ Out=GenZ Grid = Grids.DtSalt; Var DtSalt;
Proc GenZ Out=GenZ Grid = Grids.Dwbt ; Var Dwbt;
Proc GenZ Out=GenZ Grid=GravMag.Means ; Var BaseSalt;

Run;*****;
*** Sec.60: Calculate comparison parameters.
*****;
Title3 "4) Compare Salt Thickness Values";
Data Compare; Set GenZ;
      GravThck = BaseSalt - DtSalt ;
      Salthick = (Bslt - Tslt) * 14500 / 2000 ;
      DbSalt = DtSalt + Salthick;
      Compare = Log(Salthick/GravThck) * 1e3 / Log(2);
      Deviate = Abs(Salthick-Gravthck);
      DeltaH = (Salthick-Gravthck);
      Format Numeric Comma10.0 ;
      Format DatePick Date.; Drop Symbol;

Proc Print Data=Compare(Obs=20);
Proc Chart; Hbar Deviate Compare GravThck Salthick;

Proc Sort;
      By Color PolyNo Alin Nlin Picker DatePick SpId;

Run;*****;
*** Sec.70: Univariate Statistics for each Polygon.
*****;
Title3 "5) Summary Comparison for Polygon Areas";
Proc Univariate NoPrint Data=Compare;
      By Color PolyNo;
      Output Out=Univar
            N=Nshots Q1=DevQ1 Q3=DevQ3
            Median=Deviate Compare GravThck Salthick DtSalt;
            Var Deviate Compare GravThck Salthick DtSalt;

Data Univar;
      Merge Univar Parea;
      By Color PolyNo;
      Xmid = X / 1e5 ;
      Ymid = Y / 1e5 ;
      Drop X Y _Nvert_;
Proc Means;

Proc Chart;
      Hbar Deviate Compare GravThck Salthick DtSalt;

Run;*****;
*** Sec.75: Rank from worst to best comparison.
*****;
Proc Sort; By descending Deviate;

```

```

Data Univar; Set;      If Deviate;
    R1 + 1 ;
    R1Val = Round(Deviate/100);
Proc Print;

Run;*****;
*** Sec.80: Summarize comparison results for each Line.
*****;
Title3 "6) Summary Comparison for each Seismic Line";
Proc Summary Data=Compare;
    By Color PolyNo Alin Nlin Picker DatePick;
    Var Deviate Compare GravThck Salthick DtSalt;
    Output Mean=
        Out=Summary(Drop=_Type_);

Run;*****;
*** Sec.90: Rank each line and combine with Polygon summary.
*****;
Proc Sort Data=Summary; By Color PolyNo descending Deviate;
Proc Sort Data=Univar; By Color PolyNo;

Data Summary;
    Merge Univar(In=In) Summary;
    By Color PolyNo; If In;
    If First.PolyNo then R2=0; R2+1;
    R2Val = Round(Deviate/100);
    Drop DevQ1 DevQ3 Nshots _Name_ Area;

Run;*****;
*** Sec.95: Sort into Rank order for Report.
*****;
Proc Sort;
    By R1 Color PolyNo;
    By R2 Alin Nlin Picker;
Proc Print;
    By R1 Color PolyNo Xmid Ymid R1Val;

Run;*****;
*** Sec.99: The End - That's All Folks !!!
*****;

```

"CLEAN-UP" PROCEDURE

- DELETE "FALSE STARTS" FOR EACH SECTION
- WORK FROM BOTTOM, LAST ONE IS GOOD ONE
- REMOVE USE OF ALL "FS" PROCEDURES
- ADD SUB-TITLES FOR ALL OUTPUT
- REVIEW COMMENTS FOR EACH SECTION

TRICKS

- SAVE AND INCLUDE FROM EXTERNAL PDS
(USE DATE STAMP TO KEEP TRACK)
- TEST WITH SUB-SETS OF THE DATA
- CATALOG "WORK" FILE FOR DEBUGGING

THE ART OF PROGRAMMING

- INDIVIDUAL "STYLES" MAY VARY
 - SOME ARE BETTER THAN OTHERS
- STRIVE TO CONTINUOUSLY IMPROVE
 - MORE EFFICIENT PROCESSING
 - MORE ELOQUENT CONSTRUCTIONS
 - MORE READABLE AND UNDERSTANDABLE LOGIC
 - LESS LINES OF CODE TO DIGEST

SOFTWARE MAINTENANCE

- NEAT AND CLEAN PROGRAM
 - DOCUMENTED INTERNALLY WITH COMMENTS
 - LOGICAL (DRAWABLE) FLOW
 - * RESULT: MUCH EASIER MAINTENANCE
- "A stitch in time saves nine..."

SOFTWARE LIFE CYCLE

- IF THE PRODUCT IS GOOD,
IT WILL CONTINUE TO BE USED AND IMPROVED
- IF THE CODE IS ORGANIZED AND FLEXIBLE
IT WILL CONTINUE TO EVOLVE
- AT SOME POINT,
WHEN IT BECOMES TOO COMPLICATED
IT WILL DIE