

Developing a Suite of Clinical Report Macros for SAS/PH-Clinical®

Paul Gilbert & Steve Light at DataCeutics, Inc

Abstract

In an effort to streamline the production of clinical reports, many companies are developing SAS report generation tools to produce listings, summary tables and graphics. This presentation discusses critical factors in the design of a suite of SAS-based Clinical Report Macros. Additionally, it describes how these report macros can be used within SAS/PH-Clinical as interactive report templates.

Introduction

At many companies collections of SAS reporting tools accumulate over time as programmers create SAS macros for use in specific projects. Once these macros are created, programmers are tempted to reuse and adapt existing macros to new projects. However, adaptation of project-specific macros can be problematic as they are created without consistent guidelines and may suffer from a lack of consistent syntax (macro parameters), output characteristics or analytical methods. In this paper we discuss developing a suite of general use reporting tools for producing a set of standardized reports, with emphasis on identifying common attributes, features and methods to be shared by all of the reporting tools.

This paper discusses the early phases in a Systems Development Life Cycle (SDLC) for development of a suite of SAS reporting tools. The specific methodology used has been described previously (Gilbert and Light, 1996). In each of the SDLC phases we will discuss the relevant points of developing Clinical Report Macros that are ultimately intended for use within SAS/PH-Clinical interactive templates. The phases of the SDLC to be discussed include:

- formation of a development team
- requirements definition
- report tool design

It is during these phases that features to be shared by all reporting tools, analytical methods and a common syntax for SAS macro parameters are defined.

Development Team

The first step in development process is to identify a development team. Development of a comprehensive suite of report macros can be a costly endeavor so it is important to have participation from all decision makers from the start. A project of this magnitude needs

management support as it will affect the appearance and format of all future clinical reporting. The team should include all user groups involved in planning, creating and reviewing clinical reports and/or clinical data:

- statisticians
- programmers
- medical monitors
- medical writers
- medical reviewers
- quality assurance

There are three main objectives for the team.

- identify reports that are common across projects. The common reports are the focus of standardization and subsequent candidate report macros.
- define rules for standard reports that can be applied consistently across projects.
- *agree* upon a final suite of reports and commitment from management.

Requirements Definition

In this phase decisions are made on a standard page layout for all reports, the scope of reports to be managed by the reporting tools, and the structure of individual reports.

The **standard page layout** should be defined first. This refers to page attributes (listed below) which are consistent across all reports. One or more templates are developed to illustrate the standard page layout. This should be approved and committed to by the team and management before going ahead with development of the tools. Standard page attributes should include

- landscape and portrait output settings: page size, line size, fonts, margins
- graphics display settings: drivers, orientation, fonts, sizes, margins, colors,
- placement of standard information such as company name, project, protocol, appendix/table number, report titles, program name/location, output file name/location and date/time stamp.

The next step is to **define the scope of the report suite**.

Many standard reports that are common across projects can be handled by standard report tools if agreement is reached on a standard report structure. These reports usually include patient listings, demographic summaries, patient characteristic summaries and safety summaries. Sample reports are reviewed to identify common characteristics and a standard structure is derived, detailing components of a report which will be fixed (e.g. will dose groups be grouped in rows or columns) and which will be flexible, allowing user or study-defined options. Common attributes to standardize may include:

- placement/text of titles
- placement/text of footnotes
- placement of dosing information
- dosing information standard labels
- by group placement
- page break placement
- line break placement

Next **identify each individual report.**

- define an overall organization for each individual report
- specify required fields, major column and row content, optional fields, paging options, display formats and required data manipulations
- identify any data requirements such as structure of file, required variables, variable types and variable names.

For each report create a template for the report structure (figures 1 and 2), annotated with description of required and optional components.

The overall format of the reports must be consistent, but some flexibility should be incorporated into the standards. Typical flexible portions of the reports are listed below.

- titles and footnote content
- column and Row Variables
- column and Row Labels
- dosing or Treatment Labels
- number of Doses Displayed
- number of Time Points Displayed
- groupings (i.e. by center or overall)

Once all reports have been defined they can be grouped and categorized by type of data (demographics, adverse event, laboratory summary), or type of report (listing or frequency summary for data structured with a single record per patient, summary statistics tabulation for data structured as multiple records per patient and parameter). Specific report tools can now be defined to generate

each report type.

A sample suite of report macros, described below, illustrates how reports can be categorized. This suite includes both data-specific (AE1, MED1) and data structure report macros (TABLE0 thru TABLE5). Figures 1 and 2 are example page layouts for two of the AE1 and TABLE1 report macros. These two report macros are used as illustrative examples in the next section.

- **%PRINT1** generates data listings using PROC REPORT. This macro enables long fields to be split over multiple lines and allows long lines to be wrapped.
- **%AE1** generates an adverse event table of frequency and percent of group. Reporting options include grouping by body system, preferred term, or both. Results are split out across a grouping variable (usually dose), a pageby variable is optional.
- **%MED1** generates a medications table of frequency and percent of group. Reporting options include grouping by ATC classifications or medication name or by both. Results are split out across a grouping variable (usually dose), a pageby parameter is optional.
- **%TABLE0** generates a frequency count and/or statistics summary for data which is structured one record per patient (demographics, patient data, summary data). It produces a table with N and % ff treatment group of discrete variables such as AGE, SEX, etc. and summary statistics for continuous variables such as WEIGHT.
- **%TABLE1** generates a table of frequency and percent of group for a parameter or a combination of a parameter and a classification. The dataset needs to be structured so that parameter and classification are variables. Examples of this type of table include medical history, baseline physical exams, baseline diagnosis and baseline ECGs.
- **%TABLE2** generates a table of *baseline vs post* patient frequency counts of a classification for a parameter. The dataset needs to be structured so that parameter and classification are variables. Examples of this type of table are physical examination and ECGs.
- **%TABLE3** generates a table of row summary

statistics for a parameter/value or parameter/timepoint/value with dosing displayed as columns. The dataset needs to be structured so that parameter and classification/value are variables. Examples of this type of table are laboratory, vital signs, ECGs and efficacy.

- **%TABLE4** generates a table of column summary statistics for result and change from baseline for a parameter, dose and timepoint. There is an option for counting the number of patients below and above the reference range, which is commonly used for laboratory data. The dataset needs to be structured so that parameter and classification/value are variables. Examples of this type of table are laboratory, vital signs and ECGs.

- **%TABLE5** generates a table of laboratory data reference range shift displays with baseline as the row variable and post baseline as the column variables. Options include inclusion or exclusion of missing data, display or suppression of percentages and pageby variables.

Report Tool Design

Begin the design process by identifying tasks and functions which will be shared by all report tools and consider centralizing those tasks in a (small) set of utility macros. For example, the previous section addresses a **standard page layout**. When a standard layout for portrait, landscape or graphics reports has been defined, a utility macro can be created to define and manipulate the standard page layout. This macro can be called by the report macros. This modular design assures the use of a standard page layout across all reports and simplifies code maintenance; if page layout standards are revised the changes can be implemented by modifying a single program. Utility macros can be created for other common report macro functions such as:

- data access
- dictionary access
- page numbering
- operating system specific interfacing
- dynamic formatting
- defining titles & footnotes

Once features common to all reports are implemented through utility macros, an individual report tool can be designed and a design document prepared for each report type identified. The design document should precisely describe the features and specifications for each report

and address each of five key points.

- data structure specification
- macro call syntax
- analytical methods
- PH-Clinical user interface
- macro code design

The Macro Design document should be reviewed and approved by the team before actual programming starts.

The **data structure specification** defines any requirements for the structure of the input dataset. Structural requirements may include normalization, required variables, required variable names, optional variables and any required formatting. The input requirements are driven by the output needs.

The **macro call syntax** is used to describe the macro functionality. This detailed description of the macro parameters is used for defining the purpose, usage and default values for all parameters. There are important factors to consider when designing the report macro parameters for a series of report macros.

- To provide a common interface and facilitate ease of use apply consistent macro parameter naming, purpose and syntax across all macros.
- Identify all parameters as required or optional. Keep the parameter type consistent across all macros.
- Always use keyword macro parameters with descriptive names. Positional parameters are not a viable option in a tool which may have 10 or more parameters.

Consistency of report macro calls allows the use of modular coding, reduces the time required for code maintenance, and facilitates ease of use for the users of the report macro system. The sample AE1 and TABLE1 macro calls, in italics, illustrate these points. Parameters such as IN, TITLE, FOOTNOTE are identical in both the report macros. Parameters such as ROW have a common purpose and argument syntax, but have different default values in the two macros.

```
%AE1 (in=, title=, footnote=, pageby=, pat=sub,  
datyes=data_yn='Y', column=dose, nlabel=y,  
count=patients, row=aewhobod aewhopre,  
order=FREQ, descend=Y, totalg=y, squeeze=, props=);
```

in= Required, no default. Input data set.

title= Optional, no default. Number and text

for titles used on table. The number is specified followed by the text enclosed in double quotes.

footnote= Optional, no default. Number and text for footnotes used on table. The number is specified followed by the text enclosed in double quotes.

pageby= Optional, no default. Name of variable(s) used to PAGEBY on report.

pat= Required, default=**sub**. Patient identifier variable.

datyes= Required, default=**data_yn='Y'**. Condition used to select observations with adverse event data.

column= Required, default=**dose**. Name of the column variable.

nlabel= Optional, default=**Y**. Option to display (N=) in the dose label. The calculation of (N=) in the label will be based on a combination of the pageby and column variables.

count= Required, default=**patients**. Sets the macro to count either the number of patients(count=**patients**) with an event or total count of events(count=**events**) recorded.

row= Required, default=**aewhobod aewhopre**. Name of the row variable(s). The default produces a display of WHO body system and WHO preferred term. Row=**aewhobod** or Row=**aewhopre** can be used. A length may be supplied for each row parameter in the format of row parameter=length. For example: row=**aehwhobod=25 aewhopre=10**.

order= Optional, default=**FREQ**. Determines the order of rows on the table. **FREQ** orders rows by incidence. Assigning any other value to order will cause ordering of rows in alphabetical order of row variables.

descend= Optional, default=**Y**. When order=**FREQ** this parameter controls

ascending or descending order of frequency.

squeeze= Optional, no default. Set *squeeze=Y* to compress the display of data on the table. It can be used when the data does not fit across a single line.

props= Optional, proc report options.

totalg= Optional, default=**Y**. Controls creation and display of a 'TOTAL' column on the final table.

%TABLE1 (in=, title=, footnote=, pageby=, pat=sub, datyes=, column=dose, nlabel=y, row=, order=, descend=, squeeze=, totalg=Y, props=);

in= Required, no default. Input data set.

title= Optional, no default. Number and text for titles used on table. The number is specified followed by the text enclosed in double quotes.

footnote= Optional, no default. Number and text for footnotes used on table. The number is specified followed by the text enclosed in double quotes.

pageby= Optional, no default. Name of variable(s) used to PAGEBY on report.

pat= Required, default=**sub**. Patient identifier variable.

datyes= Optional, no default. Condition used to select observations with table data. Group size (N) is calculated directly from the data set passed to **%TABLE1**. One observation per patient must be present in the data set in order to determine the proper group N.

column= Required, default=**dose**. Name of the column variable.

nlabel= Optional, default=**Y**. Option to display (N=) in the dose label. The calculation of (N=) in the label will be based on a combination of the pageby and column variables.

row= Required, no default. Name of row variable(s). Either a single variable representing a parameter or a pair of

variables representing parameter and classification may be used. A length may optionally be supplied for each row parameter in the format of row parameter=length. For example row=var1=25 var2=10.

order= Optional, no default. Determines the order of rows on the table. The default order rows by alphabetical order of row variables. Setting *order=FREQ* will order rows by frequency.

descend= Optional, no default. When *order=FREQ* this parameter controls ascending or descending order of frequency.

squeeze= Optional, no default. Set *squeeze=Y* to compress the display of data on the table. It can be used when the data does not fit across a single line.

totalg= Optional, default=Y. Controls creation and display of a 'TOTAL' column on the final table.

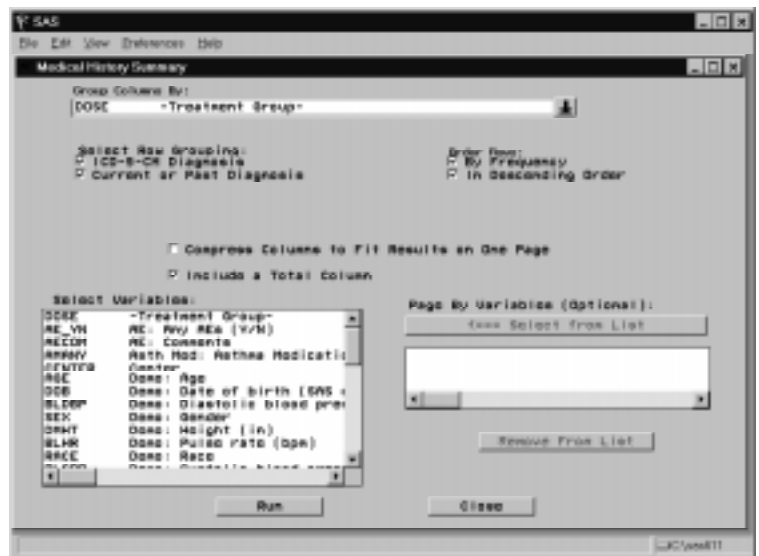
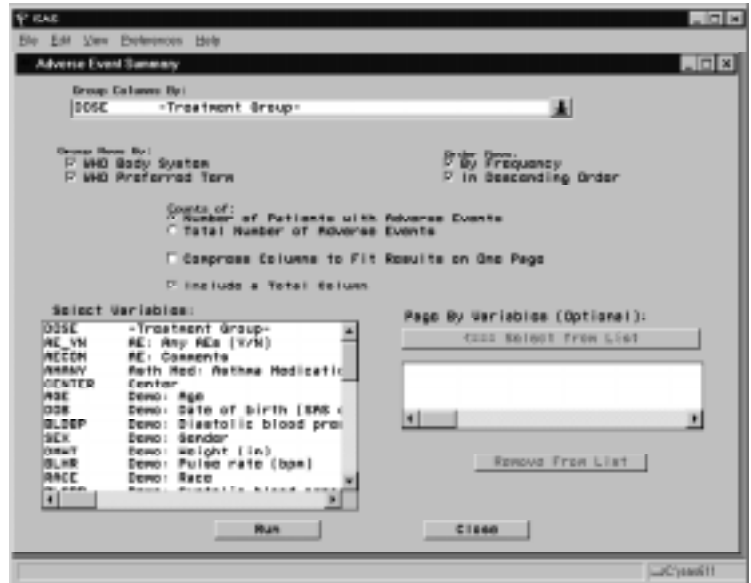
props= proc report options.

Analytical methods define how calculations are performed, statistical methods (procs and model statements), number of decimal places to display in results, etc. Significant digits are often an issue of extended discussion and it is wise to reach agreement in advance on how this will be handled.

The **PH-Clinical User Interface** is a mock-up of the PH-Clinical Template User Window which will give clinical reviewers access to the reporting tool. It describes the user-selected and optional parameters as well as the default values. There are important factors to consider when designing the PH-Clinical Template User Windows for a series of templates.

- use a standard user window size
- use consistent template title placement, font types and font sizes
- limit user selections and options to the most frequently used parameters
- use consistent descriptions of and placement of user selections and options
- identify required parameters and use defaults for required parameters

This consistency of the PH-Clinical User Interface speeds user training and facilitates ease of use for reviewers. The sample AE1 and TABLE1 PH-Clinical Template User Window illustrate these points.



The final design consideration is the **macro code design**. In the design document the macro code is outlined in an English language description and possibly a flow chart. The description includes the logical flow, procedures to use, data manipulation and external macro calls. We follow a few simple guidelines when developing report macros.

- Use structured programming technique. Strive for modular programs which will be easy to maintain and will maximize reusable code.
- Develop data-driven code, allowing the data to determine the report structure. Pre-process the dataset to create formats, labels and groupings. If possible, DO NOT hard code values, labels, column widths, etc. Parameters controlling column widths, number of columns, etc. should be optional and the report macro should be programmed to be 'smart' enough to do most table formatting without intervention.
- Use a toolkit of utility macros to develop the report macros. Maintain tight control of utility macro development, focusing on flexible, generic tools. Avoid creating a *macro-hell* development environment containing a large number of similar (redundant) or project-specific utility macros.
- Use SAS procedures to process data. For example, when calculating counts or means use Proc Means or Proc Univariate rather than data step programming. This will ensure consistency of results.
- If possible, use Proc Report for formatting and displaying final listings or tables. Proc Report is a very powerful procedure that provides many features that are hard to duplicate using Proc Print or Data _Null_ and Put statements.

SAS/PH-Clinical Considerations

A well designed report macro will work well as a stand alone procedure or within a SAS/PH-Clinical Template. There are a few points to keep in mind as the macro is being designed for use both as a stand-alone procedure and within a SAS/PH-Clinical Template. These considerations affect how specific features are programmed into the report macros.

- SAS/PH-Clinical is an interactive tool and system options can be defined in an unpredictable manner by a user/reviewer. If the report will have specific requirements for

system options (e.g. line or page size requirements) consideration will have to be made as to how the report tool will control these settings.

- PH-Clinical titles and footnotes can be defined within the template TITLES object. There are special PH-Clinical functions which can be used within titles and footnotes: open study name(s), open query name(s), user name and date/time. Consider the default settings to be assigned for PH-Clinical and how this will impact titles and/or standard headers and footers assigned by report macros.
- PH-Clinical utilizes a data dictionary which has specific requirements for data organization. Is your PH-Clinical data already in the required data structure?
- A PH-Clinical interactive template is designed for non-SAS users, such as a Clinical Reviewer. The template user window should be simple in design. The user selections and options should be limited in number and easy for reviewers to understand.

Use of report macros significantly simplifies adaptation of report programs into SAS/PH-Clinical templates (Gilbert *et al*, 1996). Powerful, interactive templates can be built by linking selections made by users in the template user window to a short program calling the report macro. Two programs are listed below. The first is the AE1 macro call outside of PH-Clinical as a stand-alone procedure. The second is the AE1 macro call within a PH-Clinical template, with descriptions of how user interface tools are linked directly to macro parameters. These programs illustrate the ease of importing report macro calls into PH-Clinical.

Stand Alone Code

```
%getdata(drg=newdrug, prt=prot9999, type=a, lname=db1);
```

```
proc sort data=db1.dose0 out=dose0;
  by sub;
run;
```

```
proc sort data=db1.ae0 out=ae0;
  by sub;
run;
```

```
data ae0; merge dose0 ae0;
  by sub;
run;
```

```
%ae1(
  in=ae0,
  title= 6 "A Basic AE1 Macro Call",
  footnote= 1 " Num Patients with an Adverse Event",
  suffix=A,
  footnum=3);
```

PH-Clinical Template Code

```
*****
* Define standard output format.
*****;
%output(type=1);

*****

Substitution field to enable use of PH
* title and footnote widget
*****;
&titles;

*****
* Select required variables:
* &dose_fmt &aewhopre &aewhobod
*****;
*****
* PH Conversion for Interactive Template:
* replace substitution fields for macro
* parameters from variable references
* to widget references.
*
* column is assigned to a var. drop down list
* count is assigned to a multichoice station
* order is assigned to a check box
* descend is assigned to a check box
* row is assigned to a check box list
* squeeze is assigned to a check box
* totalg is assigned to a check box
* pageby is a variable list & button combo
*****;
```

```
%ae1(in=&ae
  column = &column,
  count = &count
  order = &order,
  descend = &descnd,
  pat = &patid,
  row = &row,
  pageby = &pageby,
  squeeze = &squeeze,
  totalg = &totalg,
  datyes = &ae_yn= 'Y');
```

References

Gilbert, P. and Light, S. (1996) A Simple Methodology for Developing Clinical Reporting Macros. *Proceedings of the Twenty-first Annual SAS Users Group International Conference*. SAS Institute, Inc, Cary, NC pp. 1233-1235

Gilbert, P., S. Light, A. Siegel and J. Jacobsen (1996) Conversion of Report Programs to SAS/PH-Clinical Templates. *Proceedings of the Ninth Annual NorthEast SAS Users Group Conference*. SAS Institute, Inc, Cary, NC pp. 281-289

Biography

Paul Gilbert is Vice President of DataCeutics, Inc., a consulting group specializing in software solutions, system integration, programming and support in the areas of clinical data management and statistical reporting. His fifteen years of experience includes clinical data management, implementing and maintaining BBN/Clintrial, designing SAS based biostatistics reporting systems, managing SAS based NDA programming support, SAS/PH-Clinical implementation and CANDa integration.

Steve Light, a Project Manager at DataCeutics, Inc., is responsible for SAS report macro development and biostatistics/clinical SAS project management. His twelve years of experience with clinical information systems include SAS systems development and validation, NDA driven project management, SAS/PH-Clinical implementation, programming and clinical data management.



**Figure 1
AE1 Macro**

OPTIONAL HEADER TITLES ==>

DataCeutics Sample Study
Protocol Prot-9999

OPTIONAL ==> Page 1 of 3

OPTIONAL PLACEMENT ==>

Example 2.1

OPTIONAL TITLES ==>

AE1 Macro Example 1
A Basic AE1 Macro Call

OPTIONAL BY= VAR

<i>USER SELECTED SYSTEM AND/OR TERM</i>		<i>USER SELECTED COLUMN VAR</i>						<i>OPTIONAL TOTAL</i>		
		Dose Group								
WHO Body System	WHO Preferred Term	Placebo (N=8)	100 mg (N=7)	200 mg (N=16)	Total (N=31)					
<i>USER SELECTED COUNTS OF EVENTS OR PATIENTS, (N=n) AND (%)</i>										
All Systems	Any AE	7 (88%)	7 (100%)	16 (100%)	30 (97%)					
RESPIRATORY SYSTEM DISORDERS	RESPIRATORY INFECTION	5 (63%)	4 (57%)	7 (44%)	16 (52%)					
	RHINITIS	3 (38%)	1 (14%)	4 (25%)	8 (26%)					
BODY AS A WHOLE - GENERAL DISORDERS	FEVER	4 (50%)	1 (14%)	7 (44%)	12 (39%)					
	ACCIDENT AND/OR INJURY	2 (25%)	1 (14%)	.	3 (10%)					
	PAIN	1 (13%)	.	1 (6%)	2 (6%)					
	CHEST PAIN	.	.	1 (6%)	1 (3%)					
GASTRO- INTESTINAL SYSTEM DISORDERS	ABDOMINAL PAIN	2 (25%)	1 (14%)	.	3 (10%)					
	DIARRHOEA	1 (13%)	.	2 (13%)	3 (10%)					

OPTIONAL FOOTNOTES

Counts of the number of Patients with an Adverse Event

STANDARD DOCUMENTATION FOOTNOTE

Program: C:\DISCS\NEWDRUG\PROT9999\PROG\DEV\PROG\AE1.sas

Output: AE1A.lst

15DEC96 10:33

**Figure 2
TABLE1 Macro**

OPTIONAL HEADER TITLES ==>

DataCeutics Sample Study
Protocol Prot-9999

OPTIONAL ==> Page 1 of 5

OPTIONAL PLACEMENT ==>

Example 5.1

OPTIONAL TITLES ==>

TABLE1 Macro Example 1

A Basic TABLE1 Macro Call using the Medical Diagnosis Category

OPTIONAL BY= VAR

<i>USER SELECTED ROW VAR</i>	<i>USER SELECTED COLUMN VAR</i>						<i>OPTIONAL TOTAL</i>	
	Placebo		100 mg		200 mg		Total	
<i>OPTIONAL RESULT VAR</i>	(N=8)		(N=7)		(N=16)		(N=31)	
ICD-9-CM Diagnosis	<i>OPTIONAL (N=n) AND (%)</i>							
ASTHMA W/O STATUS ASTHM	9	(113%)	7	(100%)	16	(100%)	32	(103%)
OTITIS MEDIA NOS	7	(88%)	4	(57%)	9	(56%)	20	(65%)
CHRONIC RHINITIS	2	(25%)	4	(57%)	5	(31%)	11	(35%)
CARDIAC MURMURS NEC	2	(25%)	.	.	3	(19%)	5	(16%)
NASAL & SINUS DIS NEC	1	(13%)	.	.	3	(19%)	4	(13%)
ACUTE BRONCHIOLITIS	.	.	2	(29%)	1	(6%)	3	(10%)
ANAPHYL SHOCK W FOOD NOS	1	(13%)	1	(14%)	1	(6%)	3	(10%)
BRONCHITIS NOS	1	(13%)	.	.	2	(13%)	3	(10%)

OPTIONAL FOOTNOTES

STANDARD DOCUMENTATION FOOTNOTE

Program: C:\DISCS\NEWDRUG\PROT9999\PROG\DEV\PROG\TABLE1.sas

Output: TABLE1A.lst

26DEC96 16:12