The SAS® System
Release 6.11 (TS040 and above)
MIPS ABI

Please Read Before Beginning Installation

Introduction

Alert Notes list problems that you need to be aware of before installing or using this software. Should you need assistance with the software, we ask that only the SAS Installation Representative or SAS Support Consultant call our Technical Support Division. Sites in the U.S. and Canada may call (919) 677-8008. Other sites should contact their SAS Installation Representative or SAS Support Consultant for the nearest SAS Institute office.

Installation Issues

- Although the latest release of the SAS System for most operating systems is Year 2000 compliant, it is crucial that you read this information and take appropriate action to make sure that your programs and applications that use the SAS System will process dates correctly before, during, and after the Year 2000.

  SAS software (after Release 6.04) uses the `YEARCUTOFF=` option to determine what century prefix a two-digit year will be associated with. For example, if you specify `YEARCUTOFF=1900`, all two-digit years processed by SAS applications will be assumed to be between 1900 and 1999; if `YEARCUTOFF=1950` is specified, all two-digit years between 50 and 99 are assumed to be in the 1900s, while all two-digit years from 00 to 49 are assumed to be from 2000 to 2049.

  For Version 6 SAS software (after Release 6.04), the default value of `YEARCUTOFF=` is 1900, unless it has been reset by SAS support personnel at your site. This means that all two-digit years processed by SAS software are assumed to be in the 1900s and processing any date information with values greater than December 31, 1999 may produce incorrect results if they are represented with two-digit years. For the Version 8, Developer's Release of the SAS System, the default value of `YEARCUTOFF=` is 1920. To provide for correct processing of two-digit years by SAS software, you should determine the value of the `YEARCUTOFF=` option on your system and modify it if necessary. To determine the value of the `YEARCUTOFF=` option, simply invoke the SAS System and submit the following statements:

```
proc options option=yearcutoff;
run;
```

The values of the `YEARCUTOFF=` option will be displayed in the SAS `LOG` window. If the `YEARCUTOFF=` option is set to 1900, we suggest modifying it to a value between 1920 and 1950. The optimum value will depend on the range of dates that you typically process with your SAS applications. If you do not anticipate processing date values greater than 2020, you can set `YEARCUTOFF=1920`; if your SAS applications process dates greater than 2020, you may want to set `YEARCUTOFF=` to a higher value, such as 1930 or 1950. The process for changing the default value of `YEARCUTOFF=` (or any system option) depends on your specific operating system - consult the **SAS Companion** for your operating system or the SAS Help facility for specific details.

We also recommend that SAS Installation Representatives and SAS Software Consultants make all SAS software users at their site aware of the default `YEARCUTOFF` settings for Version 6 and the Version 8, Developer's Release. An easy way to do this is to display the information at the top of the SAS `LOG` window using the `NEWS` system option. See the **SAS Companion** for your operating system or the SAS Help facility for specific details on using the `NEWS` option.

For additional details on how the `YEARCUTOFF=` option works and how to determine the optimum setting for the option, refer to the document **A Guide to the YEARCUTOFF= Option, TS-618**, which is available on our Web site at:

http://www.sas.com/techsup/download/technote/ts618.html

If you do not have access to our Web site, you can obtain a copy of the document by contacting our Technical Support Division at (919) 677-8008. (Those of you outside the United States or Canada should contact your local SAS Institute office or subsidiary.) As always, we encourage you to use the latest version of the SAS System. For complete details on the Year 2000 compliance of SAS software products, as well as information and resources for testing your SAS applications for Year 2000 compliance, refer to our Year 2000 compliance Web page at:

http://www.sas.com/y2k

- Before installing Release 6.11 of the SAS System from TS051 media, please refer to the **Addendum to the Installation Instructions for the SAS System, Release 6.11 (TS051) for MIPS ABI and ABI+ for Intel Architecture** to ensure the proper installation procedure is followed for your environment.

- If you are installing an add-on product to a system that has already been upgraded to (TS051), you *must* re-apply (TS051) after installing the new product. Failure to do so will result in a SAS System installation with mismatched updates. Unpredictable results will occur when running from such an installation. To re-apply SAS Notes (TS051), choose the following path from the `sasmanager` panels:

```
Invoke Custom Installation
Invoke SAS Installation Utilities
Apply Special Tech Support Fixes
```

  This will re-apply the SAS Notes fixes only for the new, add-on products.

- If you choose to install Korean-translated files, the install only loads DBCS-translated files onto your system. After these files are loaded on your system and you exit the install, add the following line to the CONFIG file, `config.sas611`, located in the `SASROOT` directory (e.g., `/usr/local/sas611/config.sas611`):

```
dbcslang korean
```

  Adding the `-dbcslang` option to the CONFIG file sets up the language type for the installed Korean translation.

Base SAS Software

- Release 6.11 of the SAS System for MIPS ABI is currently not supported on the following systems.

Vendor	Operating System
Control Data Systems	EP/IX, Version 2.2.1
NEC	EWS-UX/V (Release 4.2) Release 9.1 on R3000 processors
Sony	NEWS-OS, Version 6.1

- When running on a Sony system, Release 6.11 of the SAS System for MIPS ABI does not support Q150 multi-volume tapes.

- When running on a NEC system, Release 6.11 of the SAS System for MIPS ABI displays incorrect tape numbers when reading Q150 multi-volume tapes. Even though incorrect tape numbers are displayed, the user can successfully read the data by inserting the tapes in the proper sequence.

- When running on a NEC system with the UX/4800, Release 11.1 operating system, the SAS System requires patches NECmb0098, NECmb0334, and NECmb0473 to be applied to UX/4800, Release 11.1. Contact NEC for these patches.

- When running on a SGI system, Release 6.11 of the SAS System for MIPS ABI does not support reading Q150 multi-volume tapes created with Release 6.10 of the SAS System for MIPS ABI.

- When creating SAS data sets greater than 2 GB on an SGI system utilizing SGI's XFS file system, users must specify the `-largefile sasvlfs` experimental system option.

  Without this option, SAS files that grow larger than 2 GB may become corrupted without any warning. This option can be specified at SAS invocation or in the `!SASROOT/config.sas611` file.

- In some circumstances, `PROC TABULATE` may calculate the `PCTSUM` or `PCTN` statistics incorrectly. The problem is related to the positioning of other columns in the table. If multiple percentage statistics are requested for the same crossing of variables, but these expressions are not adjacent to one another in the `TABLE` statement, the resulting statistics will be incorrect. For example, the following `TABLE` statement produces incorrect values:

  ```
  TABLE A*B*PCTSUM A*C*PCTSUM A*B*PCTN;
  ```

  Notice that there are two expressions that both contain `A` and `B` crossed with a percentage. Because these expressions are not next to one another, the values in the `A*C*PCTSUM` and `A*B*PCTN` columns are not computed correctly. To circumvent the problem, re-arrange the columns of the table so all crossings containing the same set of variables are grouped together, for example:

  ```
  TABLE A*B*PCTSUM A*B*PCTN A*C*PCTSUM;
  ```

  SAS Note V6-TABULATE-B424 documents this problem.

- Beginning in Release 6.07 of the SAS System, `PROC MEANS` and `PROC SUMMARY` compute incorrect confidence limits `CLM LCLM UCLM` in the presence of a `WEIGHT` statement. Currently, no circumvention for this problem is available.

  SAS Note V6-MEANS-C132 documents this problem.

- When the SAS System is invoked on a MIPS chip-set that is in performance mode, the SAS System may have problems recovering from floating point exceptions. The MIPS R8000 chip-set is currently the only MIPS chip-set that runs in performance mode by default. In performance mode, floating point interrupts are asynchronous and the reported program counter is meaningless.

  When running the SAS System on the MIPS R8000 chip-set, the SAS System must be run in precise floating point mode. The SAS System must be invoked via the `fpmode precise` command. Invoking `fpmode precise <_sas_command_>` executes the `_sas_command_` in precise floating point mode. As a suggestion, you may want to create an alias prefacing the SAS command with `fpmode precise` to ensure that the SAS System is run in precise floating point mode. For example:

  For a Bourne Shell or Korn Shell:

  ```
  alias sas='fpmode precise /usr/local/sas611/sas'
  ```

  For a C Shell:

  ```
  alias sas 'fpmode precise /usr/local/sas611/sas'
  ```

  For more information on the `fpmode` command, refer to the operating system manpage for `fpmode`.

  SAS Note V6-SYS.SYS-B851 documents this problem.

- Using the `MODIFY` statement with the `UNIQUE` option to provide a consecutive `KEY=` value whose matching records were deleted in prior iterations causes the SAS System to incorrectly remove the next logical, non-match record in the index. A subsequent read of a consecutive `KEY=` value on a non-match record incorrectly produces a `_IORC_` value of zero. The following DATA step replicates the problem:

  ```
              master              tran
        OBS    X    Y      OBS    X     Y
         1     1    3       1     1     2
         2     2    4
         3     3    5
  data master ;
  set tran ;
  do i = 1 to 3 ;
     modify master key = x / unique ;
     if _iorc_ = %sysrc(_sok) then remove ;
     else _error_ = 0 ;
  end ;
  run ;
  ```

  SAS Note V6-SYS.DATA-C121 documents this problem.

- If you scroll through a multi-page report in the `PROC REPORT` window using the `PAGE` and `PAGEBACK` commands repeatedly, the values of some computed columns may be recalculated incorrectly.

  It is possible to correct the values by returning to the first page of the report and then using the `PAGE` command to scroll forward again. The problem only occurs when you scroll forward and back several times.

  SAS Note V6-REPORT-C829 documents this problem.

- In Version 6 of the SAS System, an `OUTER JOIN` may produce incorrect results based on the underlying subquery. There are no error messages or warnings to indicate a problem.

  An example of this type of `OUTER JOIN` is:

  ```
  PROC SQL;
   SELECT *
    FROM TABLE_A RIGHT JOIN
     (SELECT * FROM TABLE_B
      WHERE 1 ^=(SELECT COUNT(*) FROM TABLE_A
       WHERE TABLE_A.VAR1 = TABLE_B.VAR1
        AND TABLE_A.VAR2 = TABLE_B.VAR2)
         ) AS XYZ
          ON TABLE_A.VAR1 = TABLE_B.VAR1;
  ```

  A circumvention is to create a table of the subquery and then reference the table in the subsequent query that performs the `OUTER JOIN`.

  SAS Note V6-SQL-A586 documents this problem.

- In Version 6 of the SAS System, the `SQL` procedure may return the incorrect number of rows. This is only likely to occur in a situation where:

  1. The query is a join of two or more tables/views.
  2. At least one of the tables is a view or SAS data set access by the sequential engine such that `SQL` cannot determine the number of rows it contains prior to the join.
  3. Part of the `WHERE` clause involves the equality comparison of two or more variables from that same sequential data set or view.

  ```
  proc sql;
         select * from VIEW1 as a, VIEW2 as b
         where a.KEY1=b.KEY2 and b.X=b.Y;
                                 -------
  ```

  SAS Note V6-SQL-C979 documents this problem.

SAS/AF Software

- In a Data Form or Data Table object within a SAS/AF FRAME entry, if you make changes to the current row and then move to a different row using the `_GOTO_ABSOLUTE_ROW_` method, the changes from the previous row may be copied to the row to which you moved instead of the previous row. To circumvent, execute the `_UPDATE_ROW_` and `_UNLOCK_ROW_` methods before the `_GOTO_ABSOLUTE_ROW_` method.

  Within the Data Form object, you may see the same problem if you make a change to the current row and then scroll to a different row by issuing the number of the row as a command.

  SAS Note V6-AF-B944 documents this problem.

- With the SAS/AF graphics object, all horizontal bar charts (2D and 3D) that have a group specified should be verified by the application developer. Bars can be missing due to the limited vertical space left after titles, footnotes, and legends have been drawn. The problem is likely to occur in applications created with extremely small region sizes or a large number of groups.

  To avoid this problem, application developers should limit the number of groups they allow their application users to specify and recommend they use a larger graph region when grouping is allowed.

  SAS Note V6-AF-C264 documents this problem.

- The value of a numeric column whose length is less than 8 may be updated incorrectly in a Data Form or Data Table object within a SAS/AF FRAME entry. The value for this column may change as you simply scroll through the rows in edit mode. To circumvent the problem, change the length of numeric variables to 8 when they will be linked to columns in a Data Form or Data Table object.

  SAS Note V6-AF-B737 documents this problem.

SAS/CONNECT Software

- When using the TCP access method to connect to a UNIX spawner, you must use a two-level *`machine.service`* name, where *`machine`* is the node name of the node on which the spawner is running and *`service`* is the value of the `-service` option specified when you invoke the spawner program.

  If you are using a `signon` command, you cannot specify a two-level name from the command line. You can assign the two-level name to a macro variable and then use the macro variable in the `signon` command as follows:

  1. Submit the following in the `Program Editor` window:

     ```
     %let host1=machine service;
     ```

  2. Enter the following from the command line:

     ```
     signon host1
     ```

  Or, you can use an `options` statement to assign the two-level name to the `REMOTE=` option and then execute the `signon` command as follows:

  1. Submit the following in the `Program Editor` window:

     ```
     options remote=machine.service;
     ```

  2. Enter the following from the command line:

     ```
     signon
     ```

  The field for the remote session id (`REMOTE=` value) in both the `PMENU SIGNON` menu and the `Edit Remote Configuration` screen of SAS/ASSIST software are limited to eight characters. Therefore, you must assign the two-level name to a macro variable before attempting to signon through either of these two methods.

- Data set corruption may occur on a remote SAS/CONNECT single user server or on a SAS/SHARE server if the following conditions apply.

  1. You are using a remote SAS/CONNECT single user server or a SAS/SHARE server on a Windows 95 operating system with the SAS System, Release 6.11 (TS040 or higher).
  2. You are using the APPC communications access method.
  3. You are utilizing Microsoft's SNA Server as the underlying APPC protocol stack. In particular, you are using Version 2.11+ of the Windows 95 client software.
  4. The Windows 95 client software is configured to use TCP/IP as the LAN transport.

  The only verified circumvention is to configure the Windows 95 client software to use IPX/SPX LAN transport, instead of TCP/IP.

  SAS Note V6-RLIBRARY-C655 documents this problem.

SAS/ETS Software

- If you use GMM to estimate the parameters of a model in which a hard-coded negative sign is associated with the intercept term, such as:

  ```
  y = -a + b*x;
  ```

  then PROC MODEL may either return incorrect results or have difficulty converging to a solution.

  To circumvent the problem, reparameterize the model specification so the intercept term does not have a negative sign associated with it.

  SAS Note V6-MODEL-C938 documents this problem.

- In PROC MODEL, if a Lagrange Multiplier test statistic is requested on the TEST statement with either the LM or ALL options and the estimation method specified is 2SLS, 3SLS, or GMM, the Lagrange Multiplier test statistic and its associated p-value are incorrect. If the OUT= option is also used on the TEST statement, then the incorrect results are output to the OUT= data set. To circumvent the problem, request a likelihood ratio test.

  SAS Note V6-MODEL-B973 documents this problem.

- In PROC MODEL, when a Wald statistic is requested on the TEST statement, either by default or explicitly with the WALD or ALL option, and a system of equations is being estimated, the WALD statistic and its associated p-value are incorrect. This is regardless of the estimation method specified. If the OUT= option is also used on the TEST statement, then the incorrect results are output to the OUT= data set. To circumvent the problem, request a likelihood ratio test.

  SAS Note V6-MODEL-B987 documents this problem.

- When PROC EXPAND is used to interpolate missing values for some of the variables in a data set, all variables not processed by a CONVERT statement are copied from the input data set to the OUT= data set.

  However, if the ID variable has missing values at the beginning or end of the data set, then the range of observations output to the OUT= data set should be truncated. The copy of the input observations for the variables not processed by a CONVERT statement does not take this into account; therefore, when missing values occur at the beginning or end of the file, then wrong values are copied to the OUT= data set.

  To circumvent the problem, use a WHERE clause or statement to subset the data so observations with missing values for the ID variable are omitted.

  SAS Note V6-EXPAND-C367 documents this problem.

- The DATASOURCE procedure may return incorrect information for the OUTEVENT data set when accessing CRSP character or binary data. Information may be invalid for the distribution, share, delisting, and Nasdaq event records.

  When the problem occurs, a company that does not have event records will be assigned event records from a preceding company. The resulting invalid data may not be obvious.

  This problem has been reported for CRSP monthly character data. However, the problem may appear in other CRSP file formats (character or binary, daily or monthly).

  Currently, there is no known circumvention for this problem.

  SAS Note V6-DATASOURCE-C484 documents this problem.

- The compiler used in PROC MODEL will compute incorrect analytic derivatives when a SUM statement is used in the model specification. The SUM statement is often used in a summation DO loop. For example, the following code would return an incorrect derivative for y:

```
y=0;
do i=1 to 3;
   y + x + a;
end;
```

  To circumvent the problem, replace the SUM statement with an assignment statement to define the summation. The above example would be modified as the following:

```
y=0;
do i=1 to 3;
   y=y + x + a;
  end;
```

  SAS Note V6-MODEL-C784 documents this problem.

SAS/FSP Software

- If you edit a character variable whose value cannot be entirely displayed in the FSVIEW window because the width of the variable is longer than the width of the FSVIEW window, the updated data value saved to the data set may be truncated to only those characters that were displayed in the FSVIEW window.

  To circumvent the problem, use the FSEDIT window to edit these character values.

  SAS Note V6-FSVIEW-C730 documents this problem.

SAS/OR Software

- The compiler used in PROC NLP will compute incorrect analytic derivatives when a SUM statement is used in the model specification. The SUM statement is often used in a summation DO loop. For example, the following code would return an incorrect derivative for y:

```
y=0;
do i=1 to 3;
   y + x + a;
end;
```

  To circumvent the problem, replace the SUM statement with an assignment statement to define the summation. The above example would be modified as the following:

```
y=0;
do i=1 to 3;
   y=y + x + a;
end;
```

  SAS Note V6-NLP-C785 documents this problem.

SAS/QC Software

- If data in the key cell (top left cell) of a comparative histogram are outside the range of midpoints specified with a `MIDPOINTS=` option on a `COMPHISTOGRAM` statement, then these outlying points will be missing from the plot in the key cell. This does not happen when cells other than the key cell contain data beyond specified midpoints. In this situation, the procedure correctly extends the midpoint list to accommodate the data ranges in all cells. Note that if outliers occur in the key cell and non-key cells, then the procedure will only extend the axis enough to accommodate the non-key cell data and points may still be missing from the key cell.

  To circumvent this problem, specify midpoints that span the range of data values in the key cell or use the default horizontal axis scaling by omitting the `MIDPOINTS=` option.

  SAS Note V6-CAPABILITY-C519 documents this problem.

- The standard errors for the parameter estimates in the XADX menu system are incorrect. The reported standard errors are for parameter estimates associated with a different coding than the ones presented in the table. The standard errors that are printed are consistently off by a factor of sqrt(2) in the `Fit`, `Response Calculator`, and `Report` windows. Note that only the standard errors are incorrect; the parameter estimates, t-statistics, and p-values are all correct.

  SAS Note V6-ADX-G125 documents this problem.

SAS/SHARE Software

- Data set corruption may occur on a remote SAS/CONNECT single user server or on a SAS/SHARE server if the following conditions apply.

  1. You are using a remote SAS/CONNECT single user server or a SAS/SHARE server on a Windows 95 operating system with the SAS System, Release 6.11 (TS040 or higher).
  2. You are using the APPC communications access method.
  3. You are utilizing Microsoft's SNA Server as the underlying APPC protocol stack. In particular, you are using Version 2.11+ of the Windows 95 client software.
  4. The Windows 95 client software is configured to use TCP/IP as the LAN transport.

  The only verified circumvention is to configure the Windows 95 client software to use IPX/SPX LAN transport, instead of TCP/IP.

  SAS Note V6-RLIBRARY-C655 documents this problem.

SAS/STAT Software

- In `PROC MIXED`, incorrect predicted values can be returned when using a `RANDOM` statement with the `GROUP=` option or when using a `RANDOM` and a `REPEATED` statement with completely different (non-nested) `SUBJECT=` effects.

  SAS Note V6-MIXED-B925 documents this problem.

- In PROC MIXED, an incorrect likelihood value will be printed for models that use both the REPEATED statement and the RANDOM statement with the /G option when the estimated G matrix is singular. This incorrect value will affect the AIC and the SBC as well. To circumvent the problem, remove the /G option from the RANDOM statement and the correct value for the likelihood statistic will be displayed.

  SAS Note V6-MIXED-B580 documents this problem.

- In PROC MIXED, the DDFM=SATTERTH option will produce incorrect degrees of freedom for models involving a REPEATED statement. This error will affect the p-values as well.

  SAS Note V6-MIXED-B724 documents this problem.

- In PROC MIXED, using the V= option with the SUBJECT= option on the RANDOM statement will cause the procedure to print the incorrect values for the log-likelihood based statistics and for the residual variance estimate. Correct values for these statistics can be obtained by rewriting the RANDOM statement without the SUBJECT= option.

  SAS Note V6-MIXED-C252 documents this problem.

- PROC MIXED can potentially produce incorrect zero variance component estimates with the default METHOD=REML. If PROC MIXED computes the variance component of the effect to be between 1e-16 and 1e-6 at any time during the Newton-Raphson iteration process, the procedure may treat that variance component as constrained to be zero. This behavior is rare and will not happen for all models with zero variance components. You can check the iteration history with the ITDETAILS option in the PROC MIXED statement or the variance component estimates can be checked with PROC VARCOMP using METHOD=REML to see if you have encountered the problem.

  This problem is generally caused by a poor selection of starting values. To circumvent the problem, use the PARMS statement with the OLS option to change the starting values or provide new nonzero starting values with the PARMS statement.

  SAS Note V6-MIXED-C292 documents this problem.

- In PROC MIXED, the RATIOS column in models with a RANDOM statement and TYPE=FA(0) will be incorrect. The square root of the residual variance is used to calculate the ratios, rather than the residual variance itself. To circumvent this problem, use the NOPROFILE option on the PROC MIXED statement.

  SAS Note V6-MIXED-C661 documents this problem.

- In PROC MIXED or PROC GLM, if the LSMEANS are correlated, then the p-values reported in the PDIFF table with ADJUST=SIMULATE or ADJUST=DUNNETT may be incorrect. This behavior will only happen in rare circumstances.

  SAS Note V6-SYS.PROC-C298 documents this problem.

- If initial parameter values are input using the INEST= option and there is a linear dependency among the columns of the design matrix, PROC LOGISTIC will issue a NOTE in the output indicating that the linear dependency exists and that parameters are set to zero as a result. However, the parameter estimates table may show nonzero values for these parameters even though their degrees of freedom are zero. Also, X*Beta and predicted values from the XBETA= and PREDICT= options on the OUTPUT statement are incorrect, as is the output of the CTABLE option that relies on predicted values. To avoid the problem, remove the linear dependencies indicated by the NOTE.

  SAS Note V6-LOGISTIC-G043 documents this problem.

- In PROC LOGISTIC, if the OFFSET= and OUTROC= options are both specified, the values in the OUTROC= data set will be incorrect. The offset variable is incorrectly omitted from the computation of the predicted probabilities, which are then used to compute the variables in the OUTROC= data set. The data set contents are correct if the OFFSET= option is not used.

  SAS Note V6-LOGISTIC-C111 documents this problem.

- In PROC TRANSREG, when the CLASS variable is crossed with a transformation with a variable list, all variables are not always created. For example, in:

```
PROC TRANSREG DESIGN;
  MODEL CLASS(X / ZERO=NONE) | IDE(Z1-Z5);
```

  the variables Z4 and Z5 are not being crossed with the CLASS variables. To circumvent this problem, cross all variables as shown below:

```
PROC TRANSREG DESIGN;
   MODEL CLASS (X/ZERO=NONE)
         CLASS (X/ZERO=NONE) * IDE(Z1)
         CLASS (X/ZERO=NONE) * IDE(Z2)
         CLASS (X/ZERO=NONE) * IDE(Z3)
         CLASS (X/ZERO=NONE) * IDE(Z4)
         CLASS (X/ZERO=NONE) * IDE(Z5)
   OUTPUT;
   RUN;
```

  SAS Note V6-TRANSREG-B934 documents this problem.

- The compiler used in PROC GENMOD and PROC CALIS will compute incorrect analytic derivatives when a SUM statement is used in the model specification. The SUM statement is often used in a summation DO loop. For example, the following code would return an incorrect derivative for y:

```
y=0;
do i=1 to 3;
   y + x + a;
end;
```

  To circumvent the problem, replace the SUM statement with an assignment statement to define the summation. The above example would be modified as:

```
y=0;
do i=1 to 3;
   y=y + x + a;
end;
```

  SAS Note V6-SYS.PROC-D515 documents this problem.

- In the PROC GENMOD, if one or more variables from the input data set are used in any of the DEVIANCE, VARIANCE, FWDLINK, or INVLINK statements, then the results of the OBSTATS option may be incorrect.

  SAS Note V6-GENMOD-B887 documents this problem.

- In PROC GENMOD, when the DIST=BINOMIAL option is used in conjunction with the FREQ statement, the DF and VALUE/DF columns in the Criteria for Assessing Goodness of Fit table are incorrect. The values of the FREQ variable are incorrectly ignored when computing degrees of freedom in binomial models. Correct values can be obtained by replicating each observation as many times as its FREQ value and running PROC GENMOD without the FREQ statement.

  SAS Note V6-GENMOD-C144 documents this problem.

- Incorrect results can be reported from the CONTRAST or ESTIMATE statements when multiple RANDOM statements are used. At least one of the RANDOM statements must use the GROUP= option for this problem to occur.

  The only workaround for this problem is to recode your RANDOM statements without using the GROUP= option.

  SAS Note V6-MIXED-C520 documents this problem.

- If you are using the TIMELIST= option to specify times greater than the greatest time in the DATA= data set, and if the greatest time in the DATA= data set is a censored observation, PROC LIFETEST will print the survival estimates that correspond to the last event time, rather than the missing values.

  Because the survival estimates are not defined for time values greater than the last observed time when the last observed time is censored, the estimates should appear as missing values in the output.

  SAS Note V6-LIFETEST-B511 documents this problem.

- Under rare circumstances, an incorrect P-value can be returned from the EXACT statement. This behavior will only occur in 2x2 cases where the following is true:

  Let the 4 cell counts for the 2x2 table be denoted as c(1,1), c(1,2), c(2,1), and c(2,2). Define the column totals as c1=c(1,1)+c(2,1) and c2=c(1,2)+c(2,2). Define the row totals r1 and r2 accordingly. Define the average ranks for the columns as u1=(c1+1)/2 and u2=c1 + (c2+1)/2. Let N be the total sample size. Then, calculate the following terms:

  ```
  z = c(2,2) - 2*r2*c2/N
  c = z - (r2+c2-N)
  y = c(2,1)*u1 + c(2,2)*u2 - (N+1)*r2/2
  ```

  The bug will occur only if y>0, c=0, and z>=0. There is no circumvention for this problem.

  SAS Note V6-NPAR1WAY-C257 documents this problem.

- The critical values for the REGWQ multiple comparison test in PROC ANOVA or PROC GLM should be monotone non-decreasing in the number of means. Occasionally, they are not if the error degrees of freedom is relatively small. A reference that discusses this issue is **Multiple Comparisons: Theory and Methods** by Jason Hsu (1996). The publisher is Chapman & Hall.

  SAS Note V6-SYS.PROC-C295 documents this problem.

- The REGWF multiple comparison test was incorrectly implemented in the MEANS statement. The underlying problem was that our implementation incorrectly assumed that only contiguous subsets for the groups ordered by sample means needed to be tested for equality, as is the case with REGWQ. In general, for REGWF, all subsets of means must be tested for equality.

  SAS Note V6-SYS.PROC-C294 documents this problem.

- Incorrect predicted values from the `P` or `PM` options or incorrect values for the dependent variable can be reported in the `PREDICTED` table when multiple `RANDOM` statements are used with non-nested `SUBJECT=` effects.

  There is no circumvention for this problem.

  SAS Note V6-MIXED-C557 documents this problem.

- The standard errors of fixed effects in a `GLM` model (a model with no `RANDOM` or `REPEATED` statements) will be incorrect when the `NOPROFILE` option is used. There is no circumvention for this problem.

  SAS Note V6-MIXED-C780 documents this problem.

- The critical values for the DUNCAN's test should be monotone non-decreasing in the number of means. Occasionally they are not, if the `CLASS` variable has many levels. This is a machine-dependent problem, but no machine should have a problem with fewer than 30 means.

  To circumvent the problem, try other multiple comparison tests, such as `TUKEY` or `LSD`.

  SAS Note V6-GLM-C842 documents this problem.

- Prior to Release 6.12 of the SAS System, if one specifies more than `SS2`, `SS3`, `SS4` on the `MODEL` statement in `PROC GLM`, the degrees of freedom (`DF`) for all of them are the same and equal to the `DF` for the highest `SS` computed.

  For example, if:

  ```
  MODEL Y=A B A*B/SS1 SS2 SS3 SS4
  ```

  `PROC GLM` will report the `SS4 DF` for the `SS2` and `SS3 DF`.

  This is a problem when the true `DF` for the different `SS` options are not equal to the `DF` for the highest `SS` computed. To see if one has encountered this problem, it will be necessary to run a separate `GLM` for each type of `SS` requested on the `MODEL` statement and compare the `DF` to those reported when more than one `SS` is specified on the `MODEL` statement.

  To circumvent the problem, specify a separate `GLM` for each type of `SS` requested.

  SAS Note V6-GLM-C889 documents this problem.

- Derivatives of the `_WEIGHT_` variable (including the differences used in the `DUD` method) are not calculated with respect to the parameters. Thus, if your `_WEIGHT_` variable is a function of the parameters, there is no contribution to the gradient and/or the Hessian of the objective function (`SSE`). This is the desired effect if you are performing an iteratively reweighted least squares analysis. However, if you are performing an estimation using a `LOSS` function, this may not be the desired effect.

  SAS Note V6-NLIN-D106 documents this problem.

SAS/TOOLKIT Software

- FORTRAN is not a language that SAS/TOOLKIT software supports in Release 6.11 of the SAS System for Digital UNIX, ABI+ for Intel Architecture, and MIPS ABI.