Making the Most Out of Multilabel Formats

Tasha Chapman
Oregon Department of Consumer and Business Services

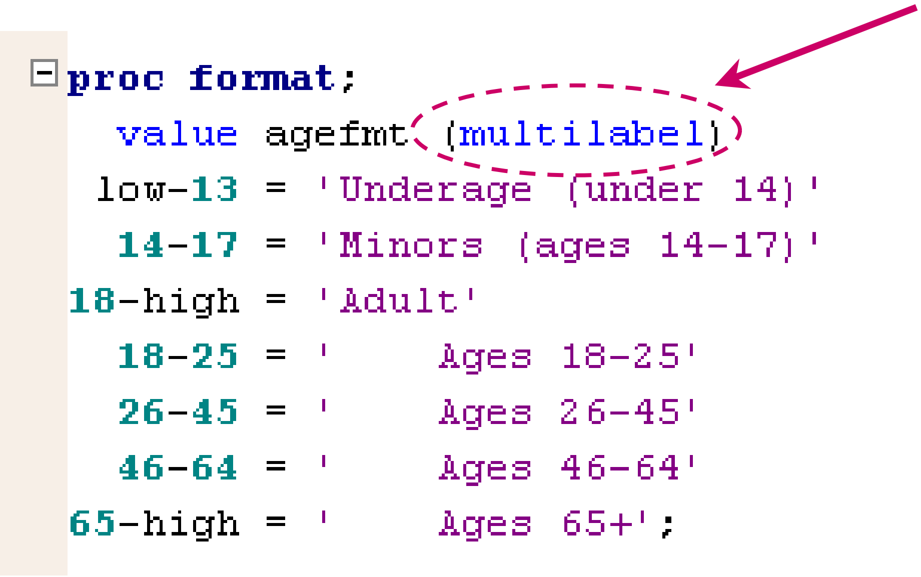
This paper will discuss how to use multilabel formats, how to control the sort order of your formats, how to print rows with missing data, and how to maintain indentations and leading spaces in ODS output.

Multilabel formats allow a SAS® user to display over-lapping ranges. Among other things, multilabel formats allow a user to create hierarchical formats, showing multiple levels of groups and sub-groups in the same table.

Multilabel formats have been available since SAS Version 8, and are currently compatible with Proc Tabulate, Proc Means and Proc Summary. Multilabel formats cannot currently be used with Proc Report (as of version 9.2). The following examples were created in SAS version 9.1.3, using multilabel formats in Proc Tabulate and output using ODS PDF (SASweb style).

Using Multilabel Formats

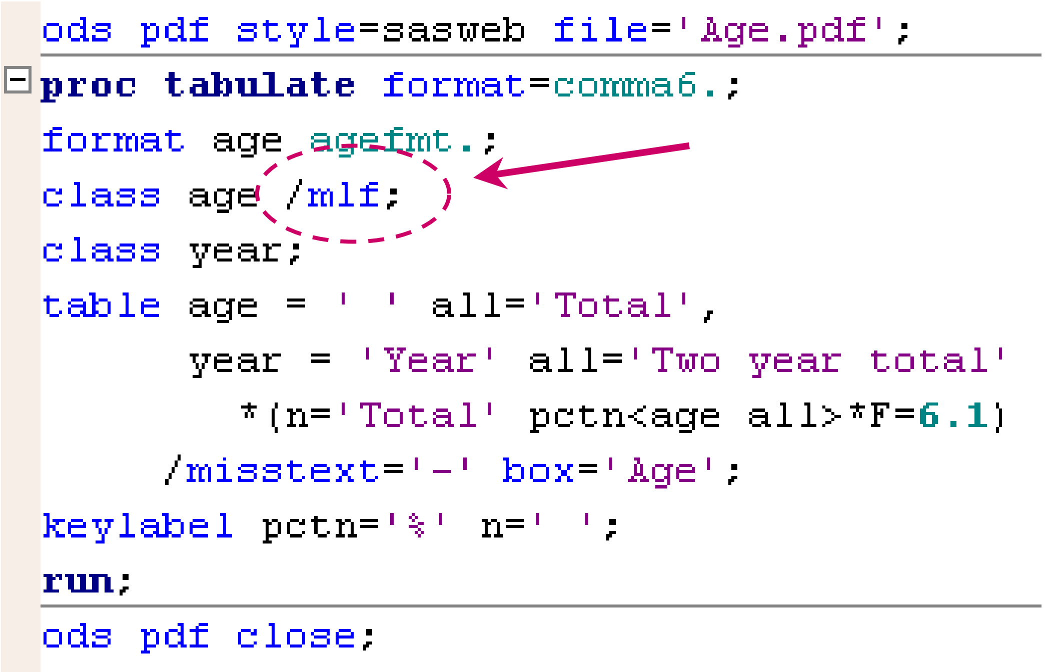
When creating our multilabel format, we need to specify in the Proc Format statement that the Value format we are creating will have both groups and sub-groups, using the **multilabel** option. Attempting to create a multilabel format without this specified option will result in an error in the log warning that there are over-lapping ranges, and the procedure will not run. This example only has a two-level hierarchy, but with multilabel formats you can have as many subgroups as you like.



```
proc format;
  value agefmt (multilabel)
 low-13 = 'Underage (under 14)'
  14-17 = 'Minors (ages 14-17)'
18-high = 'Adult'
  18-25 = '    Ages 18-25'
  26-45 = '    Ages 26-45'
  46-64 = '    Ages 46-64'
65-high = '    Ages 65+';
```

® SAS is a registered trademark of SAS Institute in the USA and other countries.
® indicates USA registration.

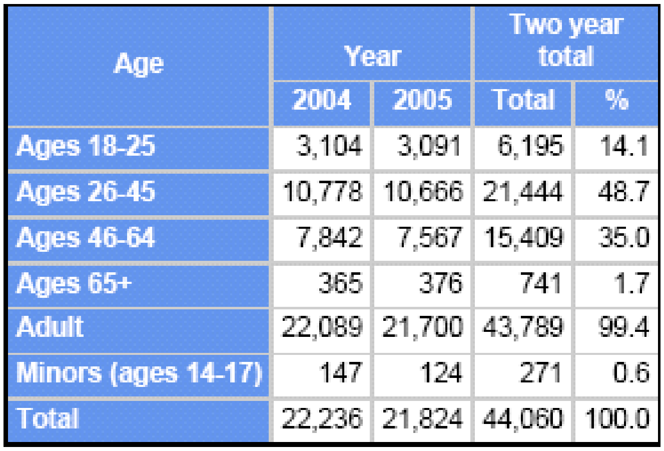
In addition to the **multilabel** option in the Format Procedure, we need a corresponding statement in the Proc Tabulate, telling SAS that the format we will be using is a multilabel format. In Proc Tabulate, we put this syntax in the class statement:



```
ods pdf style=sasweb file='Age.pdf';
proc tabulate format=comma6.;
format age agefmt.;
class age /mlf;
class year;
table age = ' ' all='Total',
      year = 'Year' all='Two year total'
        *(n='Total' pctn<age all>*F=6.1)
     /misstext='-' box='Age';
keylabel pctn='%' n=' ';
run;
ods pdf close;
```

The **mlf** option in Proc Tabulate (short for "multilabel format") tells SAS that the incoming format will have over-lapping ranges. Without this statement, the tabulate procedure may ignore some of the over-lapping ranges in the format, often without an error in the log or any other warning. If your resulting output is missing some categories, the most likely cause is a missing mlf option.

Example 1. Multilabel format



Age	Year		Two year total	
	2004	2005	Total	%
Ages 18-25	3,104	3,091	6,195	14.1
Ages 26-45	10,778	10,666	21,444	48.7
Ages 46-64	7,842	7,567	15,409	35.0
Ages 65+	365	376	741	1.7
Adult	22,089	21,700	43,789	99.4
Minors (ages 14-17)	147	124	271	0.6
Total	22,236	21,824	44,060	100.0

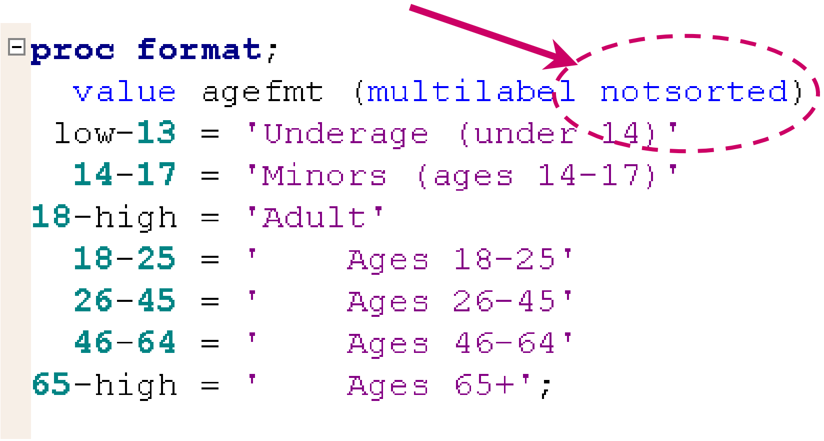
Example 1 shows the resulting output, using the multilabel format option. Both our major group ("Adult") and sub-groups ("Ages 18-25"; "Ages 26-45"; "Ages 46-64"; and "Ages 65+") have been printed in the resulting table.

However, there are still some problems with our output. The format has been sorted alphabetically[1]. One category – "Underage" – has been left off of the table because it had no observations. Finally, our leading spaces have been truncated making it difficult to tell which categories are major groups and which are sub-groups.

[1] "Ages" came before "Adult" because of the leading spaces in our original format

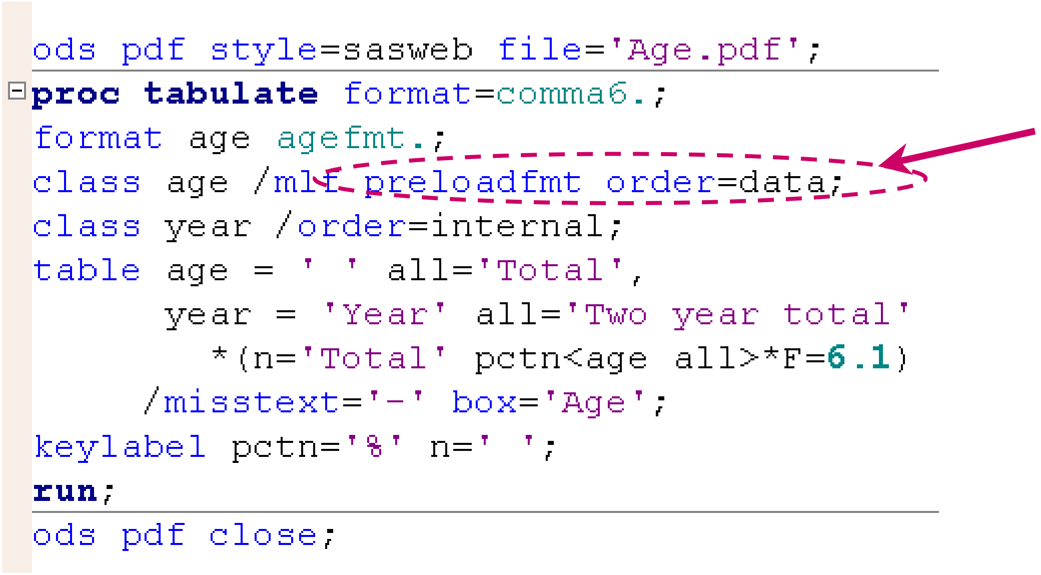
Maintaining the Sort Order of the Format

To fix the sorting issue, we first need to use the **notsorted** option in Proc Format. This option maintains the sort order as specified in the format and can be used in conjunction with the multilabel format option starting in Version 9.1. The notsorted option can also be used alone for non-multilabel formats.



```
proc format;
  value agefmt (multilabel notsorted)
 low-13 = 'Underage (under 14)'
  14-17 = 'Minors (ages 14-17)'
18-high = 'Adult'
  18-25 = '    Ages 18-25'
  26-45 = '    Ages 26-45'
  46-64 = '    Ages 46-64'
65-high = '    Ages 65+';
```

In the Proc Tabulate statement, we need to use two additional options in the class statement: **preloadfmt** and **order=data**. These specified options will tell the Tabulate Procedure to honor the order of our format, rather than sorting alphabetically by default.



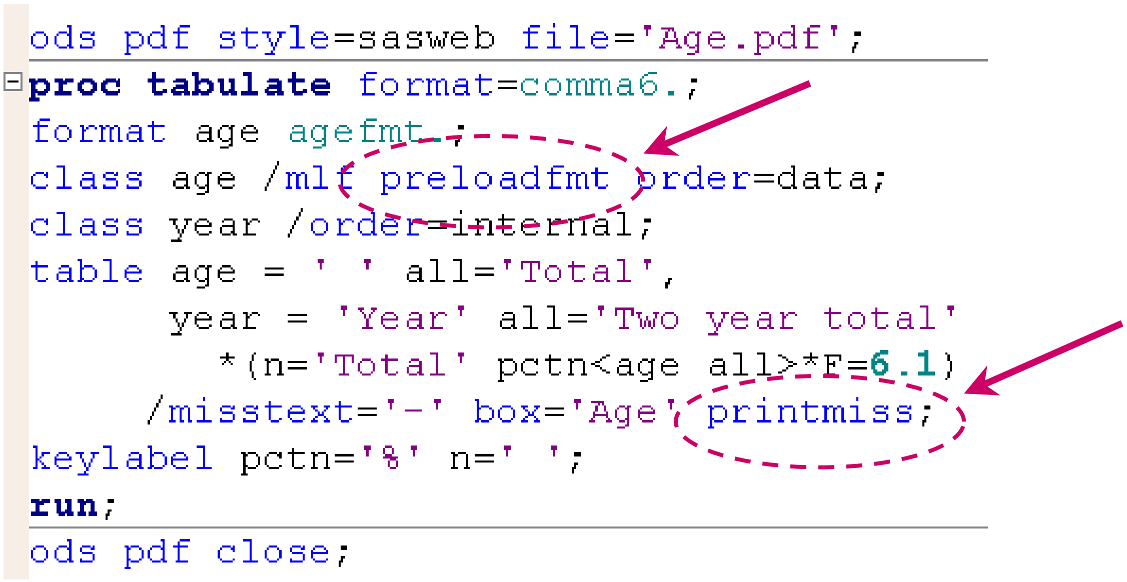
```
ods pdf style=sasweb file='Age.pdf';
proc tabulate format=comma6.;
format age agefmt.;
class age /mlf preloadfmt order=data;
class year /order=internal;
table age = ' ' all='Total',
      year = 'Year' all='Two year total'
        *(n='Total' pctn<age all>*F=6.1)
     /misstext='-' box='Age';
keylabel pctn='%' n=' ';
run;
ods pdf close;
```

In this particular example, I have also added an additional **order=internal** statement in the class variable “year.” Order=internal tells SAS to order the year columns according to the internal date value. I have intentionally specified it here to prevent the order of the “age” variable from interfering with the order of the “year” variable.

Internal is generally the best ordering option to use when working with dates. In our example, “year” is actually a date field formatted with the SAS supplied date format **year4.**, and since 2004 will always come before 2005, regardless of whether we are using the interval value or the formatted value of the date for sorting, it’s a moot issue. However, if we were to use a different date format, such as **monname.**, we would run into trouble if April came before January due to an alphabetical sorting of the formatted values.

Rows with Missing Data

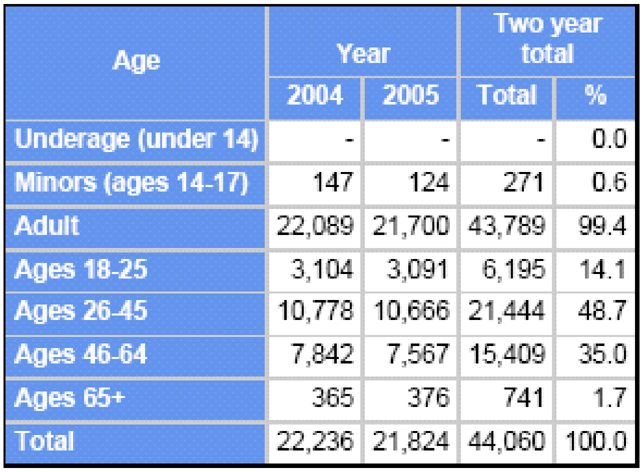
In our format we have a category for “Underage,” which includes all records with ages younger than 14 years old. However, the dataset that we are working with doesn’t have any observations with ages younger than 14. By default, Proc Tabulate eliminates all rows or columns with no observations, even if they are in the format. We can tell Proc Tabulate to include the “Underage” row by using two simple options.



```
ods pdf style=sasweb file='Age.pdf';
proc tabulate format=comma6.;
format age agefmt.;
class age /mlf preloadfmt order=data;
class year /order=internal;
table age = ' ' all='Total',
           year = 'Year' all='Two year total'
             *(n='Total' pctn<age all>*F=6.1)
        /misstext='-' box='Age' printmiss;
keylabel pctn='%' n=' ';
run;
ods pdf close;
```

First, we use the **preloadfmt** option in the class statement. The preloadfmt statement will create a header for all class variables (found in the format) even when the values do not appear in the dataset. In our example, preloadfmt was already in the class statement for sorting purposes. Second, we add the **printmiss** option in the table statement. The printmiss statement tells Proc Tabulate to print all categories of age – even those with missing data – in our table.

Example 2. Multilabel format with notsorted, preloadfmt, and printmiss



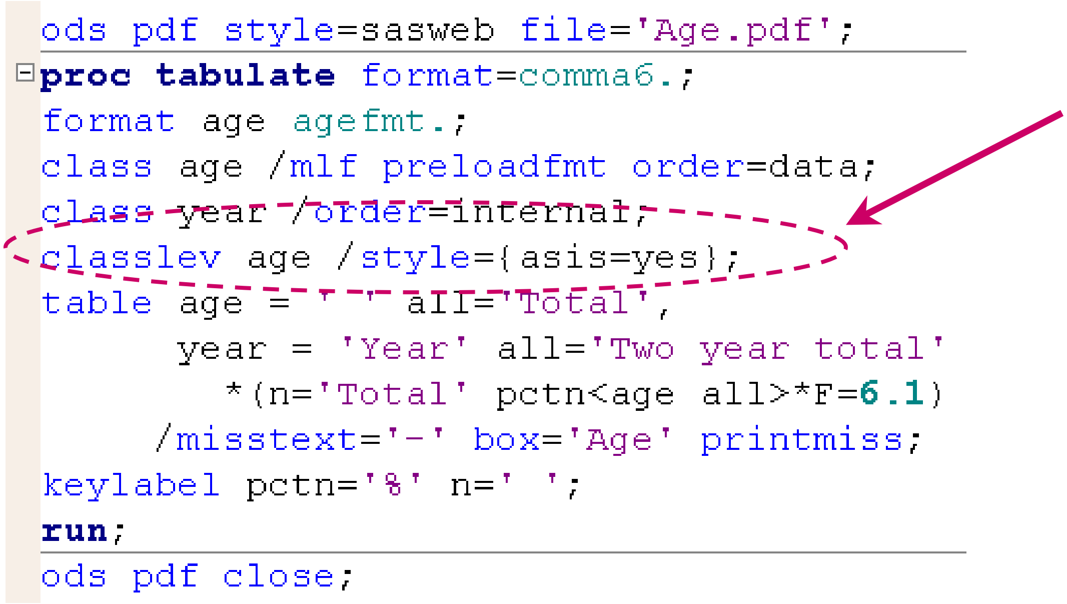
Age	Year		Two year total	
	2004	2005	Total	%
Underage (under 14)	-	-	-	0.0
Minors (ages 14-17)	147	124	271	0.6
Adult	22,089	21,700	43,789	99.4
Ages 18-25	3,104	3,091	6,195	14.1
Ages 26-45	10,778	10,666	21,444	48.7
Ages 46-64	7,842	7,567	15,409	35.0
Ages 65+	365	376	741	1.7
Total	22,236	21,824	44,060	100.0

In Example 2, we can see our results so far. Our categories are listed by descending age, the same order as our format, and the “Underage” category has been printed, even though there are no data points in that category.

There is only one issue left to fix – the leading spaces in the formatted labels of our sub-groups.

Maintain leading spaces

In Proc Tabulate, and many other procedures, style statements known as style overrides can be used to override specific attributes of the default style (or the specified style in the ODS statement). These style overrides will change the appearance in a particular area of a report. In other words, we can use a style override to modify one small aspect of the output without using the cumbersome Proc Template procedure to change the entire style template. In our example, we want to maintain the leading spaces in our format. By default, SAS will truncate these leading blanks, but one simple style override can change this.



```
ods pdf style=sasweb file='Age.pdf';
proc tabulate format=comma6.;
format age agefmt.;
class age /mlf preloadfmt order=data;
class year /order=internal;
classlev age /style={asis=yes};
table age = ' ' all='Total',
      year = 'Year' all='Two year total'
        *(n='Total' pctn<age all>*F=6.1)
     /misstext='-' box='Age' printmiss;
keylabel pctn='%' n=' ';
run;
ods pdf close;
```

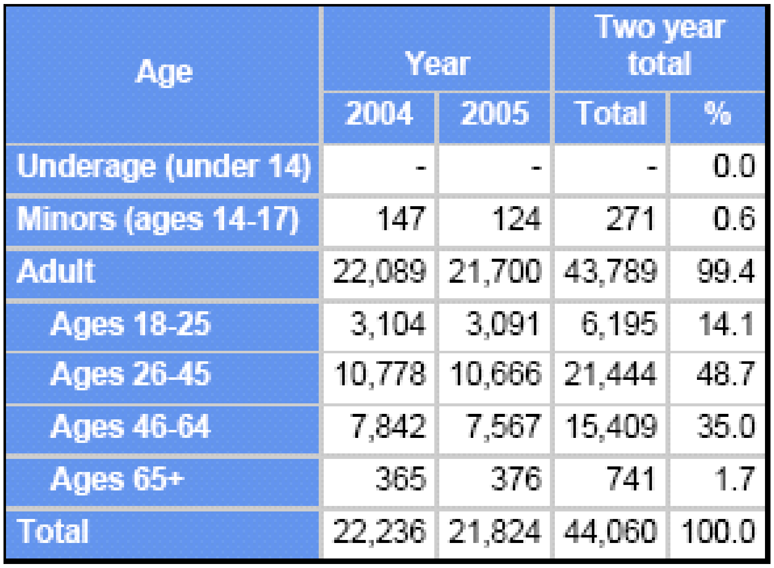
Where you place the style override statement will determine which part of the output table will be effected. In this instance we want to change the output style of the labels for the rows. This particular part of Proc Tabulate is controlled through the classlev statement. In the classlev statement, we first list the variable for which want to specify the style element ("age"), and then specify the style. In this case we'll be using the **asis** style override. **ASIS=yes** will tell Proc Tabulate that we want to maintain the leading spaces in our format (in other words, "leave them as is").

If the listing destination is still open when you run this procedure, you may find that the output in the output window does not have the leading spaces. This is because the style overrides will only work successfully in certain ODS formats. According to the SAS documentation for Version 9.1.3, the asis option is compatible with ODS HTML, RTF, PDF, PCL, and PS. However, it should be noted that as with all style overrides, each destination may implement the asis style override in a unique way. The resulting output may appear differently in HTML than it does in RTF or PDF; the same output file may even appear differently depending on which browser or application the file is opened in. Because PDF documents tend to be static in their appearance, I have traditionally had the most success using asis with ODS PDF, but you are free to experiment with other ODS destinations.

Final results

Example 3 shows the final resulting output from our Proc Tabulate. Our finished table has groups and sub-groups; the rows are sorted per our specifications in the format; all rows in the format are displayed, including those with missing data; and we have indented certain rows to make it easy to distinguish between major groups and sub-groups.

Example 3. Final results



Age	Year		Two year total	
	2004	2005	Total	%
Underage (under 14)	-	-	-	0.0
Minors (ages 14-17)	147	124	271	0.6
Adult	22,089	21,700	43,789	99.4
Ages 18-25	3,104	3,091	6,195	14.1
Ages 26-45	10,778	10,666	21,444	48.7
Ages 46-64	7,842	7,567	15,409	35.0
Ages 65+	365	376	741	1.7
Total	22,236	21,824	44,060	100.0

You want to...	Use this...	Procedure	Where to put it
Have groups and sub-groups	`multilabel` `mlf`	Proc Format Proc Tabulate	Value statement Class statement

Control the sort order of your format	`notsorted`
	`preloadfmt`
	`order=data`
		Proc Format
		Proc Tabulate
		Proc Tabulate
			Value statement
			Class statement
			Class statement

Print rows with no observations	`preloadfmt` `printmiss`	Proc Tabulate Proc Tabulate	Class statement Table statement
Maintain leading spaces in your format	`style={asis=yes}`	Proc Tabulate	Classlev statement

Special thanks to Eric Gebhart for his comments and suggestions while preparing this paper.

Comments and questions are always welcome. The author can be contacted at:
Tasha.L.Chapman@state.or.us