Chapter 3
The DTREE Procedure

Chapter Table of Contents

Chapter 3
The DTREE Procedure

Overview

The DTREE procedure in SAS/OR software is an interactive procedure for decision analysis. The procedure interprets a decision problem represented in SAS data sets, finds the optimal decisions, and plots on a line printer or a graphics device the decision tree showing the optimal decisions.

To use PROC DTREE you first construct a decision model to represent your problem. This model, called *a generic decision tree model*, is made up of *stages*[*]. Every stage has a *stage name*, which identifies the stage, a *type*, which specifies the type of the stage, and the possible *outcomes* of the stage. There are three types of stages: decision stages, chance stages, and end stages.

A *decision stage* represents a particular decision you have to make. The outcomes of a decision stage are the possible *alternatives* (or *actions*) of the decision. A *chance stage* represents an *uncertain factor* in the decision problem (a statistician may call it a *random variable*; here it is called an *uncertainty*). The outcomes of a chance stage are *events*, one of which will occur according to a given probability distribution. An *end stage* terminates a particular *scenario* (a sequence of alternatives and events). It is not necessary to include an end stage in your model; the DTREE procedure adds an end stage to your model if one is needed.

Each outcome of a decision or chance stage also has several attributes, an *outcome name* to identify the outcome, a *reward* to give the instant reward of the outcome, and a *successor* to specify the name of the stage that comes next when this outcome is realized. For chance stages, a *probability* attribute is also needed. It gives the relative likelihood of this outcome. Every decision stage should have at least two alternatives, and every chance stage should have at least two events. Probabilities of events for a chance stage *must* sum to 1. End stages do not have any outcomes.

The structure of a decision model is given in the STAGEIN= data set. It contains the stage name, the type, and the attributes (except probability) of all outcomes for each stage in your model. You can specify each stage in one observation or across several observations. If a diagrammatic representation of a decision problem is all you want, you probably do not need any other data sets.

If you want to evaluate and analyze your decision problem, you need another SAS data set, called the PROBIN= data set. This data set describes the probabilities or conditional probabilities for every event in your model. Each observation in the data set contains a list of given conditions (list of outcomes), if there are any, and at least one combination of event and probability. Each event and probability combination

[*]The stages are often referred to as *variables* in many decision analysis articles.

identifies the probability that the event occurs given that all the outcomes specified in the list occur. If no conditions are given, then the probabilities are unconditional probabilities.

The third data set, called the PAYOFFS= data set, contains the value of each possible scenario. You can specify one or more scenarios and the associated values in one observation. If the PAYOFFS= data set is omitted, the DTREE procedure assumes that all values are zeros and uses rewards for outcomes to evaluate the decision problem.

You can use PROC DTREE to display, evaluate, and analyze your decision problem. In the PROC DTREE statement, you specify input data sets and other options. A VARIABLES statement identifies the variables in the input data set that describe the model. This statement can be used only once and must appear immediately after the PROC DTREE statement. The EVALUATE statement evaluates the decision tree. You can display the optimal decisions by using the SUMMARY statement, or you can plot the complete tree with the TREEPLOT statement. Finally, you can also associate HTML pages with decision tree nodes and create web-enabled decision tree diagrams.

It is also possible to modify interactively some attributes of your model with the MODIFY statement and to change the order of decisions by using the MOVE statement. Before making any changes to the model, you should save the current model with the SAVE statement so that you can call it back later by using the RECALL statement. Questions about the value of perfect information or the value of perfect control are answered using the VPI and VPC statements. Moreover, any options that can be specified in the PROC DTREE statement can be reset at any time with the RESET statement.

All statements can appear in any order and can be used as many times as desired with one exception. The RECALL statement must be preceded by at least one SAVE statement. In addition, only one model can be saved at any time; the SAVE statement overwrites the previously saved model. Finally, you can use the QUIT statement to stop processing and exit the procedure.

The DTREE procedure produces one output data set. The IMAGEMAP= data set contains the outline coordinates for the nodes in the decision tree that can be used to generate HTML MAP tags.

PROC DTREE uses the Output Delivery System (ODS), a SAS subsystem that provides capabilities for displaying and controlling the output from SAS procedures. ODS enables you to convert any of the output from PROC DTREE into a SAS data set. For further details, refer to the chapter on ODS in the *SAS/STAT User's Guide*.

Getting Started

Introductory Example

A decision problem for an oil wildcatter illustrates the use of the DTREE procedure. The oil wildcatter must decide whether or not to drill at a given site before his option expires. He is uncertain about many things: the cost of drilling, the extent of the oil or gas deposits at the site, and so on. Based on the reports of his technical staff, the hole could be '**`Dry`**' with probability 0.5, '**`Wet`**' with probability 0.3, and '**`Soaking`**' with probability 0.2. His monetary payoffs are given in the following table.

Table 3.1. Monetary Payoffs of Oil Wildcatter's Problem

	Drill	Not Drill
Dry	0	0
Wet	$700,000	0
Soaking	$1,200,000	0

The wildcatter also learned from the reports that the cost of drilling could be $150,000 with probability 0.2, $300,000 with probability 0.6, and $500,000 with probability 0.2. He can gain further relevant information about the underlying geological structure of this site by conducting seismic soundings. A cost control procedure that can make the probabilities of the '**`High`**' cost outcomes smaller (and hence, the probabilities of the '**`Low`**' cost outcomes larger) is also available. However, such information and control are quite costly, about $60,000 and $120,000, respectively. The wildcatter must also decide whether or not to take the sounding test or the cost control program before he makes his final decision: to drill or not to drill.

The oil wildcatter feels that he should structure and analyze his basic problem first: whether or not to drill. He builds a model that contains one decision stage named '**`Drill`**' (with two outcomes, '**`Drill`**' and '**`Not_Drill`**') and two chance stages named '**`Cost`**' and '**`Oil_Deposit`**'. A representation of the model is saved in three SAS data sets. In particular, the STAGEIN= data set can be saved as follows:

```
   /* -- create the STAGEIN= data set            -- */
data Dtoils1;
   input _STNAME_ $12. _STTYPE_  $4. _OUTCOM_ $12.
         _SUCCES_ $12.;
   datalines;
Drill       D   Drill       Cost
.           .   Not_Drill   .
Cost        C   Low         Oil_Deposit
.           .   Fair        Oil_Deposit
.           .   High        Oil_Deposit
Oil_Deposit C   Dry         .
.           .   Wet         .
.           .   Soaking     .
;
```

The structure of the decision problem is given in the Dtoils1 data set. As you apply this data set, you should be aware of the following points:

- There is no reward variable in this data set; it is not necessary.
- The ordering of the chance stages '**`Cost`**' and '**`Oil_Deposit`**' is arbitrary.

- Missing values for the _SUCCES_ variable are treated as '**_ENDST_**' (the default name of the end stage) unless the associated outcome variable (_OUTCOM_) is also missing.

The following PROBIN= data set contains the probabilities of events:

```
   /* -- create the PROBIN= data set               -- */
data Dtoilp1;
   input _EVENT1 $12. _PROB1
         _EVENT2 $12. _PROB2
         _EVENT3 $12. _PROB3;
   datalines;
Low         0.2     Fair        0.6     High        0.2
Dry         0.5     Wet         0.3     Soaking     0.2
;
```

Notice that the sum of the probabilities of the events '**Low**', '**Fair**', and '**High**' is 1.0. Similarly, the sum of the probabilities of the events '**Dry**', '**Wet**', and '**Soaking**' is 1.0.

Finally, the following statements produce the PAYOFFS= data set that lists all possible scenarios and their associated payoffs.

```
   /* -- create PAYOFFS= data set                     -- */
data Dtoilu1;
   input (_STATE1-_STATE3) ($12.) ;
   format _VALUE_ dollar12.0;

      /* determine the cost for this scenario */
   if _STATE1='Low' then _COST_=150000;
   else if _STATE1='Fair' then _COST_=300000;
   else _COST_=500000;

      /* determine the oil deposit and the          */
      /* corresponding net payoff for this scenario */
   if _STATE2='Dry' then _PAYOFF_=0;
   else if _STATE2='Wet' then _PAYOFF_=700000;
   else _PAYOFF_=1200000;

      /* calculate the net return for this scenario */
   if _STATE3='Not_Drill' then _VALUE_=0;
   else _VALUE_=_PAYOFF_-_COST_;

      /* drop unneeded variables */
   drop _COST_ _PAYOFF_;

   datalines;
Low         Dry         Not_Drill
Low         Dry         Drill
Low         Wet         Not_Drill
Low         Wet         Drill
```

```
Low          Soaking      Not_Drill
Low          Soaking      Drill
Fair         Dry          Not_Drill
Fair         Dry          Drill
Fair         Wet          Not_Drill
Fair         Wet          Drill
Fair         Soaking      Not_Drill
Fair         Soaking      Drill
High         Dry          Not_Drill
High         Dry          Drill
High         Wet          Not_Drill
High         Wet          Drill
High         Soaking      Not_Drill
High         Soaking      Drill
;
```

This data set can be displayed, as shown in Figure 3.1, with the following PROC PRINT statements:

```
   /* -- print the payoff table               -- */
title "Oil Wildcatter's Problem";
title3 "The Payoffs";

proc print data=Dtoilu1;
run;
```

```
                      Oil Wildcatter's Problem

                           The Payoffs

     Obs    _STATE1    _STATE2    _STATE3            _VALUE_

       1     Low       Dry        Not_Drill               $0
       2     Low       Dry        Drill             $-150,000
       3     Low       Wet        Not_Drill               $0
       4     Low       Wet        Drill              $550,000
       5     Low       Soaking    Not_Drill               $0
       6     Low       Soaking    Drill            $1,050,000
       7     Fair      Dry        Not_Drill               $0
       8     Fair      Dry        Drill             $-300,000
       9     Fair      Wet        Not_Drill               $0
      10     Fair      Wet        Drill              $400,000
      11     Fair      Soaking    Not_Drill               $0
      12     Fair      Soaking    Drill              $900,000
      13     High      Dry        Not_Drill               $0
      14     High      Dry        Drill             $-500,000
      15     High      Wet        Not_Drill               $0
      16     High      Wet        Drill              $200,000
      17     High      Soaking    Not_Drill               $0
      18     High      Soaking    Drill              $700,000
```

Figure 3.1. Payoffs of the Oil Wildcatter's Problem

The $550,000 payoff associated with the scenario '**Low**', '**Wet**', and '**Drill**' is a net figure; it represents a return of $700,000 for a wet hole less the $150,000 cost for drilling. Similarly, the net return of the consequence associated with the scenario

'**High**', '**Soaking**', and '**Drill**' is $700,000, which is interpreted as a return of $1,200,000 less the $500,000 '**High**' cost.

Now the wildcatter can invoke PROC DTREE to evaluate his model and to find the optimal decision using the following statements:

```
   /* -- PROC DTREE statements              -- */
title "Oil Wildcatter's Problem";

proc dtree stagein=Dtoils1
           probin=Dtoilp1
           payoffs=Dtoilu1
           nowarning;

   evaluate / summary;
```

The following message, which notes the order of the stages, appears on the SAS log:

```
NOTE: Present order of stages:

      Drill(D), Cost(C), Oil_Deposit(C), _ENDST_(E).
```

```
                      Oil Wildcatter's Problem

                        The DTREE Procedure
                      Optimal Decision Summary

                          Order of Stages

                      Stage          Type
                      -----------------------
                      Drill          Decision
                      Cost           Chance
                      Oil_Deposit    Chance
                      _ENDST_        End


                        Decision Parameters

         Decision Criterion:    Maximize Expected Value (MAXEV)
    Optimal Decision Yields:    $140,000


                      Optimal Decision Policy

                         Up to Stage Drill

             Alternatives    Cumulative      Evaluating
             or Outcomes         Reward           Value
             ------------------------------------------
             Drill                             $140,000*
             Not_Drill                               $0
```

Figure 3.2. Optimal Decision Summary of the Oil Wildcatter's Problem

The SUMMARY option in the EVALUATE statement produces the optimal decision summary shown in Figure 3.2.

The summary shows that the best action, in the sense of maximizing the expected payoff, is *to drill*. The expected payoff for this optimal decision is $140,000, as shown on the summary.

Perhaps the best way to view the details of the results is to display the complete decision tree. The following statement draws the decision tree, as shown in Figure 3.3, in line-printer format:

```
   /* plot decision tree diagram in line-printer mode */
OPTIONS LINESIZE=100;
treeplot/ lineprinter;
```

```
                                      Oil Wildcatter’s Problem

                                        The DTREE Procedure
                                         Line-printer Plot

                                                                              Dry
                                                                          -----------------------E
                                                                          | p=0.5  EV=  $-150,000
                                                     Low                  |   Wet
                                                 -----------------------C-|----------------------E
                                                 | p=0.2  EV=   $300,000  | p=0.3  EV=   $550,000
                                                 |                        | Soaking
                                                 |                        -----------------------E
                                                 |                          p=0.2  EV= $1,050,000
                                                 |                            Dry
                                                 |                        -----------------------E
                                                 |                        | p=0.5  EV=  $-300,000
                           Drill                 |  Fair                  |   Wet
                        -======================C-|----------------------C-|----------------------E
                        |        EV=   $140,000  | p=0.6  EV=   $150,000  | p=0.3  EV=   $400,000
                        |                        |                        | Soaking
                        |                        |                        -----------------------E
                        |                        |                          p=0.2  EV=   $900,000
                        |                        |                            Dry
----------------------D-|                        |                        -----------------------E
        EV=   $140,000  |                        |                        | p=0.5  EV=  $-500,000
                        |                        |  High                  |   Wet
                        |                        -----------------------C-|----------------------E
                        |                          p=0.2  EV=   $-50,000  | p=0.3  EV=   $200,000
                        |                                                 | Soaking
                        |                                                 -----------------------E
                        |                                                   p=0.2  EV=   $700,000
                        |Not_Drill
                        -----------------------E
                                 EV=         $0
```

Figure 3.3. The Decision Tree

Attitudes towards Risk

Assume now that the oil wildcatter is constantly risk averse and has an exponential utility function with a *risk tolerance* (RT) of $700,000. The risk tolerance is a measure of the decision maker’s attitude to risk. See the “Evaluation” section beginning on page 294 for descriptions of the utility function and risk tolerance. The new optimal decision based on this utility function can be determined with the following statement:

```
evaluate / criterion=maxce rt=700000 summary;
```

The summary, shown in Figure 3.4, indicates that the venture of investing in the oil well is worth $-13,580 to the wildcatter, and he should not drill the well.

```
                         Oil Wildcatter's Problem

                           The DTREE Procedure
                         Optimal Decision Summary

                             Order of Stages

                         Stage          Type
                         -----------------------
                         Drill          Decision
                         Cost           Chance
                         Oil_Deposit    Chance
                         _ENDST_        End


                           Decision Parameters

      Decision Criterion:    Maximize Certain Equivalent Value (MAXCE)
          Risk Tolerance:    $700,000
 Optimal Decision Yields:    $0


                         Optimal Decision Policy

                            Up to Stage Drill

                Alternatives    Cumulative      Evaluating
                or Outcomes         Reward           Value
                ------------------------------------------
                Drill                            $-13,580
                Not_Drill                              $0*
```

Figure 3.4. Summary of the Oil Wildcatter's Problem with RT = $700,000

Sensitivity Analysis and Value of Perfect Information

The oil wildcatter learned that the optimal decision changed when his attitude toward risk changed. Since risk attitude is difficult to express quantitatively, the oil wildcatter wanted to learn more about the uncertainties in his problem. Before spending any money on information-gathering procedures, he would like to know the benefit of knowing, before the '**`Drill`**' or '**`Not_Drill`**' decision, the amount of oil or the cost of drilling. The simplest approach is to calculate the value of perfect information for each uncertainty. This quantity gives an upper limit on the amount that could be spent profitably on information gathering. The expected value of information for the amount of oil is calculated by the following statement:

```
vpi Oil_Deposit;
```

The result of the previous statement is written to the SAS log as

```
NOTE: The currently optimal decision yields 140000.
NOTE: The new optimal decision yields 295000.
NOTE: The value of perfect information of stage Oil_Deposit
      yields 155000.
```

This means that the wildcatter could spend up to $\$155,000$ to determine the amount of oil in the deposit with certainty before losing money. There are several alternative ways to calculate the expected value of perfect information. For example, the following statement

```
vpi Cost;
```

is equivalent to

```
save;
move Cost before Drill;
evaluate;
recall;
```

The messages, which appear on the SAS log, show that if there is some way that the wildcatter knows what the cost to drill will be before his decision has to be made, it will yield an expected payoff of $\$150,000$. So, the expected value of perfect information about drilling cost is $\$150,000$ - $\$140,000$ = $\$10,000$.

```
NOTE: The current problem has been successfully saved.

NOTE: Present order of stages:

      Cost(C), Drill(D), Oil_Deposit(C), _ENDST_(E).


NOTE: The currently optimal decision yields 150000.

NOTE: The original problem has been successfully recalled.
NOTE: Present order of stages:

      Drill(D), Cost(C), Oil_Deposit(C), _ENDST_(E).
```

Value of Perfect Control

The oil wildcatter may also want to know what the value of perfect control (VPC) is on the cost of drilling. That is, how much is he willing to pay for getting complete control on the drilling cost? This analysis can be performed with the following statement:

```
vpc Cost;
```

The result is written to the SAS log as

```
NOTE: The currently optimal decision yields 140000.
NOTE: The new optimal decision yields 300000.
NOTE: The value of perfect control of stage Cost
      yields 160000.
```

Oil Wildcatter's Problem with Sounding Test

The wildcatter is impressed with the results of calculating the values of perfect information and perfect control. After comparing those values with the costs of the sounding test and the cost-controlling procedure, he prefers to spend $60,000 on sounding test, which has a potential improvement of $155,000. He is informed that the sounding will disclose whether the terrain below has no structure (which is bad), open structure (which is okay), or closed structure (which is really hopeful). The expert also provides him with the following table, which shows the conditional probabilities.

Table 3.2. Conditional Probabilities of Oil Wildcatter's Problem

State	**Seismic Outcomes**		
	No Structure	Open Structure	Closed Structure
Dry	0.6	0.3	0.1
Wet	0.3	0.4	0.3
Soaking	0.1	0.4	0.5

To include this additional information into his basic problem, the wildcatter needs to add two stages to his model: a decision stage to represent the decision whether or not to take the sounding test, and one chance stage to represent the uncertain test result. The new STAGEIN= data set is

```
   /* -- create the STAGEIN= data set              -- */
data Dtoils2;
   input _STNAME_ $12. _STTYPE_ $3.  _OUTCOM_ $16.
         _SUCCES_ $12. _REWARD_ dollar12.0;
   datalines;
Drill       D  Drill           Cost                .
.           .  Not_Drill       .                   .
Cost        C  Low             Oil_Deposit         .
.           .  Fair            Oil_Deposit         .
.           .  High            Oil_Deposit         .
Oil_Deposit C  Dry             .                   .
.           .  Wet             .                   .
.           .  Soaking         .                   .
Sounding    D  Noseismic       Drill               .
.           .  Seismic         Structure     -$60,000
Structure   C  No_Struct       Drill               .
.           .  Open_Struct     Drill               .
.           .  Closed_Struct   Drill               .
;
```

Note that the cost for the seismic soundings is represented as negative reward (of the outcome Seismic) in this data set. The conditional probabilities for stage Structure are added to the PROBIN= data set as follows:

```
   /* -- create PROBIN= data set                       -- */
data Dtoilp2;
   input _GIVEN_ $8. _EVENT1 $10. _PROB1
                     _EVENT2 $12. _PROB2
                     _EVENT3 $14. _PROB3;
   datalines;
.       Low       0.2 Fair        0.6 High          0.2
.       Dry       0.5 Wet         0.3 Soaking       0.2
Dry     No_Struct 0.6 Open_Struct 0.3 Closed_Struct 0.1
Wet     No_Struct 0.3 Open_Struct 0.4 Closed_Struct 0.3
Soaking No_Struct 0.1 Open_Struct 0.4 Closed_Struct 0.5
;
```

It is not necessary to make any change to the PAYOFFS= data set. To evaluate his new model, the wildcatter invokes PROC DTREE as follows:

```
   /* -- PROC DTREE statements                         -- */
title "Oil Wildcatter's Problem";

proc dtree stagein=Dtoils2
           probin=Dtoilp2
           payoffs=Dtoilu1
           nowarning;

   evaluate;
```

As before, the following messages are written to the SAS log:

```
NOTE: Present order of stages:

      Sounding(D), Structure(C), Drill(D), Cost(C),
      Oil_Deposit(C), _ENDST_(E).

NOTE: The currently optimal decision yields 140000.
```

The following SUMMARY statements produce optimal decision summary as shown in Figure 3.5 and Figure 3.6:

```
   summary / target=Sounding;
   summary / target=Drill;
```

The optimal strategy for the oil-drilling problem is found to be the following:

- No soundings test should be taken, and always drill. This alternative has an expected payoff of $140,000.
- If the soundings test is conducted, then drill unless the test shows the terrain below has no structure.
- The soundings test is worth $180,100 - $140,000 = $40,100 (this quantity is also called the *value of imperfect information* or the *value of sample information*), but it costs $60,000; therefore, it should not be taken.

```
                    Oil Wildcatter's Problem

                      The DTREE Procedure
                    Optimal Decision Summary

                        Order of Stages

                    Stage          Type
                    -----------------------
                    Sounding       Decision
                    Structure      Chance
                    Drill          Decision
                    Cost           Chance
                    Oil_Deposit    Chance
                    _ENDST_        End


                      Decision Parameters

       Decision Criterion:    Maximize Expected Value (MAXEV)
  Optimal Decision Yields:    $140,000


                    Optimal Decision Policy

                     Up to Stage Sounding

          Alternatives     Cumulative      Evaluating
           or Outcomes         Reward           Value
          -------------------------------------------
          Noseismic                 0       $140,000*
          Seismic              -60000       $180,100
```

Figure 3.5. Summary of the Oil Wildcatter's Problem for SOUNDING

```
                    Oil Wildcatter's Problem

                      The DTREE Procedure
                    Optimal Decision Summary

                        Order of Stages

                    Stage          Type
                    -----------------------
                    Sounding       Decision
                    Structure      Chance
                    Drill          Decision
                    Cost           Chance
                    Oil_Deposit    Chance
                    _ENDST_        End


                      Decision Parameters

       Decision Criterion:    Maximize Expected Value (MAXEV)
    Optimal Decision Yields:    $140,000


                    Optimal Decision Policy

                       Up to Stage Drill

                                                 Cumulative     Evaluating
           Alternatives or Outcomes                  Reward          Value
-----------------------------------------------------------------------------
Noseismic                         Drill                   0       $140,000*
Noseismic                         Not_Drill               0             $0
Seismic         No_Struct         Drill              -60000       $-97,805
Seismic         No_Struct         Not_Drill          -60000            $0*
Seismic         Open_Struct       Drill              -60000       $204,286*
Seismic         Open_Struct       Not_Drill          -60000             $0
Seismic         Closed_Struct     Drill              -60000       $452,500*
Seismic         Closed_Struct     Not_Drill          -60000             $0
```

Figure 3.6. Summary of the Oil Wildcatter's Problem for DRILL

Note that the value of sample information also can be obtained by using the following statements:

```
modify Seismic reward 0;
evaluate;
```

The following messages, which appear in the SAS log, show the expected payoff with soundings test is $\$180,100$. Recall that the expected value without test information is $\$140,000$. Again, following the previous calculation, the value of test information is $\$180,100$ - $\$140,000 = \$40,100$.

```
NOTE: The reward of outcome Seismic has been changed to 0.

NOTE: The currently optimal decision yields 180100.
```

Now, the wildcatter has the information to make his best decision.

Syntax

The following statements are available in PROC DTREE:

PROC DTREE *options* **;**
EVALUATE */ options* **;**
MODIFY *specifications* **;**
MOVE *specifications* **;**
QUIT ;
RECALL ;
RESET *options* **;**
SAVE ;
SUMMARY */ options* **;**
TREEPLOT */ options* **;**
VARIABLES */ options* **;**
VPC *specifications* **;**
VPI *specifications* **;**

The DTREE procedure begins with the PROC DTREE statement and terminates with the QUIT statement. The VARIABLES statement can be used only once, and if it is used, it must appear before any other statements. The EVALUATE, MODIFY, MOVE, RECALL, RESET, SAVE, SUMMARY, TREEPLOT, VPC, and VPI statements can be listed in any order and can be used as many times as desired with one exception: the RECALL statement must be preceded by at least one SAVE statement.

You can also submit any other valid SAS statements, for example, OPTIONS, TITLE, and SAS/GRAPH global statements. In particular, the SAS/GRAPH statements that can be used to enhance the DTREE procedure's output on graphics devices are listed in Table 3.3. Refer to *SAS/GRAPH Software: Reference* for more explanation of these statements.

Table 3.3. Statements to Enhance Graphics Output

Statement	Function
FOOTNOTE	Produce footnotes that are displayed on the graphics output
GOPTIONS	Define default values for graphics options
NOTE	Produce text that is displayed on the graphics output
SYMBOL	Create symbol definitions
TITLE	Produce titles that are displayed on the graphics output

Functional Summary

The following tables outline the options available for the DTREE procedure classified by function.

Table 3.4. Accuracy Control Options

Description	Statement	Option
accuracy of numerical computation	DTREE and RESET	TOLERANCE=

Table 3.5. Data Set Specifications

Description	Statement	Option
Annotate data set	DTREE, RESET, and TREEPLOT	ANNOTATE=
Imagemap output data set	DTREE, RESET, and TREEPLOT	IMAGEMAP=
Payoffs data set	DTREE	PAYOFFS=
Probability data set	DTREE	PROBIN=
Stage data set	DTREE	STAGEIN=

Table 3.6. Error Handling Options

Description	Statement	Option
automatically rescale the probabilities of an uncertainty if they are not sum to 1	DTREE and RESET	AUTOSCALE
react to errors being detected	DTREE and RESET	ERRHANDLE=
do not automatically rescale probabilities	DTREE and RESET	NOSCALE
do not display warning message	DTREE and RESET	NOWARNING
display warning message	DTREE and RESET	WARNING

Table 3.7. Evaluation Control Options

Description	Statement	Option
criterion to determine the optimal decision	DTREE, EVALUATE, and RESET	CRITERION=
risk tolerance	DTREE, EVALUATE, and RESET	RT=

Table 3.8. Format Control Options

Description	Statement	Option
maximum decimal width to format numerical values	DTREE, EVALUATE, RESET, SUMMARY, and TREEPLOT	MAXPREC=
maximum field width to format numerical values	DTREE, EVALUATE, RESET, SUMMARY, and TREEPLOT	MAXWIDTH=
maximum field width to format names	DTREE, EVALUATE, RESET, SUMMARY, and TREEPLOT	NWIDTH=

Table 3.9. Graphics Catalog Options

Description	Statement	Option
description field for catalog entry	DTREE, RESET, and TREEPLOT	DESCRIPTION=
name of graphics catalog	DTREE, RESET, and TREEPLOT	GOUT=
name field for catalog entry	DTREE, RESET, and TREEPLOT	NAME=

Table 3.10. Line-printer Options

Description	Statement	Option
characters for line-printer plot	DTREE, RESET, and TREEPLOT	FORMCHAR=

Table 3.11. Link Appearance Options

Description	Statement	Option
color of LOD*	DTREE, RESET, and TREEPLOT	CBEST=
color of all links except LOD*	DTREE, RESET, and TREEPLOT	CLINK=
symbol definition for all links except LOD* and LCP†	DTREE, RESET, and TREEPLOT	LINKA=
symbol definition for LOD*	DTREE, RESET, and TREEPLOT	LINKB=
symbol definition for LCP†	DTREE, RESET, and TREEPLOT	LINKC=
line type of all links except LOD* and LCP†	DTREE, RESET, and TREEPLOT	LSTYLE=
line type of LOD*	DTREE, RESET, and TREEPLOT	LSTYLEB=
line type of LCP†	DTREE, RESET, and TREEPLOT	LSTYLEC=
line thickness of all links except LOD*	DTREE, RESET, and TREEPLOT	LWIDTH=
line thickness of LOD*	DTREE, RESET, and TREEPLOT	LWIDTHB=

Table 3.12. Node Appearance Options

Description	Statement	Option
color of chance nodes	DTREE, RESET, and TREEPLOT	CSYMBOLC=
color of decision nodes	DTREE, RESET, and TREEPLOT	CSYMBOLD=
color of end nodes	DTREE, RESET, and TREEPLOT	CSYMBOLE=
height of symbols for all nodes	DTREE, RESET, and TREEPLOT	HSYMBOL=
symbol definition for chance nodes	DTREE, RESET, and TREEPLOT	SYMBOLC=
symbol definition for decision nodes	DTREE, RESET, and TREEPLOT	SYMBOLD=
symbol definition for end nodes	DTREE, RESET, and TREEPLOT	SYMBOLE=
symbol to draw chance nodes	DTREE, RESET, and TREEPLOT	VSYMBOLC=
symbol to draw decision nodes	DTREE, RESET, and TREEPLOT	VSYMBOLD=
symbol to draw end nodes	DTREE, RESET, and TREEPLOT	VSYMBOLE=

Table 3.13. Output Control Options

Description	Statement	Option
suppress displaying the optimal decision summary	DTREE, EVALUATE, and RESET	NOSUMMARY
display the optimal decision summary	DTREE, EVALUATE, and RESET	SUMMARY
decision stage up to which the optimal decision summary is displayed	DTREE, EVALUATE, RESET, and SUMMARY	TARGET=

*LOD denotes links that indicate optimal decisions.
†LCP denotes links that continue on subsequent pages.

Table 3.14. Plot Control Options

Description	Statement	Option
draw diagram on one page in graphics mode	DTREE, RESET, and TREEPLOT	COMPRESS
information are displayed on the decision tree diagram	DTREE, RESET, and TREEPLOT	DISPLAY=
processing of the Annotate data set	DTREE, RESET, and TREEPLOT	DOANNOTATE
invoke graphics version	DTREE, RESET, and TREEPLOT	GRAPHICS
display labels	DTREE, RESET, and TREEPLOT	LABEL
display legend	DTREE, RESET, and TREEPLOT	LEGEND
invoke line-printer version	DTREE, RESET, and TREEPLOT	LINEPRINTER
suppress processing of the Annotate data set	DTREE, RESET, and TREEPLOT	NOANNOTATE
draw diagram across multiple pages	DTREE, RESET, and TREEPLOT	NOCOMPRESS
suppress displaying label	DTREE, RESET, and TREEPLOT	NOLABEL
suppress displaying legend	DTREE, RESET, and TREEPLOT	NOLEGEND
use rectangular corners for turns in the links	DTREE, RESET, and TREEPLOT	NORC
use rounded corners for turns in the links	DTREE, RESET, and TREEPLOT	RC
vertical space between two end nodes	DTREE, RESET, and TREEPLOT	YBETWEEN=

Table 3.15. Text Appearance Options

Description	Statement	Option
text color	DTREE, RESET, and TREEPLOT	CTEXT=
text font	DTREE, RESET, and TREEPLOT	FTEXT=
text height	DTREE, RESET, and TREEPLOT	HTEXT=

Table 3.16. Variables in PAYOFFS= Data Set

Description	Statement	Option
action outcome names	VARIABLES	ACTION=
state outcome names	VARIABLES	STATE=
payoffs	VARIABLES	VALUE=

Table 3.17. Variables in PROBIN= Data Set

Description	Statement	Option
event outcome names	VARIABLES	EVENT=
given outcome names	VARIABLES	GIVEN=
(conditional) probabilities	VARIABLES	PROB=

Table 3.18. Variables in STAGEIN= Data Set

Description	Statement	Option
outcome names	VARIABLES	OUTCOME=
rewards	VARIABLES	REWARD=
stage name	VARIABLES	STAGE=
successor names	VARIABLES	SUCCESSOR=
type of stage	VARIABLES	TYPE=
web reference variable	VARIABLES	WEB=

PROC DTREE Statement

PROC DTREE *options* **;**

The options that can appear in the PROC DTREE statement are listed in the following section. The options specified in the PROC DTREE statement remain in effect for all statements until the end of processing or until they are changed by a RESET statement. These options are classified under appropriate headings: first, all options that are valid for all modes of the procedure are listed followed by the options classified according to the mode (line-printer or graphics) of invocation of the procedure.

General Options

AUTOSCALE | NOSCALE

specifies whether the procedure should rescale the probabilities of events for a given chance stage if the total probability of this stage is not equal to 1. The default is NOSCALE.

CRITERION=*i*

indicates the decision criterion to be used for determining the optimal decision and the certain equivalent for replacing uncertainties. The following table shows all valid values of *i* and their corresponding decision criteria and certain equivalents.

Table 3.19. Values for the CRITERION= Option

i	**Criterion**	**Certain Equivalent**
MAXEV	maximize	expected value
MINEV	minimize	expected value
MAXMLV	maximize	value with largest probability
MINMLV	minimize	value with largest probability
MAXCE	maximize	certain equivalent value of expected utility
MINCE	minimize	certain equivalent value of expected utility

The default value is MAXEV. The last two criteria are used when your utility curve can be fit by an exponential function. See the “Evaluation” section beginning on page 294 for more information on the exponential utility function.

DISPLAY=*(information-list)*

specifies information that should be displayed on each link of the decision tree diagram. Table 3.20 lists the valid keywords and corresponding information.

Table 3.20. Information on Decision Tree and Keywords

Keyword	Information
ALL	all information listed in this table
CR	cumulative rewards of outcomes on the path that leads to the successor of the link
EV	evaluating value that can be expected from the successor of the link
LINK	outcome name represented by the link
P	probability of the outcome represented by the link
R	instant reward of the outcome represented by the link
STAGE	stage name of the successor of the link

The default value is (LINK P EV R CR).

Note that the probability information displays on links that represent chance outcomes only. In addition, the PROBIN= option must be specified. The expected values display only if the decision tree has been evaluated. The reward information displays on a link only if the instant reward of the outcome represented by the link is nonzero. The cumulative rewards do not display if the cumulative rewards of links are all zeros.

ERRHANDLE=DRAIN | QUIT

specifies whether the procedure should stop processing the current statement and wait for next statement or quit PROC DTREE when an error has been detected by the procedure. The default value is DRAIN.

GRAPHICS

creates plots for a graphics device. To specify this option, you need to have SAS/GRAPH software licensed at your site. This is the default.

LABEL | NOLABEL

specifies whether the labels for information displayed on the decision tree diagram should be displayed. If the NOLABEL option is not specified, the procedure uses the following symbols to label all the information that is displayed on each link

Table 3.21. Labels and Their Corresponding Information

Label	Information
cr=	the cumulative rewards of outcomes on the path that lead to the successor of the link
EV=	the value that can be expected from the successor of the link
p=	the probability of the outcome represented by the link
r=	the instant reward of the outcome

The default is LABEL.

LINEPRINTER
LP

creates plots of line-printer quality. If you do not specify this option, graphics plots are produced.

MAXPREC=*d*
specifies the maximum decimal width (the precision) in which to format numerical values using $w.d$ format. This option is used in displaying the decision tree diagrams and the summaries. The value for this option must be no greater than 9; the default value is 3.

MAXWIDTH=*mw*
specifies the maximum field width in which to format numerical values (probabilities, rewards, cumulative rewards and evaluating values) using $w.d$ format. This option is used in displaying the decision tree diagrams and the summaries. The value for this option must be no greater than 16 and must be at least 5 plus the value of the MAXPREC= option. The default value is 10.

NWIDTH=*nw*
specifies the maximum field width in which to format outcome names when displaying the decision tree diagrams. The value for this option must be no greater than 40; the default value is 32.

PAYOFFS=*SAS-data-set*
names the SAS data set that contains the evaluating values (payoffs, losses, utilities, and so on) for each state and action combination. The use of PAYOFFS= is optional in the PROC DTREE statement. If the PAYOFFS= option is not used, PROC DTREE assumes that all evaluating values at the end nodes of the decision tree are 0.

PROBIN=*SAS-data-set*
names the SAS data set that contains the (conditional) probability specifications of outcomes. The PROBIN= SAS data set is required if the evaluation of the decision tree is desired.

RT=*r*
specifies the value of the risk tolerance. The RT= option is used only when CRITERION=MAXCE or CRITERION=MINCE is specified. If the RT= option is not specified, and CRITERION=MAXCE or CRITERION=MINCE is specified, PROC DTREE changes the value of the CRITERION= option to MAXEV or MINEV (which would mean straight-line utility function and imply infinite risk tolerance).

STAGEIN=*SAS-data-set*
names the SAS data set that contains the stage names, stage types, names of outcomes, and their rewards and successors for each stage. If the STAGEIN= option is not specified, PROC DTREE uses the most recently created SAS data set.

SUMMARY | NOSUMMARY
specifies whether an optimal decision summary should be displayed each time the decision tree is evaluated. The decision summary lists all paths through the tree that lead to the target stage as well as the cumulative rewards and the evaluating values of all alternatives for that path. The alternative with optimal evaluating value for each path is marked with an asterisk (*). The default is NOSUMMARY.

TARGET=*stage*

specifies the decision stage up to which the optimal decision policy table is displayed. The TARGET= option is used only in conjunction with the SUMMARY option. The stage specified must be a decision stage. If the TARGET= option is not specified, the procedure displays an optimal decision policy table for each decision stage.

TOLERANCE=*d*

specifies either a positive number close to 0 or greater than 1. PROC DTREE treats all numbers within e of 0 as 0, where

$$e = \begin{cases} d & \text{if } d < 1 \\ d \times \epsilon & \text{otherwise} \end{cases}$$

and ϵ is the *machine epsilon*. The default value is $1,000$.

WARNING | NOWARNING

specifies whether the procedure should display a warning message when

- the payoff for an outcome is not assigned in the PAYOFFS= data set
- probabilities of events for a given chance stage have been automatically scaled by PROC DTREE because the total probability of the chance stage does not equal 1

The default is WARNING.

YBETWEEN=*ybetween <units>*

specifies the vertical distance between two successive end nodes. If the GRAPHICS option is specified, the valid values for the optional *units* are listed in Table 3.22.

Table 3.22. Valid Value for the Units of the YBETWEEN= Option

Unit	Description
CELL	character cells
CM	centimeters
INCH	inches
PCT	percentage of the graphics output area
SPACE	height of the box surrounding the node, its predecessor link, and all text information

The value of the YBETWEEN= option must be greater than or equal to 0. Note that if the COMPRESS option is specified, the actual distance between two successive end nodes is scaled by PROC DTREE and may not be the same as the YBETWEEN= specification.

If the LINEPRINTER option is specified, the optional *units* value can be CELL or SPACE. The value of the YBETWEEN= option must be a nonnegative integer.

If you do not specify units, a unit specification is determined in the following order:

- the GUNIT= option in a GOPTIONS statement, if the GRAPHICS option is specified
- the default unit, CELL

The default value of YBETWEEN= option is 0.

Graphics Options

The following options are specifically for the purpose of producing a high-resolution quality decision tree diagram.

ANNOTATE=*SAS-data-set*
ANNO=*SAS-data-set*

specifies an input data set that contains appropriate Annotate variables. The ANNOTATE= option enables you to add features (for example, customized legend) to plots produced on graphics devices. For additional information, refer to the chapter on the annotate data set in *SAS/GRAPH Software: Reference*.

CBEST=*color*
CB=*color*

specifies the color for all links in the decision tree diagram that represent optimal decisions. If you do not specify the CBEST= option, the color specification is determined in the following order:

- the CI= option in the *j*th generated SYMBOL definition, if the option LINKB=*j* is specified
- the second color in the colors list

CLINK=*color*
CL=*color*

specifies the color for all links in the decision tree diagram except those that represent optimal decisions. If the CLINK= option is not specified, the color specification is determined in the following order:

- the CI= option in the *i*th generated SYMBOL definition, if the option LINKA=*i* is specified
- the third color in the colors list

COMPRESS | NOCOMPRESS
CP | NOCP

specifies whether the decision tree diagram should be drawn on one physical page. If the COMPRESS option is specified, PROC DTREE determines the scale so that the diagram is compressed, if necessary, to fit on one physical page. Otherwise, the procedure draws the diagram across multiple pages if necessary. The default is NOCOMPRESS.

CSYMBOLC=*color*
CC=*color*
specifies the color of the symbol used to draw all chance nodes in the decision tree diagram. If the CSYMBOLC= option is not specified, the color specification is determined in the following order:

- the CV= option in the *m*th generated SYMBOL definition, if the option SYMBOLC=*m* is specified
- the CSYMBOL= option in a GOPTIONS statement
- the fifth color in the colors list

CSYMBOLD=*color*
CD=*color*
specifies the color of the symbol used to draw all decision nodes in the decision tree diagram. If the CSYMBOLD= option is not specified, the color specification is determined in the following order:

- the CV= option in the *d*th generated SYMBOL definition, if the option SYMBOLD=*d* is specified
- the CSYMBOL= option in a GOPTIONS statement
- the fourth color in the colors list

CSYMBOLE=*color*
CE=*color*
specifies the color of the symbol used to draw all end nodes in the decision tree diagram. If the CSYMBOLE= option is not specified, the color specification is determined in the following order:

- the CV= option in the *n*th generated SYMBOL definition, if the option SYMBOLE=*n* is specified
- the CSYMBOL= option in a GOPTIONS statement
- the sixth color in the colors list

CTEXT=*color*
CT=*color*
specifies the color to be used for all text that appears on plots except on TITLE and FOOTNOTE lines. If the CTEXT= option is not specified, the color specification is determined in the following order:

- the CTEXT= option in a GOPTIONS statement
- the first color in the colors list

DESCRIPTION=*'string'*
DES=*'string'*
specifies a descriptive string, up to 40 characters long, that appears in the description field of the master menu of PROC GREPLAY. If the DESCRIPTION= option is omitted, the description field contains a description assigned by PROC DTREE.

DOANNOTATE | NOANNOTATE
DOANNO | NOANNO
specifies whether the Annotate data set should be processed. If the NOANNOTATE option is specified, the procedure does not process the Annotate data set even though the ANNOTATE= option is specified. The default is DOANNOTATE.

FTEXT=*name*
FONT=*name*
specifies the font to be used for text on plots. If you do not use this option, the font specification is determined in the following order:

- the FTEXT= option in a GOPTIONS statement
- the hardware font for your graphics output device

Refer to the chapter on SAS/GRAPH fonts in *SAS/GRAPH Software: Reference* for details about SAS/GRAPH fonts.

GOUT=*SAS-catalog*
specifies the name of the graphics catalog used to save the output produced by PROC DTREE for later replay. For additional information, refer to the chapter on graphics output in *SAS/GRAPH Software: Reference*.

HSYMBOL=*h*
HS=*h*
specifies that the height of symbols for all nodes in the decision tree diagram is *h* times the heights of symbols assigned by SAS/GRAPH software. You can specify the heights of decision nodes, chance nodes, and end nodes by using the HEIGHT= options in the corresponding SYMBOL statements. For example, if you specify the options HSYMBOL=*2* and SYMBOLD=*1* in the PROC DTREE statement and defined SYMBOL*1* as

```
symbol1 height=4 pct;
```

then all decision nodes in the decision tree diagram are sized at $2 \times 4 = 8\%$ of the graphics output area. The default value is 1.

HTEXT=*h*
HT=*h*
specifies that the height for all text in plots (except that in TITLE and FOOTNOTE statements) be *h* times the height of the characters assigned by SAS/GRAPH software. You can also specify character height by using the HTEXT= option in a GOPTIONS statement.

For example, if you specify the option HTEXT=0.6 in the PROC DTREE statement and also specified a GOPTIONS statement as follows

```
goptions htext=2 in;
```

then the size of all text is $0.6 \times 2 = 1.2$ inches. For more explanation of the GOPTIONS statement, refer to the chapter on the GOPTIONS statement in *SAS/GRAPH Software: Reference*. The default value is 1.

IMAGEMAP=*SAS-data-set*

names the SAS data set that receives a description of the areas of a graph and a link for each area. This information is for the construction of HTML imagemaps. You use a SAS DATA step to process the output file and generate your own HTML files. The graph areas correspond to the link information that comes from the WEB= variable in the STAGEIN= data set. This gives you complete control over the appearance and structure of your HTML pages.

LEGEND | NOLEGEND
LG | NOLG

specifies whether the default legend should be displayed. If the NOLEGEND is not specified, the procedure displays a legend at the end of each page of the decision tree diagram. The default is LEGEND.

LINKA=*i*

if the LINKA=*i* option is specified, then PROC DTREE uses the color specified with the CI= option, the type specified with the LINE= option, and the thickness specified with the WIDTH= option in the *i*th generated SYMBOL definition to draw all links in the decision tree diagram, except those that indicate optimal decisions and those that are continued on subsequent pages. There is no default value for this option. The color, type, and thickness specifications may be overridden by the specifications of the CLINK=, LSTYLE=, and LWIDTH= options in the PROC DTREE statement.

Note that if you specify the LINKA=*i* option, PROC DTREE uses the specifications in the *i*th *generated* SYMBOL *definition* and not the specifications in the SYMBOL*i* statement. Refer to *SAS/GRAPH Software: Reference* for the details about creating, canceling, reviewing, and altering SYMBOL definitions.

LINKB=*j*

if the LINKB=*j* option is specified, then PROC DTREE uses the color specified with the CI= option, the type specified with the LINE= option, and the thickness specified with the WIDTH= option in the *j*th generated SYMBOL definition to draw all links that represent optimal decisions. There is no default value for this option. The color, type, and thickness specifications may be overridden by the specifications of the CBEST=, LSTYLEB=, and LWIDTHB= options in the PROC DTREE statement.

Note that if you specify the LINKB=*j* option, PROC DTREE uses the specifications in the *j*th *generated* SYMBOL *definition* and not the specifications in the SYMBOL*j* statement. Refer to *SAS/GRAPH Software: Reference* for the details about creating, canceling, reviewing, and altering SYMBOL definitions.

LINKC=*k*

if the LINKC=*k* option is specified, then PROC DTREE uses the type specified with the LINE= option in the *k*th generated SYMBOL definition to draw all links in the decision tree diagram that are continued on subsequent pages. There is no default value for this option. The color and thickness for links continued on another page indicate whether the link represents an optimal decision or not. The type specification may be overridden by the specification of the LSTYLEC= option in the PROC DTREE statement.

Note that if you specify the LINKC=*k* option, PROC DTREE uses the specifications in the *k*th *generated* SYMBOL *definition* and not the specifications in the SYMBOL*k* statement. Refer to *SAS/GRAPH Software: Reference* for the details about creating, canceling, reviewing, and altering SYMBOL definitions.

LSTYLE=*l*
L=*l*

specifies the line type (style) used for drawing all links in the decision tree diagram, except those that represent the optimal decisions and those that are continued on subsequent pages. Valid values for *l* are 1 though 46. If the LSTYLE= option is not specified, the type specification is determined in the following order:

q

- the LINE= option in the *i*th generated SYMBOL definition, if the option LINKA=*i* is specified
- the default value, 1 (solid line)

LSTYLEB=*l2*
LB=*l2*

specifies the line type (style) used for drawing the links in the decision tree diagram that represent optimal decisions. Valid values for *l2* are 1 though 46. If the LSTYLEB= option is not specified, the type specification is determined in the following order:

- the LINE= option in the *j*th generated SYMBOL definition, if the option LINKB=*j* is specified
- the default value, 1 (solid line)

LSTYLEC=*l3*
LC=*l3*

specifies the line type (style) used for drawing the links in the decision tree diagram that are continued on the next subsequent pages. Valid values for *l3* are 1 though 46. If the LSTYLEC= option is not specified, the type specification is determined in the following order:

- the LINE= option in the *k*th generated SYMBOL definition, if the option LINKC=*k* is specified

- the default value, 2 (dot line)

LWIDTH=*w*
LTHICK=*w*

specifies the line thickness (width) used to draw all links in the decision tree diagram except those that represent the optimal decisions.

If the LWIDTH= option is not specified, the thickness specification is determined in the following order:

- the WIDTH= option in the *i*th generated SYMBOL definition, if the option LINKA=*i* is specified
- the default value, 1

LWIDTHB=*w2*
LTHICKB=*w2*

specifies the line thickness (width) used to draw the links in the decision tree diagram that represent optimal decisions. If the LWIDTHB= option is not specified, the thickness specification is determined in the following order:

- the WIDTH= option in the *j*th generated SYMBOL definition, if the option LINKB=*j* is specified
- 2 times the thickness for links that represent regular outcomes

NAME=*'string'*

specifies a descriptive string, up to 8 characters long, that appears in the name field of the master menu of PROC GREPLAY. The default is 'DTREE '.

RC | NORC

specifies whether the links in the decision tree diagram should be drawn with rounded corners or with rectangular corners. The default is RC.

SYMBOLC=*m*
SYMBC=*m*

if the SYMBOLC= option is specified, then PROC DTREE uses the color specified with the CV= option, the character specified with the VALUE= option, the font specified with the FONT= option, and the height specified with the HEIGHT= option in the *m*th generated SYMBOL definition to draw all chance nodes in the decision tree diagram. There is no default value for this option. The color and the symbol specifications may be overridden by the specification of the CSYMBOLC= and VSYMBOLC= options in the PROC DTREE statement. The height of the symbol can be changed by the HSYMBOL= option in the PROC DTREE statement.

Note that if you specify the SYMBOLC=*m* option, PROC DTREE uses the specifications in the *m*th generated SYMBOL definition and not the specifications in the SYMBOL*m* statement. Refer to *SAS/GRAPH Software: Reference* for the details about creating, canceling, reviewing, and altering SYMBOL definitions.

SYMBOLD=*d*
SYMBD=*d*

if the SYMBOLD= option is specified, then PROC DTREE uses the color specified with the CV= option, the character specified with the VALUE= option, the font specified with the FONT= option, and the height specified with the HEIGHT= option in the *d*th generated SYMBOL definition to draw all decision nodes in the decision tree diagram. There is no default value for this option. The color and the symbol specifications may be overridden by the specification of the CSYMBOLD= and VSYMBOLD= options in the PROC DTREE statement. The height of the characters can be changed by the HSYMBOL= option in the PROC DTREE statement.

Note that if you specify the SYMBOLD=*d* option, PROC DTREE uses the specifications in the *d*th generated SYMBOL definition and not the specifications in the SYMBOL*d* statement. Refer to *SAS/GRAPH Software: Reference* for the details about creating, canceling, reviewing, and altering SYMBOL definitions.

SYMBOLE=*n*
SYMBE=*n*

if the SYMBOLE= option is specified, then PROC DTREE uses the color specified with the CV= option, the character specified with the VALUE= option, the font specified with the FONT= option, and the height specified with the HEIGHT= option in the *n*th generated SYMBOL definition to draw all end nodes in the decision tree diagram. There is no default value for this option. The color and the symbol specifications may be overridden by the specification of the CSYMBOLE= and VSYMBOLE= options specified in the PROC DTREE statement. The height of the characters can be changed by the HSYMBOL= option in the PROC DTREE statement.

Note that if you specify the SYMBOLE=*n* option, PROC DTREE uses the specifications in the *n*th generated SYMBOL definition and not the specifications in the SYMBOL*n* statement. Refer to *SAS/GRAPH Software: Reference* for the details about creating, canceling, reviewing, and altering SYMBOL definitions.

VSYMBOLC=*symbolc-name*
VC=*symbolc-name*

specifies that the symbol *symbolc-name* from the special symbol table be used to draw all chance nodes in the decision tree diagram. If you do not specify this option, the symbol used is determined in the following order:

- the options VALUE= and FONT= specifications in the *m*th generated SYMBOL definition, if the option SYMBOLC=*m* is specified
- the symbol CIRCLE in the special symbol table

VSYMBOLD=*symbold-name*
VD=*symbold-name*

specifies that the symbol *symbold-name* from the special symbol table be used to draw all decision nodes in the decision tree diagram. If you do not specify this option, the symbol used is determined in the following order:

- the options VALUE= and FONT= specifications in the *d*th generated SYMBOL definition, if the option SYMBOLD=*d* is specified
- the symbol SQUARE in the special symbol table

VSYMBOLE=*symbole-name*
VE=*symbole-name*

specifies that the symbol *symbole-name* from the special symbol table be used to draw all end nodes in the decision tree diagram. If you do not specify this option, the symbol used is determined in the following order:

- the options VALUE= and FONT= specifications in the *n*th generated SYMBOL definition, if the option SYMBOLE=*n* is specified
- the symbol DOT in the special symbol table

Line-Printer Options

The following options are specifically for the purpose of producing line-printer quality decision tree diagram.

FORMCHAR<*(syni-list)*>**=** *'formchar-string'*

defines characters to be used for features on line-printer plots. The *syni-list* is a list of numbers ranging from 1 to 13. The list identifies which features are controlled with the string characters. The *formchar-string* gives characters for features in *syni-list*. Any character or hexadecimal string can be used. By default, *syni-list* is omitted, and the FORMCHAR= option gives a string for all 13 features. The features associated with values of *syni* are as follows

Table 3.23. Features Associated with the FORMCHAR= Option

Syni	Description of Character	Feature
1	vertical bar	vertical link
2	horizontal bar	horizontal link
3	box character (upper left)	vertical up to horizontal turn
5	box character (upper right)	horizontal and down vertical joint
8	box character (middle right)	horizontal to split joint
9	box character (lower left)	vertical down to horizontal turn
11	box character (lower right)	horizontal and up vertical joint
13	horizontal thick	horizontal link that represents optimal decision

Note that characters 4, 6, 7, 10, and 12 are not used in drawing a decision tree diagram.

As an example, the decision tree diagram in Figure 3.7 is produced by the following statement:

```
title "Decision Tree Showing the Effects of FORMCHAR";

data Dtoils4;
   input _STNAME_ $12. _STTYPE_  $4. _OUTCOM_ $12.
         _SUCCES_ $12.;
   datalines;
Drill        D   Drill       Cost
.            .   Not_Drill   .
Cost         C   Low         Oil_Deposit
.            .   High        Oil_Deposit
Oil_Deposit C    Dry         .
.            .   Wet         .
;

proc dtree stagein=Dtoils4
           nowarning
           ;

   treeplot / formchar(1 2 3 5 8 9 11 13)='|-/*<\+='
              lineprinter display=(LINK);
quit;
```

```
              Decision Tree Showing the Effects of FORMCHAR

                          The DTREE Procedure
                           Line-printer Plot

                                      Low          Dry
                                  /----------C-*----------E
                       Drill      |            |   Wet
                    /----------C-<             \----------E
                    |             |  High          Dry
          ----------D-<            \----------C-*----------E
                    |                          |   Wet
                    |                          \----------E
                    |Not_Drill
                    \----------E
```

Figure 3.7. Decision Tree Showing the Effects of FORMCHAR

By default, the form character list specified with the SAS system option FORMCHAR= is used; otherwise, the default is ‘`|----|+|---+=`’. Refer to the chapter on the Calendar Procedure in the *SAS Procedures Guide* for more information.

EVALUATE Statement

EVALUATE */ options* **;**

The EVALUATE statement causes PROC DTREE to evaluate the decision tree and calculate the optimal decisions. If the SUMMARY option is specified a decision summary is displayed. Otherwise, the current optimal value is displayed on the SAS log.

The following options, which can appear in the PROC DTREE statement, can also be specified in the EVALUATE statement:

CRITERION=*i*	MAXPREC=*d*	MAXWIDTH=*mw*
NOSUMMARY	NWIDTH=*nw*	RT=*r*
SUMMARY	TARGET=*stage*	

The MAXPREC=, MAXWIDTH=, and NWIDTH=, options are valid only in conjunction with the SUMMARY option. The RT= option is valid only in conjunction with the CRITERION=MAXCE or CRITERION=MINCE specification. The options specified in this statement are only in effect for this statement.

MODIFY Statement

MODIFY *outcome-name* **REWARD** *new-value* **;**
MODIFY *stage-name* **TYPE ;**

The MODIFY statement is used to change either the type of a stage or the reward from an outcome. If MODIFY *outcome-name* REWARD *new-value* is given where the *outcome-name* is an outcome specified in the STAGEIN= data set, and *new-value* is a numeric value, then the reward of the outcome named *outcome-name* is changed to *new-value*.

If MODIFY *stage-name* TYPE is given where *stage-name* is a stage name specified in the STAGEIN= data set, then the type of the stage named *stage-name* is changed to '**DECISION**' if its current type is '**CHANCE**' and is changed to '**CHANCE**' if its current type is '**DECISION**'. You cannot change the type of an '**END**' stage. The change of the type of a stage from '**CHANCE**' to '**DECISION**' can help the decision-maker learn how much improvement can be expected if he or she could pick which of the future (or unknown) outcomes would occur. However, if you want to change the type of a stage from '**DECISION**' to '**CHANCE**', the procedure is not able to determine the probabilities for its outcomes unless you specify them in the PROBIN= data set.

MOVE Statement

MOVE *stage1* **(BEFORE | AFTER)** *stage2* **;**

The MOVE statement is used to change the order of the stages. After all data in input data sets have been read, PROC DTREE determines the order (from left to right) of all stages specified in the STAGEIN= data set and display the order in the SAS log. The ordering is determined based on the rule that if stage **A** is the successor of an outcome of stage **B**, then stage **A** should occur to the right of stage **B**. The MOVE statement can be used to change the order. If the keyword BEFORE is used, *stage1* becomes the new successor for all immediate predecessors of *stage2*, and *stage2* becomes the new successor for all outcomes of *stage1*. An outcome is said to be an *immediate predecessor* of a stage if the stage is the successor of that outcome. Similarly, if the keyword AFTER is used, the old leftmost (in previous order) successor of outcomes for *stage2* becomes the new successor for all outcomes of *stage1* and the new successor of all outcomes of *stage2* is *stage1*.

There are two limitations: the END stage cannot be moved, and no stage can be moved after the END stage. In practice, any stage after the END stage is useless.

QUIT Statement

QUIT ;

The QUIT statement tells the DTREE procedure to terminate processing. This statement has no options.

RECALL Statement

RECALL ;

This statement tells PROC DTREE to recall the decision model that was saved previously with a SAVE statement. The RECALL statement has no options.

RESET Statement

RESET *options* **;**

The RESET statement is used to change options after the procedure has started. All of the options that can be set in the PROC DTREE statement can also be reset with this statement, except for the STAGEIN=, the PROBIN=, and the PAYOFFS= data set options.

SAVE Statement

SAVE ;

The SAVE statement saves the current model (attributes of stages and outcomes, the ordering of stages, and so on) to a scratch space from which you can call it back later. It is a good idea to save your decision model before you specify any MOVE or MODIFY statements. Then you can get back to your original model easily after a series of statements that change the decision model. The SAVE statement has no options.

SUMMARY Statement

SUMMARY */ options* **;**

Unlike the SUMMARY option on the PROC DTREE statement or the EVALUATE statement, which specifies that PROC DTREE display a decision summary when the decision tree is evaluated, the SUMMARY statement causes the procedure to display the summary immediately. If the decision tree has not been evaluated yet, or if it has been changed (by the MOVE, MODIFY, or RECALL statement) since last evaluated, the procedure evaluates or re-evaluates the decision tree before the summary is displayed.

The following options that can appear in the PROC DTREE statement can also be specified in this statement:

MAXPREC=*d*	MAXWIDTH=*mw*
NWIDTH=*nw*	TARGET=*stage*

The options specified in this statement are in effect only for this statement.

TREEPLOT Statement

TREEPLOT / *options* **;**

The TREEPLOT statement plots the current decision tree (a diagram of the decision problem). Each path in the decision tree represents a possible scenario of the problem. In addition to the nodes and links on the decision tree, the information for each link that can be displayed on the diagram is listed in Table 3.24.

Table 3.24. Information on Decision Tree Diagram

Information	**Labeled by**
stage name for the successor of the link	NL[‡]
outcome name for the link	NL[‡]
probability of the outcome	p=
value can be expected from the successor	EV=
instant reward of the outcome	r=
cumulative rewards of outcomes on the path that leads to the successor	cr=

If necessary, the outcome names and the stage names are displayed above the link, and other information (if there is any) is displayed below the link. The DISPLAY= option can be used to control which information should be included in the diagram. The NOLABEL can be used to suppress the displaying of the labels.

If the LINEPRINTER option is used, the decision nodes, chance nodes, and the end nodes are represented by the characters '**D**', '**C**', and '**E**', respectively. The links are displayed using the specifications of the FORMCHAR= option. See the section "PROC DTREE Statement" beginning on page 268 for more details. In graphics mode, the control of the appearances of nodes and links is more complex. Please see the "Displaying the Decision Tree" section beginning on page 298 for more information.

The following options that can appear in the PROC DTREE statement can also be specified in the TREEPLOT statement:

DISPLAY=*(information-list)*	GRAPHICS	LABEL
LINEPRINTER	MAXPREC=*d*	MAXWIDTH=*mw*
NOLABEL	NWIDTH=*nw*	YBETWEEN=*ybetween <units>*

The following line-printer options that can appear in the PROC DTREE statement can also be specified in the TREEPLOT statement if the LINEPRINTER option is specified:

FORMCHAR<*(syni-list)*>=*'formchar-string'*

[‡] 'NL' denotes this information is not labeled.

Moreover, the following graphics options that can appear in the PROC DTREE statement can also be specified in the TREEPLOT statement if the GRAPHICS option is specified:

ANNOTATE=*SAS-data-set*	CBEST=*color*	CLINK=*color*
COMPRESS	CSYMBOLC=*color*	CSYMBOLD=*color*
CSYMBOLE=*color*	CTEXT=*color*	DESCRIPTION=*'string'*
DOANNOTATE	FTEXT=*name*	GOUT=*SAS-catalog*
HSYMBOL=*h*	HTEXT=*h*	IMAGEMAP=*SAS-data-set*
LEGEND	LINKA=*i*	LINKB=*j*
LINKC=*k*	LSTYLE=*l*	LSTYLEB=*l2*
LSTYLEC=*l3*	LWIDTH=*w2*	LWIDTHB=*w2*
NAME=*'string'*	NOANNOTATE	NOCOMPRESS
NOLEGEND	NORC	RC
SYMBOLC=*m*	SYMBOLD=*d*	SYMBOLE=*n*
VSYMBOLC=*symbolc-name*	VSYMBOLD=*symbolc-name*	VSYMBOLE=*symbolc-name*

The options specified in this statement are in effect only for this statement, and they may override the options specified in the PROC DTREE statement.

VARIABLES Statement

VARIABLES */ options* **;**

The VARIABLES statement specifies the variable lists in the input data sets. This statement is optional but if it is used, it must appear immediately after the PROC DTREE statement. The options that can appear in the VARIABLES statement are divided into groups according to the data set in which they occur. Table 3.25 lists all the variables or variable lists associated with each input data set and their types. It also lists the default variables if they are not specified in this statement.

Table 3.25. Input Data Sets and Their Associated Variables

Data Set	Variable	Type[§]	Interpretation	Default
STAGEIN=	OUTCOME=	C/N	Outcome names	Variables with prefix _OUT
	REWARD=	N	Instant reward	Variables with prefix _REW
	STAGE=	C/N	Stage name	_STNAME_
	SUCCESSOR=	as STAGE=	Immediate successors	Variables with prefix _SUCC
	TYPE=	C/N	Stage type	_STTYPE_
	WEB=	C	HTML page for the stage	
PROBIN=	EVENT=	as OUTCOME=	Event names	Variables with prefix _EVEN
	GIVEN=	as OUTCOME=	Names of given outcomes	Variables with prefix _GIVE
	PROB=	N	Conditional probabilities	Variables with prefix _PROB
PAYOFFS=	ACTION=	as OUTCOME=	Action names of final decision	Variables with prefix _ACT
	STATE=	as OUTCOME=	Outcome names	Variables with prefix _STAT
	VALUE=	N	Values of the scenario	Variables with prefix _VALU

Variables in STAGEIN= Data Set

The following options specify the variables or variable lists in the STAGEIN= input data set that identify the stage name, its type, its outcomes, and the reward; and the immediate successor of each outcome for each stage in the decision model:

OUTCOME=(*variables*)
identifies all variables in the STAGEIN= data set that contain the outcome names of the stage specified by the STAGE= variable. If the OUTCOME= option is not specified, PROC DTREE looks for the default variable names that have the prefix _OUT in the data set. It is necessary to have at least one OUTCOME= variable in the STAGEIN= data set. The OUTCOME= variables can be either all character or all numeric. You cannot mix character and numeric variables as outcomes.

REWARD=(*variables*)
COST=(*variables*)
identifies all variables in the STAGEIN= data set that contain the reward for each outcome specified by the OUTCOME= variables. If the REWARD= option is not specified, PROC DTREE looks for the default variable names that have the prefix _REW in the data set. The number of REWARD= variables must be equal to the number of OUTCOME= variables in the data set. The REWARD= variables must have numeric values.

STAGE=*variable*
specifies the variable in the STAGEIN= data set that names the stages in the decision model. If the STAGE= option is omitted, PROC DTREE looks for the default variable named _STNAME_ in the data set. The STAGE= variable must be specified if the data set does not contain a variable named _STNAME_. The STAGE= variable can be either character or numeric.

SUCCESSOR=(*variables*)
SUCC=(*variables*)
identifies all variables in the STAGEIN= data set that contain the names of immediate successors (another stage) of each outcome specified by the OUTCOME= variables. These variables must be of the same type and length as those defined in the STAGE= option. If the SUCCESSOR= option is not specified, PROC DTREE looks for the default variable names that have the prefix _SUCC in the data set. The number of SUCCESSOR= variables must be equal to the number of OUTCOME= variables. The values of SUCCESSOR= variables must be stage names (values of STAGE= variables in the same data set).

TYPE=*variable*
identifies the variable in the STAGEIN= data set that contains the type identifier of the stage specified by the STAGE= variable. If the TYPE= option is omitted, PROC DTREE looks for the default variable named _STTYPE_ in the data set. The TYPE= variable must be specified if the data set does not contain a variable named _STTYPE_. The STAGE= variable can be either character or numeric.

§ 'C' denotes character, 'N' denotes numeric, 'C/N' denotes character or numeric, and 'as X' denotes the same as variable X.

The following are valid values for the TYPE= variable

Value					Description
DECISION	or	D	or	1	identifies the stage as a decision stage
CHANCE	or	C	or	2	identifies the stage as an uncertain stage
END	or	E	or	3	identifies the stage as an end stage

It is not necessary to specify an end stage in the STAGEIN= data set.

WEB=*variable*
HTML=*variable*

specifies the character variable in the STAGEIN= data set that identifies an HTML page for each stage. The procedure generates an HTML image map using this information for all the decision tree nodes corresponding to a stage.

Variables in PROBIN= Data Set

The following options specify the variables or variable lists in the PROBIN= input data set that identify the given outcome names, the event (outcome) name, and the conditional probability for each outcome of a chance stage.

EVENT=(*variables*)

identifies all variables in the PROBIN= data set that contain the names of events (outcomes) that probabilities depend on the outcomes specified by the GIVEN= variables. If the EVENT= option is not specified, PROC DTREE looks for the default variable names that have the prefix _EVEN in the data set. You must have at least one EVENT= variable in the PROB= data set. The values of EVENT= variables must be outcome names that are specified in the STAGEIN= data set.

GIVEN=(*variables*)

identifies all variables in the PROBIN= data set that contain the given condition (a list of outcome names) of a chance stage on which the probabilities of the outcome depend. If the GIVEN= option is not specified, PROC DTREE looks for the default variable names that have the prefix _GIVE in the data set. It is not necessary to have GIVEN= variables in the data set but if there are any, their values must be outcome names that are specified in the STAGEIN= data set.

PROB=(*variables*)

identifies all variables in the PROBIN= data set that contain the values of the conditional probability of each event specified by the EVENT= variables, given that the outcomes specified by the GIVEN= variables have occurred. If the PROB= option is not specified, PROC DTREE looks for the default variable names that have the prefix _PROB in the data set. The number of PROB= variables in the data set must be equal to the number of EVENT= variables. The PROB= variables must have numeric values between 0 and 1 inclusive.

Variables in PAYOFFS= Data Set

The following options specify the variables or variable lists in the PAYOFFS= input data set that identify the possible scenarios (a sequence of outcomes), the final outcome names, and the evaluating values (payoff) of combinations of scenarios and final outcomes.

ACTION=(*variables*)
identifies all variables in the PAYOFFS= data set that contain the name of the final outcome for each possible scenario. If the ACTION= option is not specified, PROC DTREE looks for the default variable names that have the prefix _ACT in the data set. It is not necessary to have any ACTION= variables in the PAYOFFS= data set, but if there are any, their values must be outcome names specified in the STAGEIN= data set.

STATE=(*variables*)
identifies all variables in the PAYOFFS= data set that contain the names of outcomes that identify a possible scenario (a sequence of outcomes or a path in the decision tree), or the names of outcomes which combine with every outcome specified by the ACTION= variables to identify a possible scenario. If the STATE= option is not specified, PROC DTREE looks for the default variable names that have the prefix _STAT in the data set. It is not necessary to have any STATE= variables in the PAYOFFS= data set, but if there are any, their values must be outcome names specified in the STAGEIN= data set.

VALUE=(*variables*)
PAYOFFS=(*variables*)
UTILITY=(*variables*)
LOSS=(*variables*)
identifies all variables in the PAYOFFS= data set that contain the evaluating values or payoffs for all possible scenarios identified by the outcomes specified by the STATE= variables and the outcomes specified by the associated ACTION= variables. If the VALUE= option is not specified, PROC DTREE looks for the default variable names that have the prefix _VALU in the data set. The number of VALUE= variables must be equal to the number of ACTION= variables if there are any ACTION= variables. If there are no ACTION= variables in the data set, at least one STATE= variable must be in the data set, and the number of VALUE= variables must be exactly 1. The VALUE= variables must have numeric values.

VPC Statement

VPC *chance-stage-name* **;**

The VPC statement causes PROC TREE to compute the value of perfect control (the value of controlling an uncertainty). The effect of perfect control is that you can pick the outcome of an uncertain stage. This value gives an upper limit on the amount you should be willing to spend on any control procedure. Only the name of a chance stage can be used to calculate the value of perfect control. The procedure evaluates the decision tree, if it has not already done so, before computing this value.

VPI Statement

VPI *chance-stage-name* **;**

The VPI statement causes PROC DTREE to compute the value of perfect information. The value of perfect information is the benefit of resolving an uncertain stage before making a decision. This value is the upper limit on the improvement that can be expected for any information gathering effort. Only the name of a chance stage can be used to calculate the value of perfect information. The procedure evaluates the decision tree, if it has not already done so, before computing this value.

Details

Input Data Sets

A decision problem is normally constructed in three steps:

1. A structuring of the problem in terms of decisions, uncertainties, and consequences.
2. Assessment of probabilities for the events.
3. Assessment of values (payoffs, losses, or preferences) for each consequence or scenario.

PROC DTREE represents these three steps in three SAS data sets. The STAGEIN= data set describes the structure of the problem. In this data set, you define all decisions and define all key uncertainties. This data set also contains the relative order of when decisions are made and uncertainties are resolved (planning horizon). The PROBIN= data set assigns probabilities for the uncertain events, and the PAYOFFS= data set contains the values (or utility measure) for each consequence or scenario. Please see the “Overview” section (beginning on page 251) and the “Getting Started” section (beginning on page 252) for a description of these three data sets.

PROC DTREE is designed to minimize the rules for describing a problem. For example, the PROBIN= data set is required only when the evaluation and analysis of a decision problem is necessary. Similarly, if the PAYOFFS= data set is not specified, the DTREE procedure assumes all payoff values are 0. The order of the observations is not important in any of the input data sets. Since a decision problem can be structured in many different ways and the data format is so flexible, all possible ways of describing a given decision problem cannot be shown here. However, some alternate ways of supplying the same problem are demonstrated. For example, the following statements show another way to input the oil wildcatter’s problem described in the “Introductory Example” section beginning on page 252.

```
data Dtoils3;
   input _STNAME_ $12. _STTYPE_  $4. _OUTCOM_ $12.
         _REWARD_ dollar12.0  _SUCCES_ $12.;
   datalines;
Drill       D   Drill               .    Cost
.           .   Not_drill           .    .
Cost        C   Low          -$150,000 Oil_deposit
.           .   Fair         -$300,000 Oil_deposit
.           .   High         -$500,000 Oil_deposit
Oil_deposit C   Dry                 .    .
.           .   Wet           $700,000 .
.           .   Soaking     $1,200,000 .
;

data Dtoilp3;
   input _EVENT1 $12. _PROB1 8.1 _EVENT2 $12. _PROB2 8.1;
   datalines;
Low         0.2     Dry         0.5
Fair        0.6     Wet         0.3
High        0.2     Soaking     0.2
;

title "Oil Wildcatter's Problem";
proc dtree stagein=Dtoils3 probin=Dtoilp3
           nowarning;

   evaluate / summary;
```

Note that the STAGEIN= data set describes the problem structure and the payoffs (via the REWARD= variable). Thus, the PAYOFFS= data set is no longer needed. Note also the changes made to the PROBIN= data set. The results, shown in Figure 3.8, are the same as those shown in Figure 3.2 on page 256. However, the rewards and the payoffs are entirely different entities in decision tree models. Recall that the reward of an outcome means the *instant returns* when the *outcome* is realized. On the other hand, the payoffs are the *return* from each *scenario*. In the other words, the decision tree model described in the previous code and the model described in the "Introductory Example" section beginning on page 252 are not equivalent, even though they have the same optimal decision.

```
                    Oil Wildcatter's Problem

                      The DTREE Procedure
                    Optimal Decision Summary

                        Order of Stages

                    Stage          Type
                    -----------------------
                    Drill          Decision
                    Cost           Chance
                    Oil_deposit    Chance
                    _ENDST_        End


                      Decision Parameters

       Decision Criterion:    Maximize Expected Value (MAXEV)
  Optimal Decision Yields:    140000


                    Optimal Decision Policy

                      Up to Stage Drill

            Alternatives    Cumulative    Evaluating
            or Outcomes         Reward         Value
            ----------------------------------------
            Drill                    0       140000*
            Not_drill                0            0
```

Figure 3.8. Optimal Decision Summary of the Oil Wildcatter's Problem

You can try many alternative ways to specify your decision problem. Then you can choose the model that is most convenient and closest to your real problem. If PROC DTREE cannot interpret the input data, it writes a message to that effect to the SAS log unless the NOWARNING option is specified. However, there are mistakes that PROC DTREE cannot detect. These often occur after the model has been modified with either the MOVE statement or the MODIFY statement. After a MOVE statement is specified, it is a good idea to display the decision tree (using the TREEPLOT statement) and check the probabilities and value assessments to make sure they are reasonable.

For example, using the REWARD= variable in the STAGEIN= data set to input the payoff information as shown in the previous code may cause problems if you change the order of the stages. Suppose you move the stage '**Cost**' to the beginning of the tree, as was done in the "Sensitivity Analysis and Value of Perfect Information" section on page 258:

```
move Cost before Drill;
evaluate / summary;
```

The optimal decision yields $\$140,000$, as shown on the optimal decision summary in Figure 3.9.

```
                         Oil Wildcatter's Problem

                           The DTREE Procedure
                         Optimal Decision Summary

                             Order of Stages

                         Stage          Type
                         -----------------------
                         Cost           Chance
                         Drill          Decision
                         Oil_deposit    Chance
                         _ENDST_        End


                           Decision Parameters

            Decision Criterion:    Maximize Expected Value (MAXEV)
       Optimal Decision Yields:    140000


                         Optimal Decision Policy

                            Up to Stage Drill

                Alternatives         Cumulative    Evaluating
                or Outcomes              Reward         Value
           --------------------------------------------------
           Low          Drill           -150000       450000*
           Low          Not_drill       -150000            0
           Fair         Drill           -300000       450000*
           Fair         Not_drill       -300000            0
           High         Drill           -500000       450000*
           High         Not_drill       -500000            0
```

Figure 3.9. Optimal Decision Summary of the Oil Wildcatter's Problem

Recall that when this was done in the "Sensitivity Analysis and Value of Perfect Information" section (page 258), the optimal decision yielded $\$150,000$. The reason for this discrepancy is that the cost of drilling, implemented as (negative) instant rewards here, is imposed on all scenarios including those that contain the outcome '**`Not_drill`**'. This mistake can be observed easily from the **`Cumulative Reward`** column of the optimal decision summary shown Figure 3.9.

Changing a decision stage to a chance stage is another example where using the MODIFY statement without care may cause problems. PROC DTREE cannot determine the probabilities of outcomes for this new chance stage unless they are included in the PROBIN= data set. In contrast to changing a chance stage to a decision stage (which yields insight on the value of gaining control of an uncertainty), changing a decision stage to a chance stage is not likely to yield any valuable insight even if the needed probability data are included in the PROBIN= data set, and it should be avoided.

Missing Values

In the STAGEIN= data set, missing values are allowed only for the STAGE= and TYPE= variables when the information of a stage is specified in more than one observation. In this case, missing values for the STAGE= and TYPE= variables are not allowed for the first observation defining the stage. Missing values for the OUTCOME=, GIVEN=, EVENT=, STATE=, and ACTION= variables are ignored. Missing values for the REWARD=, PROB=, and VALUE= variables are treated as 0. Missing values for the SUCCESSOR= variables are ignored if the value for the corresponding OUTCOME= variable is also missing.

Interactivity

The DTREE procedure is interactive. You start the procedure with the PROC DTREE statement and terminate it with the QUIT statement. It is not necessary to have a VARIABLES statement, although if you do include one, it must appear immediately after the PROC DTREE statement. The other statements such as the EVALUATE, MODIFY, MOVE, RECALL, RESET, SAVE, SUMMARY, TREEPLOT, VPC, and VPI, as well as the FOOTNOTE, GOPTIONS, NOTE, SYMBOL, and TITLE statements of SAS/GRAPH Software can be used in any order and as often as needed. One exception is that the RECALL statement has to be preceded by at least one SAVE statement.

When an error is detected during processing a statement other than the PROC DTREE statement and the QUIT statement, the procedure terminates if the option ERRHANDLE=QUIT is specified; otherwise it stops processing the current statement and waits for the next statement. In either case, an error message is written to the SAS log. If an error is detected in the PROC DTREE statement or the QUIT statement, the procedure terminates immediately with an error message.

Options on Multiple Statements

Many options that can be specified in the PROC DTREE statement can also appear in other statements. The options specified in the PROC DTREE statement remain in effect for all statements until the end of processing or until they are changed by a RESET statement. In this sense, those options are *global* options. The options specified in other statements are in effect only for the statement in which they are specified; hence, they are *local* options. If an option is specified both in the PROC DTREE statement and in another statement, the local specification overrides the global specification.

For example, the following statements

```
reset criterion=maxev;
evaluate / criterion=maxce rt=700000;
summary;
```

imply that the decision problem is evaluated and the optimal decision is determined based on the criterion MAXCE with RT=700000. However, the optimal decision summary produced by the SUMMARY statement is based on the option CRITERION=MAXEV and not the MAXCE criterion. If you want an option to be set permanently, use the RESET statement.

The Order of Stages

The order of stages is an important issue in structuring the decision problem. This sets the sequence of events or a time horizon and determines when a decision has to be made and when a chance stage has its uncertainty resolved. If a decision stage precedes another decision stage in the stages order, the decision to the right is made after the decision to the left. Moreover, the choice made in the first decision is remembered by the decision maker when he or she makes the second decision. Any chance stages that occur to the left of a decision stage have their uncertainty resolved before the decision is made. In the other words, the decision maker knows what actually happened when he or she makes the decision. However, the order of two chance stages is fairly arbitrary if there are no other decision stages between them. For example, you can change the order of stages '**`Cost`**' and '**`Oil_Deposit`**' in the oil wildcatter's problem without affecting the results.

PROC DTREE determines the order (from left to right) of all stages specified in the STAGEIN= data set. The ordering is based on the rule that if stage **`A`** is the successor of an outcome of stage **`B`**, then stage **`A`** should occur to the right of (or after) stage **`B`**. With the MOVE statement, you can change this order. The MOVE statement is very useful in determining the value (benefit or penalty) of postponing or hurrying a decision. In particular, the *value of perfect information* about an uncertainty can be determined by moving the corresponding chance stage to the beginning. However, as mentioned in early sections, the results may be misleading if you use the MOVE statement without care. See the "Input Data Sets" section beginning on page 289 for an example.

Suggestions for preventing this are as follows:

- Always save, using the SAVE statement, the original structure before making any changes.
- Use the TREEPLOT statement to display the complete decision tree and check all details after you change the order.

Evaluation

The EVALUATE statement causes PROC DTREE to calculate the optimal decision. The evaluate process is done by successive use of two devices:

- Find a certain equivalent for the uncertain evaluating values at each chance node.
- Choose the best alternative at each decision node.

The *certain equivalent* of an uncertainty is the certain amount you would accept in exchange for the uncertain venture. In other words, it is a single number that characterizes an uncertainty described by a probability distribution. This value is subjective and can vary widely from person to person. There are two quantities, closely related to the certain equivalent, that are commonly used by decision-makers: the most likely value and the expected value. The *most likely value* of an uncertainty is the value with the largest probability. The *expected value* is the sum of all outcomes multiplied by their probabilities.

Perhaps, the most popular way to find the certain equivalent for an uncertainty is the use of *utility function* or *utility curve*. *Utility* is a measurement of relative *preference* to the decision maker for particular outcomes. The utility function assigns a utility to payoff when it is in terms of continuous values such as money. The certain equivalent of an uncertainty (a random variable) is calculated by the following steps:

1. Use the utility function or the utility curve to find the utility values of the outcomes.
2. Calculate the expected utility of the uncertainty.
3. Determine the certain equivalent of the uncertainty as the value that corresponding utility value is the expected utility.

Refer to Raiffa (1970) for a complete discussion of the utility function.

A simple case that is commonly used is the straight line utility curve or the linear utility function. The linear utility function has the form

$$u(x) = a + bx$$

where x is the evaluating value, and a and b are parameters set by the choice of two points in the utility curve. For example, if the utility curve passes two points $u(0) = 0$ and $u(1000) = 1$, then parameters a and b are set by $a = 0$ and $b = 1/1000$. The certain equivalent of an uncertainty based on this function is the expected value.

Another special case that is commonly used is the exponential utility function, as

$$u(x) = a - b \times \exp(-x/r)$$

where, again, a and b can be set by the choice of two arbitrary points in the utility curve. For example, if your utility curve goes through points $(0, 0)$ and $(1000, 1)$, then a and b are given by

$$a = b = 1/[1 - \exp(-1000/r)]$$

If an uncertain venture A has n events, event i having probability p_i and payoff x_i, and if the utility function is an exponential function as in the preceding example, then the certain equivalent of A is

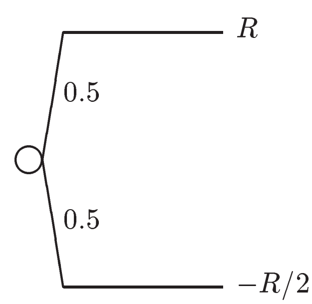
$$\mathrm{CE}(A) = -r \ln \left[\sum_{i=1}^{n} p_i \exp(-x_i/r) \right]$$

and is independent of the choice of values for a and b (provided that $b > 0$) (Raiffa 1970).

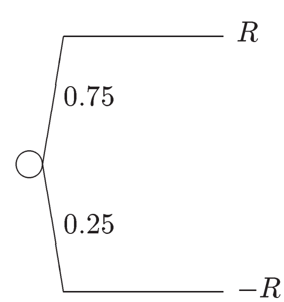
The parameter r, called the *risk tolerance*, describes the curvature of the utility function. Moreover, the quantity $1/r$, called *risk aversion coefficient* (Howard 1968) is a measure of risk aversion.

Experimental results show that within a reasonable range of values, many utility curves can be fit quite well by an exponential function.

If your utility function is an exponential function as in the preceding example, the risk tolerance can be estimated by the largest number R for which the following venture is still acceptable to you.



A similar way to approximate the risk tolerance is to find the largest value R for which the venture is acceptable (Howard 1988).



For corporate decision making, there are some rules of thumb for estimating the risk tolerance. Examples are to set risk tolerance about equal to one of the following:

- net income of the company
- one sixth of equity
- six percent of net sales

To reveal how well these rules perform in assessing corporate risk tolerance, Howard (1988) provided the following two tables: Table 3.26 shows the relationship between the risk tolerance and financial measures of four large oil and chemicals companies. There, the risk tolerances are obtained from the top executives of the companies. The net sales, net income, and equity are obtained from the annual reports of the four companies.

Table 3.26. Relating Corporate Risk Tolerance to Financial Measures

Measure ($ millions)	**Company** A	B	C	D
Net Sales	2,300	3,307	16,000	31,000
Net Income	120	152	700	1,900
Equity	1,000	1,153	6,500	12,000
Risk Tolerance	150	200	1,000	2,000

Table 3.27 shows the ratio of risk tolerance to each of the other quantities.

Table 3.27. Ratios of Corporate Risk Tolerance to Financial Measures

Measure	**Company** A	B	C	D	**Average**
RT/Sales	0.0652	0.0605	0.0625	0.0645	0.0632
RT/Income	1.25	1.32	1.43	1.05	1.26
RT/Equity	0.150	0.174	0.154	0.167	0.161

Once the certain equivalents for all chance nodes are assessed, the choice process at each decision node is fairly simple; select the alternative yielding either the maximum or the minimum (depending on the problem) future certain equivalent value*. You can use the CRITERION= option to control the way the certain equivalent is calculated for each chance node and the optimal alternative is chosen at each decision node. Possible values for the CRITERION= option are listed in Table 3.19 on page 268. If you use an exponential utility function, the RT= option can be used to specify your risk tolerance. You also have control over how to present the solution. By default, PROC DTREE writes the value of the optimal decisions to the SAS log. In addition, with the SUMMARY option, you can ask PROC DTREE to display the optimal decision summary to the output.

*The future certain equivalent value is often referred to as the evaluating value in this documentation.

Displayed Output

The SUMMARY statement and the SUMMARY option in an EVALUATE statement cause PROC DTREE to display a optimal decision summary for the decision model. This output is organized into various tables, and they are discussed in order of appearance.

Order of stages

The "Order of stages" table lists all stages, and their types, in order of appearance in the decision model. See the "The Order of Stages" section on page 294 for details.

For ODS purposes, the label of the "Order of stages" table is "Stages."

Decision Parameters

The "Decision Parameters" table describes the criterion used for determining the optimal decision and the certain equivalent for replacing uncertainties. If you specify the option CRITERION=MAXCE or CRITERION=MINCE in the PROC DTREE statement or in the EVALUATE statement, an additional row is added to the table listing the value of the risk tolerance. It also contains a row showing the value of the optimal decision yields. For additional information, see the "Evaluation" section beginning on page 294.

For ODS purposes, the label of the "Decision Parameters" table is "Parameters."

Optimal Decision Policy

By default, PROC DTREE produces an "Optimal Decision Policy" table for each decision stages. You can use the TARGET= option to force PROC DTREE to produce only one table for a particular stage. The Alternatives or Outcomes columns list the events in the scenario that leads to the current stage. The Cumulative Reward column lists the rewards accumulated along the scenario to the events of the current target stage. The Evaluating Value column lists the values that can be expected form the events of the target stage. An asterisk (*) is placed beside an evaluating value indicates the current event is the best alternative of the given scenario.

For ODS purposes, the label of the "Optimal Decision Policy" table is "Policy."

Displaying the Decision Tree

PROC DTREE draws the decision tree either in line-printer mode or in graphics mode. However, you need to have SAS/GRAPH software licensed at your site to use graphics mode. In many cases, the procedure draws the decision tree across page boundaries. If the decision tree diagram is drawn on multiple pages, the procedure numbers each page of the diagram on the upper right corner of the page. The pages are numbered starting with the upper left corner of the entire diagram. Thus, if the decision tree diagram is broken into three horizontal and four vertical levels and you want to paste all the pieces together to form one picture, they should be arranged as shown in Figure 3.10.

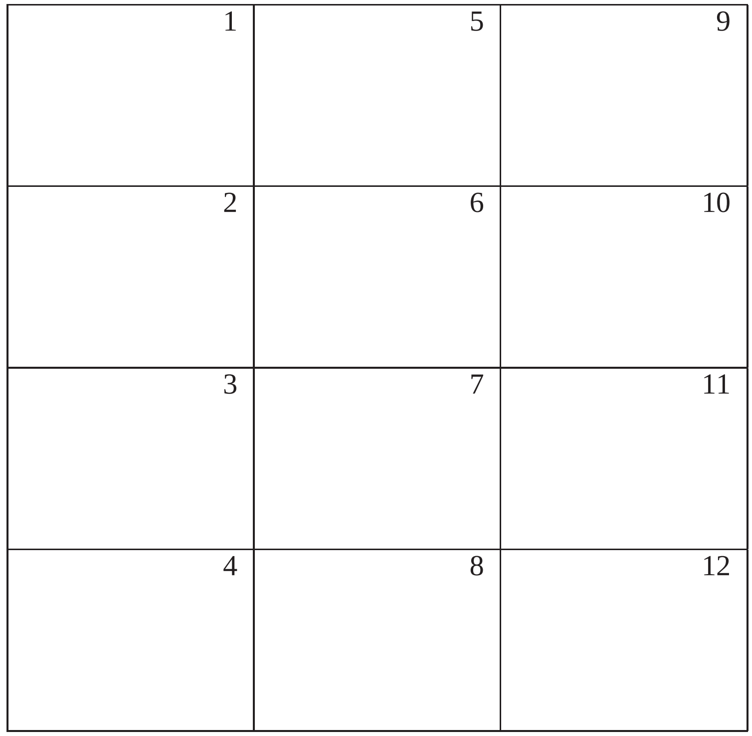


Figure 3.10. Page Layout of the Decision Tree Diagram

The number of pages that are produced depends on the size of the tree and on the number of print positions that are available in the horizontal and vertical directions. Table 3.28 lists all options you can use to control the number of pages.

Table 3.28. Options that Control the Number of Pages

Option	Effect
DISPLAY=	amounts of information displayed on the diagram
MAXPREC=	maximum decimal width allowed (the precision) to format numerical values into $w.d$ format
MAXWIDTH=	maximum field width allowed to format numerical values
NOLABEL	no labels are displayed on the diagram
NWIDTH=	maximum field width allowed to format outcome names
YBETWEEN=	vertical spaces between two successive end nodes

If the GRAPHICS option is used, the following options can be used to control the number of pages:

- The COMPRESS option draws the entire decision tree on one page.
- The HSYMBOL= option controls the height of all symbols.
- The HTEXT= option controls the height of text in the tree.
- The HEIGHT= option in a SYMBOL definition specifies the height of a symbol.
- The HTEXT= option in a GOPTIONS statement specifies the height of all text.

- The HTITLE= option in a GOPTIONS statement specifies the height of the first title line.
- The HPOS= and VPOS= options in a GOPTIONS statement changes the number of rows and columns.

Note that the font used for all text may also affect the number of pages needed. Some fonts take more space than others.

If the decision tree diagram is produced on a line printer, you can use the FORMCHAR= option to control the appearance the links and the junctions of the diagram. When the GRAPHICS options is specified, several options are available to enhance the appearance of the decision tree diagram. These are described in the "Graphics Options" section on page 272. In addition, there are many other options available in the GOPTIONS statement and the SYMBOL statement for controlling the details of graphics output. Refer to the relevant chapters in *SAS/GRAPH Software: Reference* for a detailed discussion of the GOPTIONS and SYMBOL statements.

Table 3.29, Table 3.30, and Table 3.31, show the relationship among the options for controlling the appearance of texts, nodes, and links, respectively. The order that PROC DTREE uses in determining which option is in effect is also provided.

For ODS purposes, the label of the decision tree diagram drawn in line-printer quality is "Treeplot."

Table 3.29. Options that Control Text Appearance

Object	Specification	Search Order
Text	Font	1. the FTEXT= option 2. the FTEXT= option in a GOPTIONS statement 3. hardware font
	Color	1. the CTEXT= option 2. the CTEXT= option in a GOPTIONS statement 3. the first color in the colors list
	Height	1. the value of the HTEXT= option¶ times the value of the HTEXT= option‖ in a GOPTIONS statement

¶If this option is not specified, the default value 1 is used.
‖The default value of this option is 1 unit.

Table 3.30. Options that Control Node Appearance

Object	Specification	Search Order
Chance Nodes	Symbol	1. the VSYMBOLC= option 2. the VALUE= and FONT= options in the *m*th generated SYMBOL definition, if SYMBOLC=*m* is used 3. the default symbol, CIRCLE
	Color	1. the CSYMBOLC= option 2. the CV= option in the *m*th generated SYMBOL definition, if SYMBOLC=*m* is used 3. the CSYMBOL= option in a GOPTIONS statement 4. the fifth color in the colors list
	Height	1. *h* times the value of the HEIGHT= option in the *m*th generated SYMBOL definition, if both the HSYMBOL=*h* and the SYMBOLC=*m* are specified 2. the HSYMBOL= option, if it is specified 3. the HEIGHT= option in the *m*th generated symbol definition, if SYMBOLC=*m* is used. 4. the default value, 1 cell
Decision Nodes	Symbol	1. the VSYMBOLD= option 2. the VALUE= and FONT= options in the *d*th generated SYMBOL definition, if SYMBOLD=*d* is used 3. the default value, SQUARE
	Color	1. the CSYMBOLD= option 2. the CV= option in the *d*th generated SYMBOL definition, if SYMBOLD=*d* is used 3. the CSYMBOL= option in a GOPTIONS statement 4. the fourth color in the colors list
	Height	1. *h* times the value of the HEIGHT= option in the *d*th generated SYMBOL definition, if both the HSYMBOL=*h* and the SYMBOLD=*d* are specified 2. the HSYMBOL= option, if it is specified 3. the HEIGHT= option in the *d*th generated symbol definition, if SYMBOLD=*d* is used. 4. the default value, 1 cell
End Nodes	Symbol	1. the VSYMBOLE= option 2. the VALUE= and FONT= options in the *n*th generated SYMBOL definition, if SYMBOLE=*n* is used 3. the default value, DOT
	Color	1. the CSYMBOLE= option 2. the CV= option in the *n*th generated SYMBOL definition if the option SYMBOLE=*n* is specified 3. the CSYMBOL= option in a GOPTIONS statement 4. the sixth color in the colors list
	Height	1. *h* times the value of the HEIGHT= option in the *n*th generated SYMBOL definition, if both the HSYMBOL=*h* and the SYMBOLE=*n* are specified 2. the HSYMBOL= option, if it is specified 3. the HEIGHT= option in the *n*th generated symbol definition, if SYMBOLE=*n* is used. 4. the default value, 1 cell

Table 3.31. Options that Control Link Appearance

Object	Specification	Search Order
Links for Regular Outcomes	Type	1. the LSTYLE= option 2. the LINE= in the *i*th generated SYMBOL definition, if LINKA=*i* is used 3. the default value, 1 (solid line)
	Color	1. the CLINK= option 2. the CI= option in the *i*th generated SYMBOL definition, if LINKA=*i* is used 3. the third color in the colors list
	Thickness	1. the LWIDTH= option 2. the WIDTH= option in the *i*th generated SYMBOL definition, if LINKA=*i* is used 3. the default value, 1
Links for Optimal Decision	Type	1. the LSTYLEB= option 2. the LINE= in the *j*th generated SYMBOL definition, if LINKB=*j* is used 3. the default value, 1 (solid line)
	Color	1. the CBEST= option 2. the CI= option in the *j*th generated SYMBOL definition, if LINKB=*j* is used 3. the second color in the colors list
	Thickness	1. the LWIDTHB= option 2. the WIDTH= option in the *j*th generated SYMBOL definition, if LINKB=*j* is used 3. 2 times the thickness of links that represent regular outcomes
Links That Fall Across Pages	Type	1. the LSTYLEC= option 2. the LINE= in the *k*th generated SYMBOL definition, if LINKC=*k* is used 3. the default value, 2 (dot line)
	Color	1. depends on whether or not it represents an optimal decision
	Thickness	1. depends on whether or not it represents an optimal decision

Web-Enabled Decision Tree

The WEB= variable in the STAGEIN= data set enables you to define an HTML reference for each stage. This HTML reference is currently associated with all the decision tree nodes that correspond to the stage. The WEB= variable is a character variable, and the values need to be of the form HREF=*htmlpage*.

In addition, you can also store the coordinate and link information defined via the WEB= option in a SAS data set by specifying the IMAGEMAP= option in the PROC DTREE statement or in the TREEPLOT statement. By processing this SAS data set using a DATA step, you can generate customized HTML pages for your decision tree diagram.

ODS Table Names

PROC DTREE assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, refer to the chapter on ODS in the *SAS/STAT User's Guide*.

Table 3.32. ODS Tables Produced in PROC DTREE

ODS Table Name	Description	Statement / Option
Parameters	Decision parameters	SUMMARY or EVALUATE / SUMMARY
Policy	Optimal decision policy	SUMMARY or EVALUATE / SUMMARY
Stages	List of stages in order	SUMMARY or EVALUATE / SUMMARY
Treeplot	Line-printer plot of decision tree	TREEPLOT / LINEPRINTER

Precision Errors

When PROC DTREE detects an error, it displays a message on the SAS log to call it to your attention. If the error is in a statement other than the PROC DTREE statement and the QUIT statement, and if the ERRHANDLE=QUIT option is not specified, the procedure ignores the erroneous statement and waits for you to enter another statement. This gives you a chance to correct the mistake you made and keep running. You can exit the procedure at any time by specifying the QUIT statement.

If the error is in an input data set, typically, you will have to edit the data set and then reinvoke PROC DTREE. In one case, however, you can use an option to correct the problem. You may receive an error message indicating that the sum of probabilities for a particular chance stage does not equal 1.0. If it is caused by roundoff errors in the summation, then you can reset the TOLERANCE= option to correct this error. For example, suppose that your problem contains a chance stage that has three outcomes, '**Out1**', '**Out2**' and '**Out3**', and each has probability $1/3$. Suppose also that you input their probabilities in the PROBIN= data set as follows:

```
Out1    Out2    Out3    0.3333  0.3333  0.3333
```

Then, PROC DTREE detects the total probabilities for that stage as 0.9999, not equal to 1, and hence displays an error message. The following RESET statement fixes the error:

```
reset tolerance=0.00015;
```

Alternatively, you can specify the AUTOSCALE option to ask the procedure to rescale the probabilities whenever this situation occurs.

Computer Resource Requirements

There is no inherent limit on the size of the decision tree model that can be evaluated and analyzed with the DTREE procedure. The number of stages and outcomes are constrained only by the amount of memory available. Naturally, there needs to be a sufficient amount of core memory available in order to invoke and initialize the SAS system. Furthermore, more memory is required to load the graphics sublibrary if the GRAPHICS option is specified. As far as possible, the procedure attempts to store all the data in core memory. However, if the problem is too large to fit in core memory, the procedure resorts to the use of utility data sets and swaps between core memory and utility data sets as necessary.

The storage requirement for the data area required by the procedure is proportional to the number of stages and outcomes as well as the number of nodes* in the decision tree model. The time required depends heavily on the number of nodes in the decision tree.

Examples

This section contains six examples that illustrate several features and applications of the DTREE procedure. The aim of this section is to show you how to use PROC DTREE to solve your decision problem and gain valuable insight into its structure.

Example 3.1 on page 305 and Example 3.2 on page 310 show two methods frequently used to spread the risk of a venture: buy insurance and enter a partnership. Example 3.1 also illustrates the use of the VARIABLE statement to identify the variables in the input data sets. Example 3.3 on page 322 illustrates the use of the graphics options to produce a graphics quality decision tree diagram. Example 3.4 on page 326 illustrates the use of SYMBOL and GOPTIONS statements and the Annotate facility to control the appearance of the decision tree diagram. Example 3.5 on page 330 demonstrates an application of PROC DTREE for financial decision problems. It also illustrates a situation where redundant data are necessary to determine the value of information. In addition, it shows a case where the results from the VPI and VPC statements are misleading if they are used without care. Example 3.6 on page 340 shows an application in litigation, a sophisticated use of sensitivity analysis. It also shows you how to deal with the value of future money.

Finally, Table 3.40 (page 350) and Table 3.41 (page 351) list all the examples in this chapter, and the options and statements in the DTREE procedure that are illustrated by each example.

*The number of nodes depends on the number of stages and the number of outcomes for each stage.

Example 3.1. Oil Wildcatter's Problem with an Insurance

Again consider the oil wildcatter's problem introduced in the "Introductory Example" section beginning on page 252. Suppose that the wildcatter is concerned that the probability of a dry well may be as high as 0.5.

The wildcatter has learned that an insurance company is willing to offer him a policy that, with a premium of $130,000, will redeem $200,000 if the well is dry. He would like to include the alternative of buying insurance into his analysis. One way to do this is to include a stage for this decision in the model. The following DATA step reads this new decision problem into the STAGEIN= data set named Dtoils4. Notice the new stage named '**Insurance**', which represents the decision of whether or not to buy the insurance. Also notice that the cost of the insurance is represented as a negative reward of $130,000.

```
   /* -- create the STAGEIN= data set                  -- */
data Dtoils4;
   input Stage $12. Stype  $4. Outcome $16. Succ $12.
         Premium dollar12.0;
   datalines;
Drill       D   Drill           Insurance            .
.           .   Not_Drill       .                    .
Insurance   D   Buy_Insurance   Cost          -$130,000
.           .   Do_Not_Buy      Cost                 .
Cost        C   Low             Oil_Deposit          .
.           .   Fair            Oil_Deposit          .
.           .   High            Oil_Deposit          .
Oil_Deposit C   Dry             .                    .
.           .   Wet             .                    .
.           .   Soaking         .                    .
;
```

Probabilities associated with the uncertain events are given in the PROBIN= data set named Dtoilp4. Except for the order of the variables in this data set, it is the same as the Dtoilp1 data set given in the "Introductory Example" section beginning on page 252.

```
   /* -- create the PROBIN= data set                   -- */
data Dtoilp4;
   input (V1-V3) ($12.) (P1-P3) (8.2);
   datalines;
Low         Fair        High        0.2     0.6     0.2
Dry         Wet         Soaking     0.5     0.3     0.2
;
```

The payoffs for this problem are now calculated to include the cost and value of the insurance. The following DATA step does this.

```
   /* -- create PAYOFFS= data set                      -- */
data Dtoilu4;
   input (Cost Deposit Drill Insuran ) ($16.) ;
   format Payoff dollar12.0;

   /* determine the cost for this scenario */
   if      Cost='Low'  then Rcost=150000;
   else if Cost='Fair' then Rcost=300000;
   else                     Rcost=500000;

   /* determine the oil deposit and the corresponding  */
   /* net payoff for this scenario                     */
   if      Deposit='Dry' then Return=0;
   else if Deposit='Wet' then Return=700000;
   else                       Return=1200000;


      /* calculate the net return for this scenario */
   if      Drill='Not_Drill' then Payoff=0;
   else                           Payoff=Return-Rcost;

   /* determine redeem received for this scenario */
   if Insuran='Buy_Insurance' and Deposit='Dry' then
      Payoff=Payoff+200000;

   /* drop unneeded variables */
   drop Rcost Return;

   datalines;
Low             Dry             Not_Drill       .
Low             Dry             Drill           Buy_Insurance
Low             Dry             Drill           Do_Not_Buy
Low             Wet             Not_Drill       .
Low             Wet             Drill           Buy_Insurance
Low             Wet             Drill           Do_Not_Buy
Low             Soaking         Not_Drill       .
Low             Soaking         Drill           Buy_Insurance
Low             Soaking         Drill           Do_Not_Buy
Fair            Dry             Not_Drill       .
Fair            Dry             Drill           Buy_Insurance
Fair            Dry             Drill           Do_Not_Buy
Fair            Wet             Not_Drill       .
Fair            Wet             Drill           Buy_Insurance
Fair            Wet             Drill           Do_Not_Buy
Fair            Soaking         Not_Drill       .
Fair            Soaking         Drill           Buy_Insurance
Fair            Soaking         Drill           Do_Not_Buy
High            Dry             Not_Drill       .
High            Dry             Drill           Buy_Insurance
High            Dry             Drill           Do_Not_Buy
High            Wet             Not_Drill       .
```

```
High            Wet             Drill           Buy_Insurance
High            Wet             Drill           Do_Not_Buy
High            Soaking         Not_Drill       .
High            Soaking         Drill           Buy_Insurance
High            Soaking         Drill           Do_Not_Buy
;
```

The payoff table can be displayed with the following PROC PRINT statement:

```
   /* -- print the payoff table          -- */
title "Oil Wildcatter's Problem";
title3 "The Payoffs";

proc print data=Dtoilu4;
run;
```

The table is shown in Output 3.1.1.

Output 3.1.1. Payoffs of the Oil Wildcatter's Problem with an Insurance Option

Oil Wildcatter's Problem

The Payoffs

Obs	Cost	Deposit	Drill	Insuran	Payoff
1	Low	Dry	Not_Drill		$0
2	Low	Dry	Drill	Buy_Insurance	$50,000
3	Low	Dry	Drill	Do_Not_Buy	$-150,000
4	Low	Wet	Not_Drill		$0
5	Low	Wet	Drill	Buy_Insurance	$550,000
6	Low	Wet	Drill	Do_Not_Buy	$550,000
7	Low	Soaking	Not_Drill		$0
8	Low	Soaking	Drill	Buy_Insurance	$1,050,000
9	Low	Soaking	Drill	Do_Not_Buy	$1,050,000
10	Fair	Dry	Not_Drill		$0
11	Fair	Dry	Drill	Buy_Insurance	$-100,000
12	Fair	Dry	Drill	Do_Not_Buy	$-300,000
13	Fair	Wet	Not_Drill		$0
14	Fair	Wet	Drill	Buy_Insurance	$400,000
15	Fair	Wet	Drill	Do_Not_Buy	$400,000
16	Fair	Soaking	Not_Drill		$0
17	Fair	Soaking	Drill	Buy_Insurance	$900,000
18	Fair	Soaking	Drill	Do_Not_Buy	$900,000
19	High	Dry	Not_Drill		$0
20	High	Dry	Drill	Buy_Insurance	$-300,000
21	High	Dry	Drill	Do_Not_Buy	$-500,000
22	High	Wet	Not_Drill		$0
23	High	Wet	Drill	Buy_Insurance	$200,000
24	High	Wet	Drill	Do_Not_Buy	$200,000
25	High	Soaking	Not_Drill		$0
26	High	Soaking	Drill	Buy_Insurance	$700,000
27	High	Soaking	Drill	Do_Not_Buy	$700,000

To find the optimal decision, call PROC DTREE with the following statements:

```
   /* -- PROC DTREE statements                         -- */
title "Oil Wildcatter's Problem";

proc dtree stagein=Dtoils4
           probin=Dtoilp4
           payoffs=Dtoilu4
           nowarning
           ;

  variables / stage=Stage type=Stype outcome=(Outcome)
              reward=(Premium) successor=(Succ)
              event=(V1 V2 V3) prob=(P1 P2 P3)
              state=(Cost Deposit Drill Insuran)
              payoff=(Payoff);

   evaluate;

   summary / target=Insurance;
```

The VARIABLES statement identifies the variables in the input data sets. The yield of the optimal decision is written to the SAS log as:

```
NOTE: Present order of stages:

      Drill(D), Insurance(D), Cost(C), Oil_Deposit(C),
      _ENDST_(E).

NOTE: The currently optimal decision yields 140000.
```

The optimal decision summary produced by the SUMMARY statements are shown in Output 3.1.2. The summary in Output 3.1.2 shows that the insurance policy is worth $\$240,000 - \$140,000 = \$100,000$, but since it costs $\$130,000$, the wildcatter should reject such an insurance policy.

Output 3.1.2. Summary of the Oil Wildcatter's Problem

```
                        Oil Wildcatter's Problem

                          The DTREE Procedure
                        Optimal Decision Summary

                            Order of Stages

                        Stage           Type
                        -----------------------
                        Drill           Decision
                        Insurance       Decision
                        Cost            Chance
                        Oil_Deposit     Chance
                        _ENDST_         End


                          Decision Parameters

     Decision Criterion:    Maximize Expected Value (MAXEV)
Optimal Decision Yields:    $140,000


                        Optimal Decision Policy

                         Up to Stage Insurance

                                        Cumulative      Evaluating
   Alternatives or Outcomes                 Reward           Value
------------------------------------------------------------------
Drill            Buy_Insurance             -130000       $240,000
Drill            Do_Not_Buy                      0       $140,000*
```

Now assume that the oil wildcatter is risk averse and has an exponential utility function with a risk tolerance of $\$1,200,000$. In order to evaluate his problem based on this decision criterion, the wildcatter reevaluates the problem with the following statements:

```
reset criterion=maxce rt=1200000;
summary / target=Insurance;
```

The output from PROC DTREE given in Output 3.1.3 shows that the decision to purchase an insurance policy is favorable in the risk-averse environment. Note that an EVALUATE statement is not necessary before the SUMMARY statement. PROC DTREE evaluates the decision tree automatically when the decision criterion has been changed using the RESET statement.

Output 3.1.3. Summary of the Oil Wildcatter's Problem with RT = $1,200,000$

```
                      Oil Wildcatter's Problem

                        The DTREE Procedure
                      Optimal Decision Summary

                          Order of Stages

                      Stage          Type
                      -----------------------
                      Drill          Decision
                      Insurance      Decision
                      Cost           Chance
                      Oil_Deposit    Chance
                      _ENDST_        End


                        Decision Parameters

      Decision Criterion:    Maximize Certain Equivalent Value (MAXCE)
          Risk Tolerance:    $1,200,000
 Optimal Decision Yields:    $45,728


                      Optimal Decision Policy

                       Up to Stage Insurance

                                          Cumulative      Evaluating
         Alternatives or Outcomes             Reward           Value
      ------------------------------------------------------------
      Drill           Buy_Insurance          -130000       $175,728*
      Drill           Do_Not_Buy                   0        $44,499
```

Example 3.2. Oil Wildcatter's Problem in Risk Averse Setting

Continuing with the oil wildcatter's problem, suppose that in addition to possibly buying insurance to spread the risk of the venture, the wildcatter is considering sharing the risk by selling a portion of this venture to other investors. Now, the decision he faces is whether to buy insurance or not and what percentage of the investment to divest. Again, assume that the wildcatter is risk averse with a risk tolerance of $\$1,200,000$. Notice that in the program that follows the '**`Divestment`**' decision includes possibilities of no divestment to 100% divestment in 10% increments.

```
   /* -- create the STAGEIN= data set                 -- */
data Dtoils4;
   input _STNAME_ $16. _STTYPE_  $12. _OUTCOM_ $16.
                                      _SUCCES_ $16. ;
   datalines;
Divestment      Decision    No_Divestment      Insurance
.               .           10%_Divestment     Insurance
.               .           20%_Divestment     Insurance
.               .           30%_Divestment     Insurance
.               .           40%_Divestment     Insurance
.               .           50%_Divestment     Insurance
.               .           60%_Divestment     Insurance
.               .           70%_Divestment     Insurance
.               .           80%_Divestment     Insurance
.               .           90%_Divestment     Insurance
.               .           100%_Divestment    .
Insurance       Decision    Buy_Insurance      Cost
.               .           Do_Not_Buy         Cost
Cost            Chance      Low                Oil_Deposit
.               .           Fair               Oil_Deposit
.               .           High               Oil_Deposit
Oil_Deposit     Chance      Dry                .
.               .           Wet                .
.               .           Soaking            .
;
```

The probabilities associated with the uncertain events are given in the PROBIN= data set named Dtoilp4. Except for the order of the variables in this data set, it is the same as the Dtoilp1 data set used in the "Introductory Example" section beginning on page 252.

```
   /* -- create the PROBIN= data set                  -- */
data Dtoilp4;
   input _EVENT1 $12. _PROB1 8.2 _EVENT3 $12. _PROB3 8.2;
   datalines;
Low         0.2     Dry         0.5
Fair        0.6     Wet         0.3
High        0.2     Soaking     0.2
;

   /* -- create the PAYOFFS= data set                 -- */
data Dtoilu4(drop=i j k l);
   length _STATE1-_STATE4 $16. ;
   format _VALUE_ dollar12.0;
```

```
      /* define and initialize arrays */
array DIVEST{11}  $16. _TEMPORARY_ ('No_Divestment',
                                    '10%_Divestment',
                                    '20%_Divestment',
                                    '30%_Divestment',
                                    '40%_Divestment',
                                    '50%_Divestment',
                                    '60%_Divestment',
                                    '70%_Divestment',
                                    '80%_Divestment',
                                    '90%_Divestment',
                                    '100%_Divestment'  );
   array INSUR{3} $16.    _TEMPORARY_ ('Do_Not_Buy',
                                       'Buy_Insurance',
                                       ''                );
   array COST{4} $        _TEMPORARY_ ('Low',
                                       'Fair',
                                       'High',
                                       ''                );
   array DEPOSIT{4} $     _TEMPORARY_ ('Dry',
                                       'Wet',
                                       'Soaking',
                                       ''                );

   do i=1 to 10;            /* loop for each divestment */
      _STATE1=DIVEST{i};

         /* determine the percentage of ownership */
         /* retained for this scenario            */
      PCT=1.0-((i-1)*0.1);

      do j=1 to 2;       /* loop for insurance decision */
         _STATE2=INSUR{j};

            /* determine the premium need to pay  */
            /* for this scenario                  */
         if _STATE2='Buy_Insurance' then PREMIUM=130000;
         else                            PREMIUM=0;

         do k=1 to 3;        /* loop for each well cost */
            _STATE3=COST{k};

               /* determine the cost for this scenario  */
            if      _STATE3='Low' then  _COST_=150000;
            else if _STATE3='Fair' then _COST_=300000;
            else                        _COST_=500000;
```

```
            do l=1 to 3;  /* loop for each deposit type */
               _STATE4=DEPOSIT{l};

                  /* determine the oil deposit and the  */
                  /* corresponding net payoff for this  */
                  /* scenario                           */
               if      _STATE4='Dry' then _PAYOFF_=0;
               else if _STATE4='Wet' then _PAYOFF_=700000;
               else                       _PAYOFF_=1200000;

                  /* determine redeem received for this */
                  /* scenario                           */
               if _STATE2='Buy_Insurance' and _STATE4='Dry' then
                    REDEEM=200000;
               else REDEEM=0;

                  /* calculate the net return for this  */
                  /*scenario                            */
               _VALUE_=(_PAYOFF_-_COST_-PREMIUM+REDEEM)*PCT;

                  /* drop unneeded variables */
               drop _COST_ _PAYOFF_ PREMIUM REDEEM PCT;

                  /* output this record */
               output;
            end;
         end;
      end;
   end;

      /* output an observation for the scenario */
      /* 100%_Divestment                        */
   _STATE1=DIVEST{11};
   _STATE2=INSUR{3};
   _STATE3=COST{4};
   _STATE4=DEPOSIT{4};
   _VALUE_=0;
   output;

run;
```

The Dtoilu4 data set for this problem, which contains 181 observations and 5 variables, is displayed on the following pages.

Output 3.2.1. Payoffs of the Oil Wildcatter's Problem with Risk Sharing

Oil Wildcatter's Problem

The Payoffs

Obs	_STATE1	_STATE2	_STATE3	_STATE4	_VALUE_
1	No_Divestment	Do_Not_Buy	Low	Dry	$-150,000
2	No_Divestment	Do_Not_Buy	Low	Wet	$550,000
3	No_Divestment	Do_Not_Buy	Low	Soaking	$1,050,000
4	No_Divestment	Do_Not_Buy	Fair	Dry	$-300,000
5	No_Divestment	Do_Not_Buy	Fair	Wet	$400,000
6	No_Divestment	Do_Not_Buy	Fair	Soaking	$900,000
7	No_Divestment	Do_Not_Buy	High	Dry	$-500,000
8	No_Divestment	Do_Not_Buy	High	Wet	$200,000
9	No_Divestment	Do_Not_Buy	High	Soaking	$700,000
10	No_Divestment	Buy_Insurance	Low	Dry	$-80,000
11	No_Divestment	Buy_Insurance	Low	Wet	$420,000
12	No_Divestment	Buy_Insurance	Low	Soaking	$920,000
13	No_Divestment	Buy_Insurance	Fair	Dry	$-230,000
14	No_Divestment	Buy_Insurance	Fair	Wet	$270,000
15	No_Divestment	Buy_Insurance	Fair	Soaking	$770,000
16	No_Divestment	Buy_Insurance	High	Dry	$-430,000
17	No_Divestment	Buy_Insurance	High	Wet	$70,000
18	No_Divestment	Buy_Insurance	High	Soaking	$570,000
19	10%_Divestment	Do_Not_Buy	Low	Dry	$-135,000
20	10%_Divestment	Do_Not_Buy	Low	Wet	$495,000
21	10%_Divestment	Do_Not_Buy	Low	Soaking	$945,000
22	10%_Divestment	Do_Not_Buy	Fair	Dry	$-270,000
23	10%_Divestment	Do_Not_Buy	Fair	Wet	$360,000
24	10%_Divestment	Do_Not_Buy	Fair	Soaking	$810,000
25	10%_Divestment	Do_Not_Buy	High	Dry	$-450,000
26	10%_Divestment	Do_Not_Buy	High	Wet	$180,000
27	10%_Divestment	Do_Not_Buy	High	Soaking	$630,000
28	10%_Divestment	Buy_Insurance	Low	Dry	$-72,000
29	10%_Divestment	Buy_Insurance	Low	Wet	$378,000
30	10%_Divestment	Buy_Insurance	Low	Soaking	$828,000
31	10%_Divestment	Buy_Insurance	Fair	Dry	$-207,000
32	10%_Divestment	Buy_Insurance	Fair	Wet	$243,000
33	10%_Divestment	Buy_Insurance	Fair	Soaking	$693,000
34	10%_Divestment	Buy_Insurance	High	Dry	$-387,000
35	10%_Divestment	Buy_Insurance	High	Wet	$63,000
36	10%_Divestment	Buy_Insurance	High	Soaking	$513,000
37	20%_Divestment	Do_Not_Buy	Low	Dry	$-120,000
38	20%_Divestment	Do_Not_Buy	Low	Wet	$440,000
39	20%_Divestment	Do_Not_Buy	Low	Soaking	$840,000
40	20%_Divestment	Do_Not_Buy	Fair	Dry	$-240,000
41	20%_Divestment	Do_Not_Buy	Fair	Wet	$320,000
42	20%_Divestment	Do_Not_Buy	Fair	Soaking	$720,000
43	20%_Divestment	Do_Not_Buy	High	Dry	$-400,000
44	20%_Divestment	Do_Not_Buy	High	Wet	$160,000
45	20%_Divestment	Do_Not_Buy	High	Soaking	$560,000
46	20%_Divestment	Buy_Insurance	Low	Dry	$-64,000
47	20%_Divestment	Buy_Insurance	Low	Wet	$336,000
48	20%_Divestment	Buy_Insurance	Low	Soaking	$736,000

Oil Wildcatter's Problem

The Payoffs

Obs	_STATE1	_STATE2	_STATE3	_STATE4	_VALUE_
49	20%_Divestment	Buy_Insurance	Fair	Dry	$-184,000
50	20%_Divestment	Buy_Insurance	Fair	Wet	$216,000
51	20%_Divestment	Buy_Insurance	Fair	Soaking	$616,000
52	20%_Divestment	Buy_Insurance	High	Dry	$-344,000
53	20%_Divestment	Buy_Insurance	High	Wet	$56,000
54	20%_Divestment	Buy_Insurance	High	Soaking	$456,000
55	30%_Divestment	Do_Not_Buy	Low	Dry	$-105,000
56	30%_Divestment	Do_Not_Buy	Low	Wet	$385,000
57	30%_Divestment	Do_Not_Buy	Low	Soaking	$735,000
58	30%_Divestment	Do_Not_Buy	Fair	Dry	$-210,000
59	30%_Divestment	Do_Not_Buy	Fair	Wet	$280,000
60	30%_Divestment	Do_Not_Buy	Fair	Soaking	$630,000
61	30%_Divestment	Do_Not_Buy	High	Dry	$-350,000
62	30%_Divestment	Do_Not_Buy	High	Wet	$140,000
63	30%_Divestment	Do_Not_Buy	High	Soaking	$490,000
64	30%_Divestment	Buy_Insurance	Low	Dry	$-56,000
65	30%_Divestment	Buy_Insurance	Low	Wet	$294,000
66	30%_Divestment	Buy_Insurance	Low	Soaking	$644,000
67	30%_Divestment	Buy_Insurance	Fair	Dry	$-161,000
68	30%_Divestment	Buy_Insurance	Fair	Wet	$189,000
69	30%_Divestment	Buy_Insurance	Fair	Soaking	$539,000
70	30%_Divestment	Buy_Insurance	High	Dry	$-301,000
71	30%_Divestment	Buy_Insurance	High	Wet	$49,000
72	30%_Divestment	Buy_Insurance	High	Soaking	$399,000
73	40%_Divestment	Do_Not_Buy	Low	Dry	$-90,000
74	40%_Divestment	Do_Not_Buy	Low	Wet	$330,000
75	40%_Divestment	Do_Not_Buy	Low	Soaking	$630,000
76	40%_Divestment	Do_Not_Buy	Fair	Dry	$-180,000
77	40%_Divestment	Do_Not_Buy	Fair	Wet	$240,000
78	40%_Divestment	Do_Not_Buy	Fair	Soaking	$540,000
79	40%_Divestment	Do_Not_Buy	High	Dry	$-300,000
80	40%_Divestment	Do_Not_Buy	High	Wet	$120,000
81	40%_Divestment	Do_Not_Buy	High	Soaking	$420,000
82	40%_Divestment	Buy_Insurance	Low	Dry	$-48,000
83	40%_Divestment	Buy_Insurance	Low	Wet	$252,000
84	40%_Divestment	Buy_Insurance	Low	Soaking	$552,000
85	40%_Divestment	Buy_Insurance	Fair	Dry	$-138,000
86	40%_Divestment	Buy_Insurance	Fair	Wet	$162,000
87	40%_Divestment	Buy_Insurance	Fair	Soaking	$462,000
88	40%_Divestment	Buy_Insurance	High	Dry	$-258,000
89	40%_Divestment	Buy_Insurance	High	Wet	$42,000
90	40%_Divestment	Buy_Insurance	High	Soaking	$342,000
91	50%_Divestment	Do_Not_Buy	Low	Dry	$-75,000
92	50%_Divestment	Do_Not_Buy	Low	Wet	$275,000
93	50%_Divestment	Do_Not_Buy	Low	Soaking	$525,000
94	50%_Divestment	Do_Not_Buy	Fair	Dry	$-150,000
95	50%_Divestment	Do_Not_Buy	Fair	Wet	$200,000
96	50%_Divestment	Do_Not_Buy	Fair	Soaking	$450,000

Oil Wildcatter's Problem

The Payoffs

Obs	_STATE1	_STATE2	_STATE3	_STATE4	_VALUE_
97	50%_Divestment	Do_Not_Buy	High	Dry	$-250,000
98	50%_Divestment	Do_Not_Buy	High	Wet	$100,000
99	50%_Divestment	Do_Not_Buy	High	Soaking	$350,000
100	50%_Divestment	Buy_Insurance	Low	Dry	$-40,000
101	50%_Divestment	Buy_Insurance	Low	Wet	$210,000
102	50%_Divestment	Buy_Insurance	Low	Soaking	$460,000
103	50%_Divestment	Buy_Insurance	Fair	Dry	$-115,000
104	50%_Divestment	Buy_Insurance	Fair	Wet	$135,000
105	50%_Divestment	Buy_Insurance	Fair	Soaking	$385,000
106	50%_Divestment	Buy_Insurance	High	Dry	$-215,000
107	50%_Divestment	Buy_Insurance	High	Wet	$35,000
108	50%_Divestment	Buy_Insurance	High	Soaking	$285,000
109	60%_Divestment	Do_Not_Buy	Low	Dry	$-60,000
110	60%_Divestment	Do_Not_Buy	Low	Wet	$220,000
111	60%_Divestment	Do_Not_Buy	Low	Soaking	$420,000
112	60%_Divestment	Do_Not_Buy	Fair	Dry	$-120,000
113	60%_Divestment	Do_Not_Buy	Fair	Wet	$160,000
114	60%_Divestment	Do_Not_Buy	Fair	Soaking	$360,000
115	60%_Divestment	Do_Not_Buy	High	Dry	$-200,000
116	60%_Divestment	Do_Not_Buy	High	Wet	$80,000
117	60%_Divestment	Do_Not_Buy	High	Soaking	$280,000
118	60%_Divestment	Buy_Insurance	Low	Dry	$-32,000
119	60%_Divestment	Buy_Insurance	Low	Wet	$168,000
120	60%_Divestment	Buy_Insurance	Low	Soaking	$368,000
121	60%_Divestment	Buy_Insurance	Fair	Dry	$-92,000
122	60%_Divestment	Buy_Insurance	Fair	Wet	$108,000
123	60%_Divestment	Buy_Insurance	Fair	Soaking	$308,000
124	60%_Divestment	Buy_Insurance	High	Dry	$-172,000
125	60%_Divestment	Buy_Insurance	High	Wet	$28,000
126	60%_Divestment	Buy_Insurance	High	Soaking	$228,000
127	70%_Divestment	Do_Not_Buy	Low	Dry	$-45,000
128	70%_Divestment	Do_Not_Buy	Low	Wet	$165,000
129	70%_Divestment	Do_Not_Buy	Low	Soaking	$315,000
130	70%_Divestment	Do_Not_Buy	Fair	Dry	$-90,000
131	70%_Divestment	Do_Not_Buy	Fair	Wet	$120,000
132	70%_Divestment	Do_Not_Buy	Fair	Soaking	$270,000
133	70%_Divestment	Do_Not_Buy	High	Dry	$-150,000
134	70%_Divestment	Do_Not_Buy	High	Wet	$60,000
135	70%_Divestment	Do_Not_Buy	High	Soaking	$210,000
136	70%_Divestment	Buy_Insurance	Low	Dry	$-24,000
137	70%_Divestment	Buy_Insurance	Low	Wet	$126,000
138	70%_Divestment	Buy_Insurance	Low	Soaking	$276,000
139	70%_Divestment	Buy_Insurance	Fair	Dry	$-69,000
140	70%_Divestment	Buy_Insurance	Fair	Wet	$81,000
141	70%_Divestment	Buy_Insurance	Fair	Soaking	$231,000
142	70%_Divestment	Buy_Insurance	High	Dry	$-129,000
143	70%_Divestment	Buy_Insurance	High	Wet	$21,000
144	70%_Divestment	Buy_Insurance	High	Soaking	$171,000

Oil Wildcatter's Problem

The Payoffs

Obs	_STATE1	_STATE2	_STATE3	_STATE4	_VALUE_
145	80%_Divestment	Do_Not_Buy	Low	Dry	$-30,000
146	80%_Divestment	Do_Not_Buy	Low	Wet	$110,000
147	80%_Divestment	Do_Not_Buy	Low	Soaking	$210,000
148	80%_Divestment	Do_Not_Buy	Fair	Dry	$-60,000
149	80%_Divestment	Do_Not_Buy	Fair	Wet	$80,000
150	80%_Divestment	Do_Not_Buy	Fair	Soaking	$180,000
151	80%_Divestment	Do_Not_Buy	High	Dry	$-100,000
152	80%_Divestment	Do_Not_Buy	High	Wet	$40,000
153	80%_Divestment	Do_Not_Buy	High	Soaking	$140,000
154	80%_Divestment	Buy_Insurance	Low	Dry	$-16,000
155	80%_Divestment	Buy_Insurance	Low	Wet	$84,000
156	80%_Divestment	Buy_Insurance	Low	Soaking	$184,000
157	80%_Divestment	Buy_Insurance	Fair	Dry	$-46,000
158	80%_Divestment	Buy_Insurance	Fair	Wet	$54,000
159	80%_Divestment	Buy_Insurance	Fair	Soaking	$154,000
160	80%_Divestment	Buy_Insurance	High	Dry	$-86,000
161	80%_Divestment	Buy_Insurance	High	Wet	$14,000
162	80%_Divestment	Buy_Insurance	High	Soaking	$114,000
163	90%_Divestment	Do_Not_Buy	Low	Dry	$-15,000
164	90%_Divestment	Do_Not_Buy	Low	Wet	$55,000
165	90%_Divestment	Do_Not_Buy	Low	Soaking	$105,000
166	90%_Divestment	Do_Not_Buy	Fair	Dry	$-30,000
167	90%_Divestment	Do_Not_Buy	Fair	Wet	$40,000
168	90%_Divestment	Do_Not_Buy	Fair	Soaking	$90,000
169	90%_Divestment	Do_Not_Buy	High	Dry	$-50,000
170	90%_Divestment	Do_Not_Buy	High	Wet	$20,000
171	90%_Divestment	Do_Not_Buy	High	Soaking	$70,000
172	90%_Divestment	Buy_Insurance	Low	Dry	$-8,000
173	90%_Divestment	Buy_Insurance	Low	Wet	$42,000
174	90%_Divestment	Buy_Insurance	Low	Soaking	$92,000
175	90%_Divestment	Buy_Insurance	Fair	Dry	$-23,000
176	90%_Divestment	Buy_Insurance	Fair	Wet	$27,000
177	90%_Divestment	Buy_Insurance	Fair	Soaking	$77,000
178	90%_Divestment	Buy_Insurance	High	Dry	$-43,000
179	90%_Divestment	Buy_Insurance	High	Wet	$7,000
180	90%_Divestment	Buy_Insurance	High	Soaking	$57,000
181	100%_Divestment				$0

The optimal decisions for this problem can be identified by invoking PROC DTREE and using the SUMMARY statement as follows:

```
title "Oil Wildcatter's Problem";

proc dtree stagein=Dtoils4
           probin=Dtoilp4
           payoffs=Dtoilu4
           criterion=maxce rt=1200000
           nowarning;

   evaluate;
   summary / target=Divestment;
   summary / target=Insurance;
quit;
```

The optimal decision summaries in Output 3.2.2 and Output 3.2.3 show the optimal strategy for the wildcatter.

- The wildcatter should sell 30% of his investment to other companies and reject the insurance policy offered to him.
- The insurance policy should be accepted only if the decision to not divest is made.
- If the decision to buy the insurance policy is made, then it is optimal to divest 10% of the venture.

Output 3.2.2. Summary of the Oil Wildcatter's Problem for DIVESTMENT

```
                         Oil Wildcatter's Problem

                           The DTREE Procedure
                         Optimal Decision Summary

                             Order of Stages

                         Stage          Type
                         -----------------------
                         Divestment     Decision
                         Insurance      Decision
                         Cost           Chance
                         Oil_Deposit    Chance
                         _ENDST_        End


                           Decision Parameters

      Decision Criterion:    Maximize Certain Equivalent Value (MAXCE)
          Risk Tolerance:    $1,200,000
 Optimal Decision Yields:    $50,104


                         Optimal Decision Policy

                         Up to Stage Divestment

               Alternatives      Cumulative      Evaluating
                or Outcomes          Reward           Value
              ---------------------------------------------
              No_Divestment                         $45,728
              10%_Divestment                        $48,021
              20%_Divestment                        $49,907
              30%_Divestment                        $50,104*
              40%_Divestment                        $48,558
              50%_Divestment                        $45,219
              60%_Divestment                        $40,036
              70%_Divestment                        $32,965
              80%_Divestment                        $23,961
              90%_Divestment                        $12,985
              100%_Divestment                            $0
```

Output 3.2.3. Summary of the Oil Wildcatter's Problem for INSURANCE

```
                         Oil Wildcatter's Problem

                           The DTREE Procedure
                         Optimal Decision Summary

                             Order of Stages

                         Stage          Type
                         -----------------------
                         Divestment     Decision
                         Insurance      Decision
                         Cost           Chance
                         Oil_Deposit    Chance
                         _ENDST_        End


                           Decision Parameters

       Decision Criterion:    Maximize Certain Equivalent Value (MAXCE)
           Risk Tolerance:    $1,200,000
   Optimal Decision Yields:    $50,104


                         Optimal Decision Policy

                          Up to Stage Insurance

                                              Cumulative      Evaluating
        Alternatives or Outcomes                  Reward           Value
     ----------------------------------------------------------------
     No_Divestment      Buy_Insurance                           $45,728*
     No_Divestment      Do_Not_Buy                              $44,499
     10%_Divestment     Buy_Insurance                           $46,552
     10%_Divestment     Do_Not_Buy                              $48,021*
     20%_Divestment     Buy_Insurance                           $46,257
     20%_Divestment     Do_Not_Buy                              $49,907*
     30%_Divestment     Buy_Insurance                           $44,812
     30%_Divestment     Do_Not_Buy                              $50,104*
     40%_Divestment     Buy_Insurance                           $42,186
     40%_Divestment     Do_Not_Buy                              $48,558*
     50%_Divestment     Buy_Insurance                           $38,350
     50%_Divestment     Do_Not_Buy                              $45,219*
     60%_Divestment     Buy_Insurance                           $33,273
     60%_Divestment     Do_Not_Buy                              $40,036*
     70%_Divestment     Buy_Insurance                           $26,927
     70%_Divestment     Do_Not_Buy                              $32,965*
     80%_Divestment     Buy_Insurance                           $19,284
     80%_Divestment     Do_Not_Buy                              $23,961*
     90%_Divestment     Buy_Insurance                           $10,317
     90%_Divestment     Do_Not_Buy                              $12,985*
```

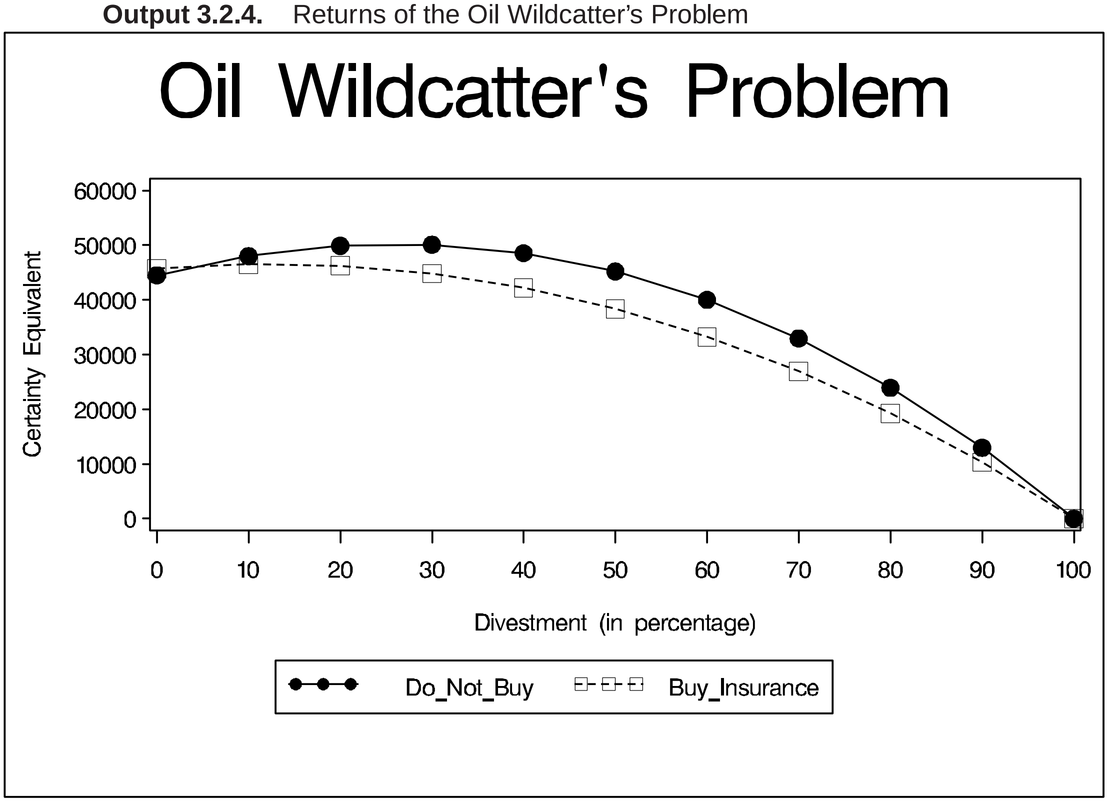
This information can be illustrated graphically using the GPLOT procedure. Output 3.2.4 on page 321, produced by the PROC GPLOT statements shown in the following code, provides a clear picture of the effects of the divestment possibilities and the insurance options.

```
   /*  create a data set for the return corresponds to each */
   /*  divestment possibilities and the insurance options   */
data Data2g;
   input  INSURE DIVEST VALUE;
   datalines;
       1       0   45728
       0       0   44499
       1      10   46552
       0      10   48021
       1      20   46257
       0      20   49907
       1      30   44812
       0      30   50104
       1      40   42186
       0      40   48558
       1      50   38350
       0      50   45219
       1      60   33273
       0      60   40036
       1      70   26927
       0      70   32965
       1      80   19284
       0      80   23961
       1      90   10317
       0      90   12985
       1     100       0
       0     100       0
;

   /* -- define a format for INSURE variable             -- */
proc format;
   value sample 0='Do_Not_Buy' 1='Buy_Insurance';
run;

   /* -- define title                                    -- */
title h=3 "Oil Wildcatter's Problem";

   /* -- set graphics options                            -- */
goptions lfactor=3;

   /* define legend                                      -- */
legend1 frame cframe=white label=none
        cborder=black position=center ;

   /* define symbol characteristics of the data points      */
   /* and the interpolation line for returns vs divestment  */
   /* when INSURE=0                                         */
symbol1 c=black i=join v=dot l=1 h=1.5;

   /* define symbol characteristics of the data points      */
   /* and the interpolation line for returns vs divestment  */
   /*  when INSURE=1                                        */
symbol2 c=black i=join v=square l=2 h=1.5;
```

```
   /* -- define axis characteristics                          -- */
axis1 minor=none label=('Divestment (in percentage)');
axis2 minor=none label=(angle=90 rotate=0 'Certainty Equivalent');

   /* plot VALUE vs DIVEST using INSURE as third variable   */
proc gplot data=Data2g ;
   plot VALUE*DIVEST=INSURE / haxis=axis1
                              vaxis=axis2
                              legend=legend1
                              name="dt2"
                              frame
                              cframe=white ;
   format INSURE SAMPLE.;
run;

quit;
```

Note that the data input into the Data2g data set is obtained from the optimal decision summary as in Output 3.2.3 on page 319. The value 1 of the INSURE variable represents the alternative '**Buy_Insurance**' and the value 0 represents the alternative '**Do_Not_Buy**'.

Output 3.2.4. Returns of the Oil Wildcatter's Problem



Example 3.3. Contract Bidding Problem

This example illustrates the use of several of the graphics options for producing graphics quality decision tree diagrams.

The production manager of a manufacturing company is planning to bid on a project to manufacture a new type of machine. He has the choice of bidding low or high. The evaluation of the bid will more likely be favorable if the bidder has built a prototype of the machine and includes it with the bid. However, he is uncertain about the cost of building the prototype. His technical staff has provided him a probability distribution on the cost of the prototype.

Table 3.33. Probability on the Cost of Building Prototype

Outcome	Cost	Probability
Expensive	$4,500	0.4
Moderate	$2,500	0.5
Inexpensive	$1,000	0.1

There is also uncertainty in whether he will win the contract or not. He has estimated the probability distribution of winning the contract as shown in Table 3.34.

Table 3.34. Probability of Winning the Contract

Givens		Events	
		Win the Contract	Lose the Contract
Build Prototype	High Bid	0.4	0.6
Build Prototype	Low Bid	0.8	0.2
No Prototype	High Bid	0.2	0.8
No Prototype	Low Bid	0.7	0.3

In addition, the payoffs of this bidding venture are affected by the cost of building the prototype. Table 3.35 shows his payoffs. The first row of the table shows the payoff is 0 if he loses the contract, regardless of whether or not he builds the prototype and whether he bids low or high. The remainder of the entries in the table give the payoff under the various scenarios.

Table 3.35. The Payoffs of the Contract Bidding Decision

States		Actions	
Result	Cost	Bid low	Bid high
Loss the Contract		0	0
Win the Contract		$35,000	$75,000
Win the Contract	Expensive	$25,000	$65,000
Win the Contract	Moderate	$35,000	$75,000
Win the Contract	Inexpensive	$45,000	$85,000

The production manager must decide whether to build the prototype and how to bid. He uses PROC DTREE to help him to make these decisions. The structure of the model is stored in the STAGEIN= data set named Stage3. There are two decision stages, '**Choose**' and '**Bid**', and two chance stages, '**Cost_Prototype**' and '**Contract**'. The '**Choose**' stage represents the decision whether or not to build a prototype. The chance stage '**Cost_Prototype**' represents the uncertain cost for building a prototype. It can be '**Expensive**', which costs $4,500, or '**Moderate**', which costs $2,500, or '**Inexpensive**', which costs $1,000. The '**Bid**' stage represents the decision whether to bid high or bid low. The last stage, '**Contract**', represents the result, either win the contract or lose the contract.

```
   /* -- create the STAGEIN= data set                  -- */
data Stage3;
   input _STNAME_ $16. _STTYPE_ $4. _OUTCOM_ $16.
         _SUCCES_ $16. _REWARD_ dollar8.0;
   datalines;
Choose          D   Build_Prototype Cost_Prototype       .
.               .   No_Prototype    Bid                  .
Cost_Prototype  C   Expensive       Bid             -$4,500
.               .   Moderate        Bid             -$2,500
.               .   Inexpensive     Bid             -$1,000
Bid             D   High_Bid        Contract             .
.               .   Low_Bid         Contract             .
Contract        C   Win_Contract    .                    .
.               .   Lose_Contract   .                    .
;
```

The PROBIN= data set, named Prob3, contains the probability information as in Table 3.33 (page 322) and Table 3.34 (page 322).

```
   /* -- create the PROBIN= data set                   -- */
data Prob3;
   input (_GIVEN1_ _GIVEN2_ _EVENT_) ($16.) _PROB_;
   datalines;
.               .               Expensive       0.4
.               .               Moderate        0.5
.               .               Inexpensive     0.1
Build_Prototype High_Bid        Win_Contract    0.4
Build_Prototype High_Bid        Lose_Contract   0.6
Build_Prototype Low_Bid         Win_Contract    0.8
Build_Prototype Low_Bid         Lose_Contract   0.2
No_Prototype    High_Bid        Win_Contract    0.2
No_Prototype    High_Bid        Lose_Contract   0.8
No_Prototype    Low_Bid         Win_Contract    0.7
No_Prototype    Low_Bid         Lose_Contract   0.3
;
```

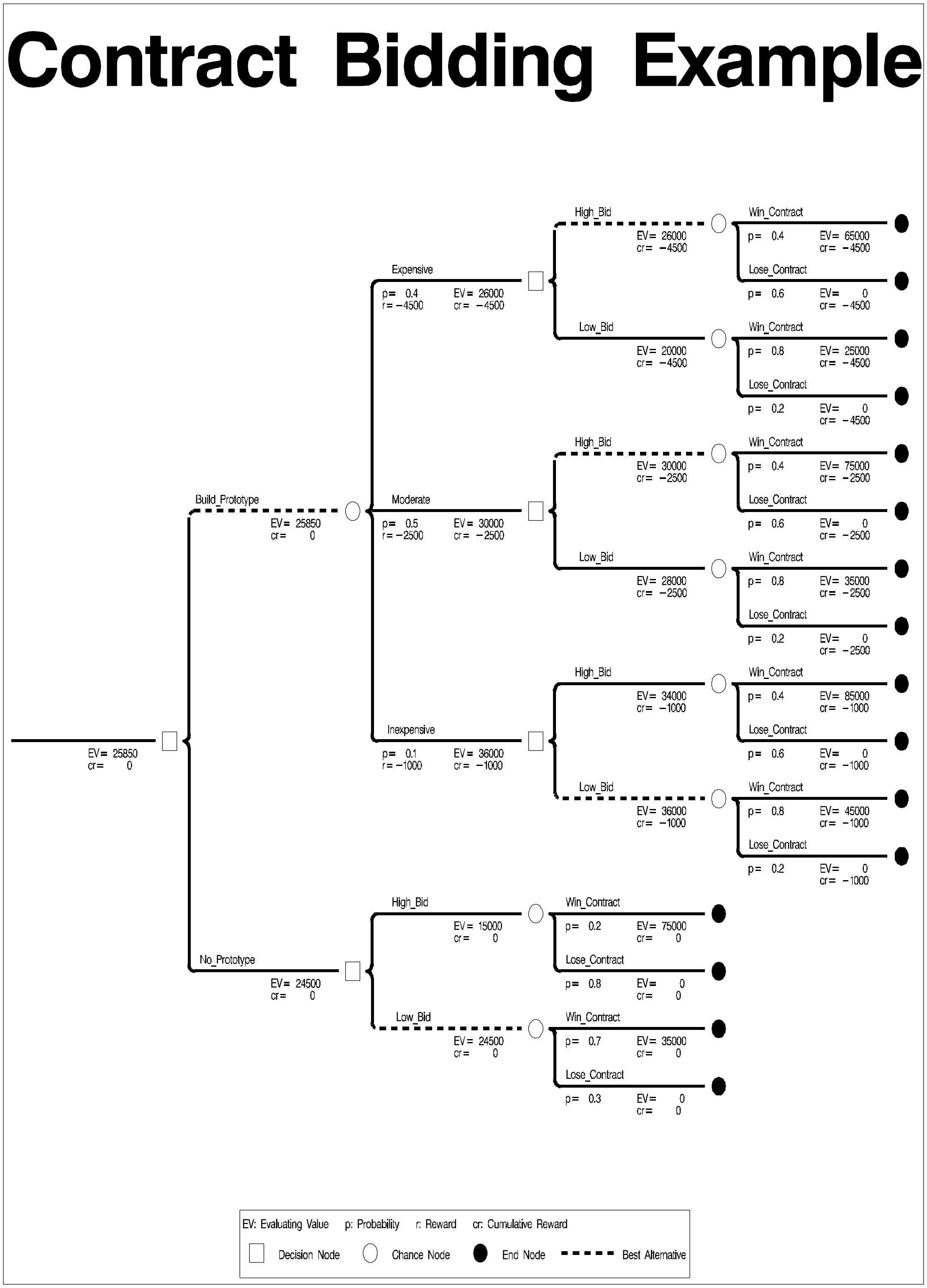
The PAYOFFS= data set named Payoff3 contains the payoff information as in Table 3.35 on page 322. Notice that the payoff to outcome '**Lose_Contract**' is not in the data set Payoff3. Since PROC DTREE assigns the default value 0 to all scenarios that are not in the PAYOFFS= data set, it is not necessary to include it.

```
   /* -- create the PAYOFFS= data set                  -- */
data Payoff3;
   input (_STATE1_ _STATE2_ _ACTION_) ($16.)
         _VALUE_  dollar8.0;
   datalines;
Win_Contract    .               Low_Bid         $35,000
Win_Contract    .               High_Bid        $75,000
Win_Contract    Expensive       Low_Bid         $25,000
Win_Contract    Expensive       High_Bid        $65,000
Win_Contract    Moderate        Low_Bid         $35,000
Win_Contract    Moderate        High_Bid        $75,000
Win_Contract    Inexpensive     Low_Bid         $45,000
Win_Contract    Inexpensive     High_Bid        $85,000
;
```

The solution, as in Output 3.3.1 on page 325, is displayed on a graphics device with the following code. Notice that the title is specified before invoking PROC DTREE. The GRAPHICS option is given on the PROC DTREE statement. Specifying the COMPRESS option in the TREEPLOT statement causes the decision tree diagram to be drawn completely on one page. The vertical distance between two successive end nodes is 1 character cell (`ybetween=1 cell`). All text, except that in the first title line is drawn with font SWISS and height 1 character cell. The height for all nodes is 3 character cells, which is specified by the HSYMBOL= option. The thickness for all links in the diagram, except those that represent optimal decisions, is specified by the LWIDTH= option as 20. The thickness of the links that represent optimal decisions is specified as 25 (lwidthb=25) , and the type of those links is 3 (`lstyleb=3`), the dash line. Colors for the text, links and nodes, and symbols to be used for nodes are not specified and hence defaults are used.

```
   /* -- define title                            -- */
title1 font=swissb h=2 "Contract Bidding Example" ;

   /* -- PROC DTREE statements                   -- */
proc dtree stagein=Stage3 probin=Prob3 payoffs=Payoff3
     graphics
     nowarning
     ;
   evaluate;
   treeplot / name="dt3" compress ybetween=1 cell
              ftext=swiss htext=1 hsymbol=3
              lstyleb=3 lwidth=20 lwidthb=25;
quit;
```

Output 3.3.1. Decision Tree for the Contract Bidding Problem



With the information on this decision tree, The production manager can select the optimal bidding strategy:

- He should build a prototype to accompany the bid and always bid high unless the cost for building the prototype is as low as $\$1,000$. This optimal strategy yields an expected return of $\$25,850$.
- If no prototype is built, the preferred decision is to make a low bid. In this case the expected return is $\$24,500$.

Example 3.4. Research and Development Decision Problem

This example illustrates the use of the SYMBOL and GOPTIONS statements for controlling the appearance of the decision tree diagram. It also uses the ANNOTATE= option to add a customized legend to the diagram.

A typical problem encountered in a research and development setting involves two decisions: whether or not to conduct research, and whether or not to commercialize the results of that research. Suppose that research and development for a specific project will cost $\$350,000$, and there is a 0.4 probability that it will fail. Also suppose that the different levels of market success and their corresponding probabilities are:

Table 3.36. Levels of Market Success and Their Probabilities

Market Success	**Net Return**	**Probability**
Great	$\$1,000,000$	0.25
Good	$\$500,000$	0.35
Fair	$\$200,000$	0.30
Poor	-$\$250,000$	0.10

The structure of the model is represented in the STAGEIN= data set Stage4.

```
   /* -- create the STAGEIN= data set              -- */
data Stage4;
   input _STNAME_ $ 1-16 _STTYPE_ $ 17-20
         _OUTCOM_ $ 21-32 _REWARD_ dollar12.0
         _SUCC_ $ 45-60;
   datalines;
R_and_D         D   Not_Conduct .           .
.               .   Conduct     -$350,000   RD_Outcome
RD_Outcome      C   Success     .           Production
.               .   Failure     .           .
Production      D   Produce     .           Sales
.               .   Abandon     .           .
Sales           C   Great       .           .
.               .   Good        .           .
.               .   Fair        .           .
.               .   Poor        .           .
;
```

The probability distributions for the various outcomes of the chance stages are given in the PROBIN= data set named Prob4.

```
   /* -- create the PROBIN= data set               -- */
data Prob4;
   input _EVENT1_ $12. _PROB1_ _EVENT2_ $12. _PROB2_;
   datalines;
Success     0.6     Failure   0.4
Great       0.25    Good      0.35
Fair        0.30    poor      0.1
;
```

The payoffs are given in the PAYOFFS= data set Payoff4.

```
 /* -- create the PAYOFFS= data set                     -- */
data Payoff4;
   input _STATE_ $12. _VALUE_ dollar12.0;
   datalines;
Great       $1,000,000
Good        $500,000
Fair        $200,000
Poor        -$250,000
;
```

The following DATA step builds a data set that contains the Annotate description of a legend. Refer to the chapter on the annotate facility in *SAS/GRAPH Software: Reference* for a description of the Annotate facility.

```
   /* -- create the ANNOTATE= data set for legend     -- */
data Legends;
   length FUNCTION STYLE $ 8;
   WHEN = 'B';  POSITION='0';
   XSYS='4';  YSYS='4';
   input FUNCTION $ X Y STYLE $ SIZE COLOR $ TEXT $ & 16.;
   datalines;
move       8   2.1  .        .   .       .
draw      12   2.1  .        8   black   .
label     14   2    swiss    0.7 black   BEST ACTION
symbol     9   3.5  marker   0.7 black   A
label     14   3.2  swiss    0.7 black   END NODE
symbol     9   4.7  marker   0.7 black   P
label     14   4.4  swiss    0.7 black   CHANCE NODE
symbol     9   5.9  marker   0.7 black   U
label     14   5.6  swiss    0.7 black   DECISION NODE
label      8   7.0  swiss    0.7 black   LEGEND:
move       5   8.5  .        .   black   .
draw      27   8.5  .        2   black   .
draw      27   1    .        2   black   .
draw       5   1    .        2   black   .
draw       5   8.5  .        2   black   .
;
```

The following program invokes PROC DTREE, which evaluates the decision tree and plots it on a graphics device using the Annotate data set Legends to draw the legend.

```
   /* define symbol characteristics for chance nodes and  */
   /* links except those that represent optimal decisions */
symbol1 f=marker h=2 v=P c=black w=5 l=1;

   /* define symbol characteristics for decision nodes    */
   /* and links that represent optimal decisions          */
symbol2 f=marker h=2 v=U c=black w=10 l=1;

   /* define symbol characteristics for end nodes         */
symbol3 f=marker h=2 v=A c=black;

   /* -- define title                                  -- */
title f=swissb h=2 'Research and Development Decision';

   /* -- PROC DTREE statements                         -- */
proc dtree stagein=Stage4 probin=Prob4 payoffs=Payoff4
           criterion=maxce rt=1800000
           graphics annotate=Legends nolegend;

   evaluate;

   treeplot / linka=1 linkb=2 symbold=2 symbolc=1 symbole=3
              compress name="dt4";

quit;
```

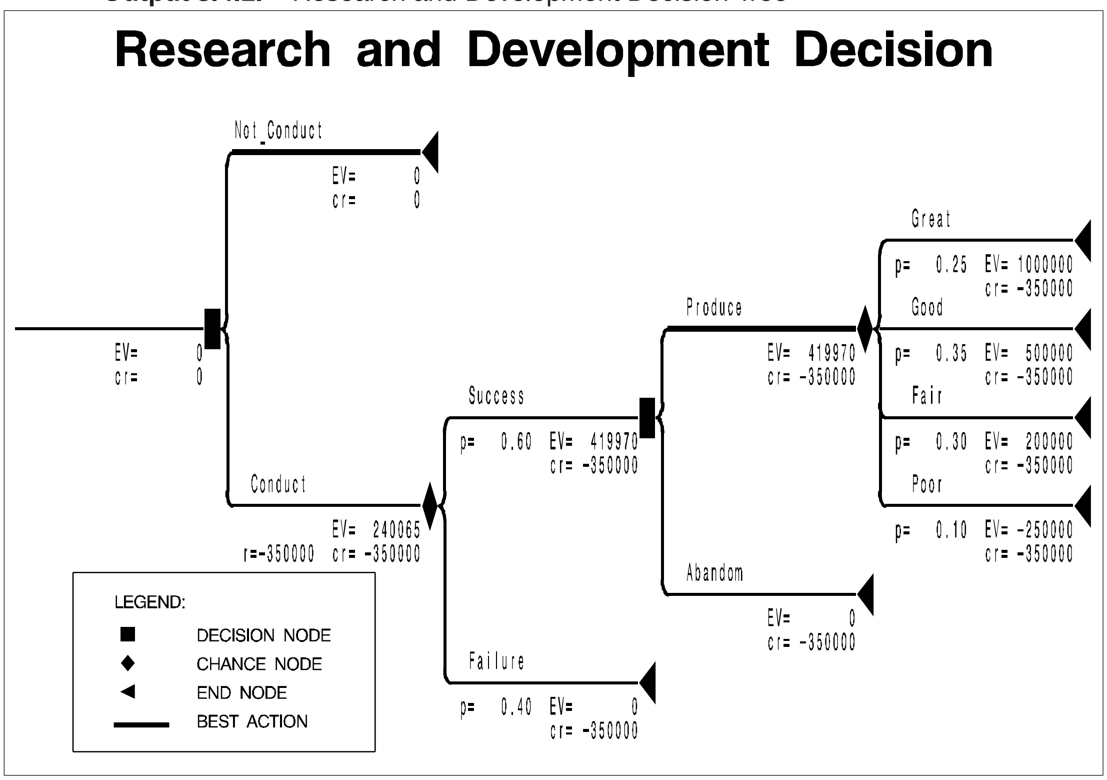
The SYMBOL1, SYMBOL2, and SYMBOL3 statements create three SYMBOL definitions that contain information for drawing nodes and links. The Legends data set and the ANNOTATE= option specified in the PROC DTREE statement causes the procedure to produce a customized legend for the decision tree diagram. The LINKA=, LINKB=, SYMBOLD=, SYMBOLC=, and SYMBOLE= specifications in the TREEPLOT statement tell PROC DTREE how to use SYMBOL definitions to draw decision tree. Table 3.37 on page 329 describes the options in SYMBOL definitions used to draw the decision tree diagram.

The decision tree diagram produced by the TREEPLOT statement is shown in Output 3.4.1 (page 329). As illustrated on the decision tree, the program recommends that one should not conduct the research and development of the product if he or she is risk averse with a risk tolerance of $\$1,800,000$. However, if he or she decides to undertake the research and development and it is a success, then he or she should commercialize the product.

Table 3.37. The Usage of SYMBOL Definitions

SYMBOL Definition	Specification	Description	Used to Draw
The First	C=*black* L=1 W=5	Color Line Type Thickness	All links except those that indicate optimal decisions
	C=*black* F=*marker* H=2 V=*P*	Color Font Height Symbol	Chance nodes
The Second	C=*black* L=1 W=10	Color Line Type Thickness	All links that indicate optimal decisions
	C=*black* F=*marker* H=2 V=*U*	Color Font Height Symbol	Decision nodes
The Third	C=*black* F=*marker* H=2 V=*A*	Color Font Height Symbol	End nodes

Output 3.4.1. Research and Development Decision Tree



Example 3.5. Loan Grant Decision Problem

Many financial decisions are difficult to analyze because of the variety of available strategies and the continuous nature of the problem. However, if the alternatives and time frame can be restricted, then decision analysis can be a useful analysis tool.

For example, a loan officer is faced with the problem of deciding whether to *approve* or *deny* an application for a one-year $30,000 loan at the current rate of 15% of interest. If the application is approved, the lender will either *pay off* the loan in full after one year or *default*. Based on experience, the default rate is about 36 out of 700. If the loan is denied, the money is put in government bonds at the interest rate of 8%.

To obtain more information about the lender, the loan officer engages a credit investigation unit at a cost of $500 per person that will give either a *positive* recommendation for making a loan or a *negative* recommendation. Past experience with this investigator yields that of those who ultimately paid off their loans, 570 out of 664 were given a positive recommendation. On the other hand, 6 out of 26 that had defaulted had also been given a positive recommendation by the investigator.

The STAGEIN= data set, Stage6, gives the structure of the decision problem.

```
   /* -- create the STAGEIN= data set                  -- */
data Stage6;
   input _STNAME_ $ 1-16 _STTYPE_ $ 17-20
         _OUTCOM_ $ 21-44  _SUCC_ $ 45-60;
   datalines;
Application     D   Approve loan            Payment
.               .   Deny loan               .
Payment         C   Pay off                 .
.               .   Default                 .
Investigation   D   Order investigation     Recommendation
.               .   Do not order            Application
Recommendation  C   Positive                Application
.               .   Negative                Application
;
```

The PROBIN= data set Prob6 gives the probability distributions for the random events at the chance nodes.

```
   /* -- create the PROBIN= data set                   -- */
data Prob6;
   length _GIVEN_ _EVENT1_ _EVENT2_ $16;

   _EVENT1_='Pay off';   _EVENT2_='Default';
   _PROB1_=664/700;      _PROB2_=1.0-_PROB1_;
   output;

   _GIVEN_='Pay off';
   _EVENT1_='Positive';   _EVENT2_='Negative';
   _PROB1_=570/664;       _PROB2_=1.0-_PROB1_;
   output;
```

```
   _GIVEN_='Default';
   _EVENT1_='Positive';   _EVENT2_='Negative';
   _PROB1_=6/26;          _PROB2_=1.0-_PROB1_;
   output;

run;
```

The PAYOFFS= data set Payoff6 gives the payoffs for the various scenarios. Notice that the first observation in this data set indicates that if the officer denies the loan application, then payoffs are the interest from the money invested in government bonds. The second and the third observations are redundant for the basic analysis but are needed to determine the value of information as shown later.

```
   /* -- create the PAYOFFS= data set                  -- */
data Payoff6(drop=loan);
   length _STATE_ _ACT_ $24;

   loan=30000;

   _ACT_='Deny loan';   _VALUE_=loan*0.08;   output;
   _STATE_='Pay off';   _VALUE_=loan*0.08;   output;
   _STATE_='Default';   _VALUE_=loan*0.08;   output;

   _ACT_='Approve loan';
   _STATE_='Pay off';   _VALUE_=loan*0.15;   output;
   _STATE_='Default';   _VALUE_=-1.0*loan;   output;

run;
```

The following code invokes the DTREE procedure to solve this decision problem.

```
   /* -- define title                                  -- */
title 'Loan Grant Decision';

   /* -- PROC DTREE statements                         -- */
proc dtree
     stagein=Stage6 probin=Prob6 payoffs=Payoff6
     summary target=investigation nowarning;

   modify 'Order investigation' reward -500;

   evaluate;

   OPTIONS LINESIZE=85;
   summary / target=Application;
   OPTIONS LINESIZE=80;
```

Note that the $500 investigation fee is not included in the Stage6 data set. Since the outcome '**`Order investigation`**' is the only outcome that has a nonzero reward, it is easier to set the reward for this outcome using the MODIFY statement. The quotes that enclose the outcome name in the MODIFY statement are necessary because the outcome name contains a space.

The results in Output 3.5.1 and Output 3.5.2 indicate that it is optimal to do the following:

- The loan officer should order the credit investigation and approve the loan application if the investigator gives the applicant a positive recommendation. On the other hand, he should the application if a negative recommendation is given to the applicant.
- Furthermore, the loan officer should order a credit investigation if the cost for the investigation is less than $\$3,725 - \$2,726 = \$999$.

Output 3.5.1. Summary of the Loan Grant Decision for Investigation

```
                         Loan Grant Decision

                        The DTREE Procedure
                      Optimal Decision Summary

                          Order of Stages

                     Stage             Type
                     --------------------------
                     Investigation     Decision
                     Recommendation    Chance
                     Application       Decision
                     Payment           Chance
                     _ENDST_           End


                        Decision Parameters

      Decision Criterion:    Maximize Expected Value (MAXEV)
 Optimal Decision Yields:    3225


                      Optimal Decision Policy

                     Up to Stage Investigation

             Alternatives        Cumulative    Evaluating
              or Outcomes            Reward         Value
          -----------------------------------------------
          Order investigation          -500         3725*
          Do not order                    0         2726
```

Output 3.5.2. Summary of the Loan Grant Decision for Application

```
                               Loan Grant Decision

                               The DTREE Procedure
                            Optimal Decision Summary

                                Order of Stages

                           Stage            Type
                           --------------------------
                           Investigation    Decision
                           Recommendation   Chance
                           Application      Decision
                           Payment          Chance
                           _ENDST_          End


                               Decision Parameters

              Decision Criterion:    Maximize Expected Value (MAXEV)
         Optimal Decision Yields:    3225


                             Optimal Decision Policy

                             Up to Stage Application

                                                              Cumulative  Evaluating
                  Alternatives or Outcomes                        Reward       Value
------------------------------------------------------------------------------------
Order investigation  Positive            Approve loan               -500       4004*
Order investigation  Positive            Deny loan                  -500       2400
Order investigation  Negative            Approve loan               -500      -3351
Order investigation  Negative            Deny loan                  -500       2400*
Do not order                             Approve loan                  0       2726*
Do not order                             Deny loan                     0       2400
```

Now, the loan officer learns of another credit investigation company that claims to have a more accurate credit checking system for predicting whether the applicants will default on their loans. However, he has not been able to find out what the company charges for their service or how accurate their credit checking system is. Perhaps the best thing he can do at this stage is to assume that the company can predict perfectly whether or not applicants will default on their loans and determine the maximum amount to pay for this perfect investigation. The answer to this question can be found with the PROC DTREE statements:

```
save;
move payment before investigation;
evaluate;
recall;
```

Notice that moving the stage '**`Payment`**' to the beginning of the tree means that the new decision tree contains two scenarios that are not in the original tree: the scenario '**`Pay off`**' and '**`Deny loan`**', and the scenario '**`Default`**' and '**`Deny loan`**'. The second and third observations in the Payoff6 data set supply values for these new scenarios. If these records are not included in the PAYOFFS= data set, then PROC DTREE assumes they are 0.

Also notice that the SUMMARY and TARGET= options are specified globally in the PROC DTREE statement and hence are not needed in the EVALUATE statement. The results from the DTREE procedure are displayed in Output 3.5.3.

Output 3.5.3. Summary of the Loan Grant Decision with Perfect Information

```
                         Loan Grant Decision

                         The DTREE Procedure
                       Optimal Decision Summary

                           Order of Stages

                      Stage             Type
                      --------------------------
                      Payment           Chance
                      Investigation     Decision
                      Recommendation    Chance
                      Application       Decision
                      _ENDST_           End


                         Decision Parameters

            Decision Criterion:    Maximize Expected Value (MAXEV)
       Optimal Decision Yields:    4392


                       Optimal Decision Policy

                      Up to Stage Investigation

                                               Cumulative    Evaluating
         Alternatives or Outcomes                  Reward         Value
     ----------------------------------------------------------------------
     Pay off                Order investigation      -500          4500
     Pay off                Do not order                0          4500*
     Default                Order investigation      -500          2400
     Default                Do not order                0          2400*
```

The optimal decision summary in Output 3.5.3 shows that the yields with perfect investigation is $\$4,329$. Recall that the yields of alternative '**`Do not order`**' the investigation, as shown in Output 3.5.1 on page 332, is $\$2,726$. Therefore, the maximum amount he should pay for the perfect investigation can be determined easily as

$$\begin{aligned} \text{VPI} &= \text{Value with Perfect Investigation} - \text{Value without Investigation} \\ &= \$4,392 - \$2,726 \\ &= \$1,666 \end{aligned}$$

Note that if you use the VPI statement to determine the value of a perfect investigation, the result is different from the value calculated previously.

```
vpi payment;
```

```
NOTE: The currently optimal decision yields 3225.4725275.
NOTE: The new optimal decision yields 4392.
```

```
NOTE: The value of perfect information of stage Payment
      yields 1166.5274725.
```

The reason for this difference is that the VPI statement causes PROC DTREE first to determine the value with perfect information, then to compare this value with the value with current information available (in this example, it is the recommendation from the original investigation unit). Therefore, the VPI statement returns a value that is calculated as

$$\begin{aligned} \text{VPI} &= \text{Value with Perfect Information} - \text{Value with Current Information} \\ &= \$4,392 - \$3,225 \\ &= \$1,167 \end{aligned}$$

The loan officer considered another question regarding the maximum amount he should pay to a company to help collect the principal and the interest if an applicant defaults on the loan. This question is similar to the question concerning the improvement that can be expected if he can control whether or not an applicant will default on his loan (of course he will always want the lender to pay off in full after one year). The answer to this question can be obtained with the following statements:

```
modify payment type;
evaluate;
```

Output 3.5.4. Summary of the Loan Grant Decision with Perfect Control

```
                           Loan Grant Decision

                           The DTREE Procedure
                         Optimal Decision Summary

                             Order of Stages

                        Stage             Type
                        -------------------------
                        Investigation     Decision
                        Recommendation    Chance
                        Application       Decision
                        Payment           Decision
                        _ENDST_           End


                           Decision Parameters

            Decision Criterion:    Maximize Expected Value (MAXEV)
       Optimal Decision Yields:    4500


                         Optimal Decision Policy

                        Up to Stage Investigation

               Alternatives        Cumulative    Evaluating
                or Outcomes            Reward         Value
             -----------------------------------------------
             Order investigation         -500          4500
             Do not order                   0          4500*
```

The result is obvious and is shown in Output 3.5.4. Using a calculation similar to the one used to calculate the value of a perfect investigation, the maximum amount one should pay for this kind of service is $\$4,500 - \$2,726 = \$1,774$. As previously described, this value is different from the value obtained by using the VPC statement. In fact, if you specify the statement

```
vpc payment;
```

you get the value of VPC, which is $\$1,274.53$, from the SAS log as

```
NOTE: The currently optimal decision yields 3225.4725275.
NOTE: The new optimal decision yields 4500.
NOTE: The value of perfect control of stage Payment yields
      1274.5274725.
```

Obviously, all of the values of investigation and other services depend on the value of the loan. Since each of the payoffs for the various scenarios given in the Payoff6 data set is proportional to the value of loan, you can safely assume that the value of the loan is 1 unit and determine the ratio of the value for a particular service to the value of the loan. To obtain these ratios, change the value of the variable LOAN to 1 in the Payoff6 data set and invoke PROC DTREE again as follows:

```
   /* -- create the alternative PAYOFFS= data set  -- */
data Payoff6a(drop=loan);
```

```
   length _STATE_ _ACT_ $24;
   loan=1;

   _ACT_='Deny loan';   _VALUE_=loan*0.08;   output;
   _STATE_='Pay off';   _VALUE_=loan*0.08;   output;
   _STATE_='Default';   _VALUE_=loan*0.08;   output;

   _ACT_='Approve loan';
   _STATE_='Pay off';   _VALUE_=loan*0.15;   output;
   _STATE_='Default';   _VALUE_=-1.0*loan;   output;
run;

   /* -- PROC DTREE statements                     -- */
title 'Loan Grant Decision';

proc dtree
     stagein=Stage6 probin=Prob6 payoffs=Payoff6a
     nowarning;

   evaluate / summary target=investigation;

   save;
   move payment before investigation;
   evaluate;


   recall;
   modify payment type;
   evaluate;

quit;
```

The optimal decision summary given in Output 3.5.5 shows that the ratio of the value of investigation that the loan officer currently engages in to the value of the loan is $0.1242 - 0.0909 = 0.0333$ to 1.

Output 3.5.5. Summary of the Loan Grant Decision with 1 Unit Loan

```
                          Loan Grant Decision

                          The DTREE Procedure
                        Optimal Decision Summary

                            Order of Stages

                       Stage            Type
                       -------------------------
                       Investigation    Decision
                       Recommendation   Chance
                       Application      Decision
                       Payment          Chance
                       _ENDST_          End


                          Decision Parameters

          Decision Criterion:    Maximize Expected Value (MAXEV)
     Optimal Decision Yields:    0.1242


                        Optimal Decision Policy

                       Up to Stage Investigation

              Alternatives        Cumulative    Evaluating
               or Outcomes            Reward         Value
            -----------------------------------------------
            Order investigation                    0.1242*
            Do not order                           0.0909
```

The following messages are written to the SAS log:

```
NOTE: Present order of stages:

      Investigation(D), Recommendation(C), Application(D),
      Payment(C), _ENDST_(E).

NOTE: The current problem has been successfully saved.

NOTE: Present order of stages:

      Payment(C), Investigation(D), Recommendation(C),
      Application(D), _ENDST_(E).

NOTE: The currently optimal decision yields 0.1464.

NOTE: The original problem has been successfully recalled.

NOTE: Present order of stages:

      Investigation(D), Recommendation(C), Application(D),
      Payment(C), _ENDST_(E).

NOTE: The type of stage Payment has been changed.

NOTE: The currently optimal decision yields 0.15.
```

The preceding messages show that the ratio of the value of perfect investigation to the value of a loan is $0.1464 - 0.0909 = 0.0555$ to 1, and the ratio of the maximum amount the officer should pay for perfect control to the value of loan is $0.15 - 0.0909 = 0.591$ to 1.

Output 3.5.6 on page 340, produced by the following statements, shows a table of the values of the investigation currently engaged in, the values of perfect investigation, and the values of perfect control for loans ranging from $\$10,000$ to $\$100,000$.

```
   /* create the data set for value of loan   */
   /* and corresponding values of services    */
data Datav6(drop=k ratio1 ratio2 ratio3);
   label loan="Value of Loan"
         vci="Value of Current Credit Investigation"
         vpi="Value of Perfect Credit Investigation"
         vpc="Value of Perfect Collecting Service";

      /* calculate ratios */
   ratio1=0.1242-0.0909;
   ratio2=0.1464-0.0909;
   ratio3=0.15-0.0909;

   Loan=0;
   do k=1 to 10;

         /* set the value of loan */
      loan=loan+10000;

         /* calculate the values of various services */
      vci=loan*ratio1;
      vpi=loan*ratio2;
      vpc=loan*ratio3;

         /* output current observation */
      output;
   end;
run;

   /* print the table of the value of loan   */
   /* and corresponding values of services   */
title 'Value of Services via Value of Loan';

proc print label;
   format loan vci vpi vpc dollar12.0;
run;
```

Output 3.5.6. Values of Loan and Associated Values of Service

Value of Services via Value of Loan

Obs	Value of Loan	Value of Current Credit Investigation	Value of Perfect Credit Investigation	Value of Perfect Collecting Service
1	$10,000	$333	$555	$591
2	$20,000	$666	$1,110	$1,182
3	$30,000	$999	$1,665	$1,773
4	$40,000	$1,332	$2,220	$2,364
5	$50,000	$1,665	$2,775	$2,955
6	$60,000	$1,998	$3,330	$3,546
7	$70,000	$2,331	$3,885	$4,137
8	$80,000	$2,664	$4,440	$4,728
9	$90,000	$2,997	$4,995	$5,319
10	$100,000	$3,330	$5,550	$5,910

Example 3.6. Petroleum Distributor's Decision Problem

The president of a petroleum distribution company currently faces a serious problem. His company supplies refined products to its customers under long-term contracts at guaranteed prices. Recently, the price for petroleum has risen substantially and his company will lose $\$450,000$ this year because of its long-term contract with a particular customer. After a great deal of discussion with his legal advisers and his marketing staff, the president learns that the contract contains a clause that may be beneficial to his company. The clause states that when circumstances are beyond its control, the company may ask its customers to pay the prevailing market prices for up to 10% of the promised amount.

Several scenarios are possible if the clause is invoked. If the customer accepts the invocation of the clause and agrees to pay the higher price for the 10%, the company would turn a loss of $\$450,000$ into a net profit of $\$600,000$. If the customer does not accept the invocation, the customer may sue for damages or accept a settlement of $\$900,000$ (resulting in a loss of $\$400,000$) or simply decline to press the issue. In any case, the distribution company could then sell the 10% on the open market for an expected value of $\$500,000$. However, the lawsuit would result in one of three possible outcomes: the company wins and pays no damages; the company loses and pays normal damages of $\$1,500,000$; the company loses and pays double damages of $\$3,000,000$. The lawyers also feel that this case might last three to five years if the customer decides to sue the company. The cost of the legal proceedings is estimated as $\$30,000$ for the initial fee and $\$20,000$ per year. The likelihood of the various outcomes are also assessed and reported as in Table 3.38. Suppose that the company decides to use a discount rate of 10% to determine the present value of future funds.

Table 3.38. Likelihood of the Outcomes in the Petroleum Distributor's Decision

Uncertainty	Outcome	Probability
Customer's Response	Accept the Invocation	0.1
	Reject the Invocation	0.9
Customer's Action if the Invocation is being Rejected	Press the Issue	0.1
	Settle the Case	0.45
	Sue for Damages	0.45
Case Last	3 Years	0.3
	4 Years	0.4
	5 Years	0.3
Lawsuit Result	Pay No Damages	0.15
	Pay Normal Damages	0.65
	Pay Double Damages	0.2

The structure for this decision problem is given in the STAGEIN= data set named Stage7.

```
   /* -- create the STAGEIN= data set                      -- */
data Stage7;
   input _STNAME_ $ 1-10  _STTYPE_ $ 11-14 _OUTCOM1 $ 15-30
         _SUCC1   $ 30-40 _OUTCOM2 $ 41-56 _SUCC2   $ 57-66;
   datalines;
Action    D   Invoking        Response  Not_Invoking    .
Response  C   Accept          .         Refuse          Lawsuit
Lawsuit   C   Press_Issue     .         Settle          .
              Sue             Last
Last      C   3_Years         Result    4_Years         Result
              5_Years         Result
Result    C   No_Damages      .         Normal_Damages  .
              Double_Damages  .
;
```

The PROBIN= data set Prob7 contains the probability distributions for the chance nodes.

```
   /* -- create the PROBIN= data set                       -- */
data Prob7;
   input _EVENT1_ $ 1-16  _PROB1_ 17-24
         _EVENT2_ $ 25-40 _PROB2_ 41-48;
   datalines;
Accept          0.1     Refuse          0.9
Press_Issue     0.1     Settle          0.45
Sue             0.45
3_Years         0.3     4_Years         0.4
5_Years         0.3
No_Damages      0.15    Normal_Damages  0.65
Double_Damages  0.20
;
```

The PAYOFFS= data set Payoff7 defines the payoffs for the various scenarios.

```
   /* -- create the PAYOFFS= data set            -- */
data Payoff7(drop=i j k D PCOST);
   length _ACTION_ _STATE1-_STATE4 $16;

      /* possible outcomes for the case last        */
   array YEARS{3}   $16. _TEMPORARY_ ('3_Years',
                                      '4_Years',
                                      '5_Years' );

      /* numerical values for the case last  */
   array Y{3}            _TEMPORARY_ (3, 4, 5);

      /* possible outcomes for the size of judgment */
   array DAMAGES{3} $16. _TEMPORARY_ ('No_Damages',
                                      'Normal_Damages',
                                      'Double_Damages' );

      /* numerical values for the size of judgment  */
   array C{3}            _TEMPORARY_ (0, 1500, 3000);

   D=0.1;                         /* discount rate */

      /* payoff for the scenario which the          */
      /* 10 percent clause is not invoked           */
   _ACTION_='Not_Invoking';   _VALUE_=-450;   output;

      /* the clause is invoked */
   _ACTION_='Invoking';

      /* payoffs for scenarios which the clause is  */
      /* invoked and the customer accepts the       */
      /* invocation                                 */
   _STATE1='Accept';           _VALUE_=600;   output;

      /* the customer refuses the invocation        */
   _STATE1='Refuse';

      /* payoffs for scenarios which the clause is  */
      /* invoked and the customer refuses the       */
      /* invocation but decline to press the issue  */
   _STATE2='Press_Issue';   _VALUE_=500;      output;

      /* payoffs for scenarios which the clause is  */
      /* invoked and the customer refuses the       */
      /* invocation but willing to settle out of    */
      /* court for 900K                             */
   _STATE2='Settle';       _VALUE_=500-900;   output;
```

```
         /* the customer will sue for damages          */
      _STATE2='Sue';
      do i=1 to 3;
         _STATE3=YEARS{i};

            /* determine the cost of proceedings       */
         PCOST=30;  /* initial cost of the proceedings */

            /* additional cost for every years in      */
            /* in present value                        */
         do k=1 to Y{i};
            PCOST=PCOST+(20/((1+D)**k));
         end;

            /* loop for all poss. of the lawsuit result */
         do j=1 to 3;
            _STATE4=DAMAGES{j}; /* the damage have to paid */

               /* compute the net return in present value  */
            _VALUE_=500-PCOST-(C{j}/((1+D)**Y{i}));

               /* output an observation for the payoffs */
               /* of this scenario                      */
            output;
         end;
      end;

   run;

      /* -- print the payoff table                         -- */
   title "Petroleum Distributor's Decision";
   title3 "Payoff table";

   proc print;
   run;
```

Output 3.6.1. Payoffs for the Petroleum Distributor's Problem

Petroleum Distributor's Decision

Payoff table

Obs	_ACTION_	_STATE1	_STATE2	_STATE3	_STATE4	_VALUE_
1	Not_Invoking					-450.00
2	Invoking	Accept				600.00
3	Invoking	Refuse	Press_Issue			500.00
4	Invoking	Refuse	Settle			-400.00
5	Invoking	Refuse	Sue	3_Years	No_Damages	420.26
6	Invoking	Refuse	Sue	3_Years	Normal_Damages	-706.71
7	Invoking	Refuse	Sue	3_Years	Double_Damages	-1833.68
8	Invoking	Refuse	Sue	4_Years	No_Damages	406.60
9	Invoking	Refuse	Sue	4_Years	Normal_Damages	-617.92
10	Invoking	Refuse	Sue	4_Years	Double_Damages	-1642.44
11	Invoking	Refuse	Sue	5_Years	No_Damages	394.18
12	Invoking	Refuse	Sue	5_Years	Normal_Damages	-537.20
13	Invoking	Refuse	Sue	5_Years	Double_Damages	-1468.58

The payoff table of this problem is displayed in Output 3.6.1. Note that the payoffs of the various scenarios are in thousands of dollars and are *net present values* (NPV) (Baird 1989). For example, the payoff for the following scenario “invoking the clause; the customer refuses to accept this and sues for damages; the case lasts 4 years and the petroleum distribution company loses and pays double damages” is calculated as

$$\begin{aligned}\text{Payoff} &= 500 - \text{NPV of proceedings cost} \\ &\qquad - \text{NPV of damages of 3,000,000} \\ &= -1642.44\end{aligned}$$

where

$$\text{NPV of proceedings cost} = 30 + \sum_{k=1}^{4} 20/(1+0.1)^k$$

and

$$\text{NPV of damages of 3,000,000} = 3000/(1+0.1)^4$$

This is because the company can sell the 10% for $\$500,000$ immediately and pay the $\$3,000,000$ damages four years from now. The net present value of the proceedings is determined by paying the $\$30,000$ initial fee now and a fee of $\$20,000$ after every year up to four years. The value of 0.1 is the discount rate used.

The following statements evaluate the problem and plot the optimal solution.

```
   /* -- define colors list                                -- */
goptions colors=(black);

   /* -- define title                                      -- */
title f=zapfb h=2.5 "Petroleum Distributor’s Decision";

   /* -- PROC DTREE statements                             -- */
proc dtree stagein=Stage7 probin=Prob7 payoffs=Payoff7;
   evaluate / summary;
   treeplot / graphics compress nolg name="dt6p1"
              ybetween=1 cell lwidth=8 lwidthb=20 hsymbol=3;

quit;
```

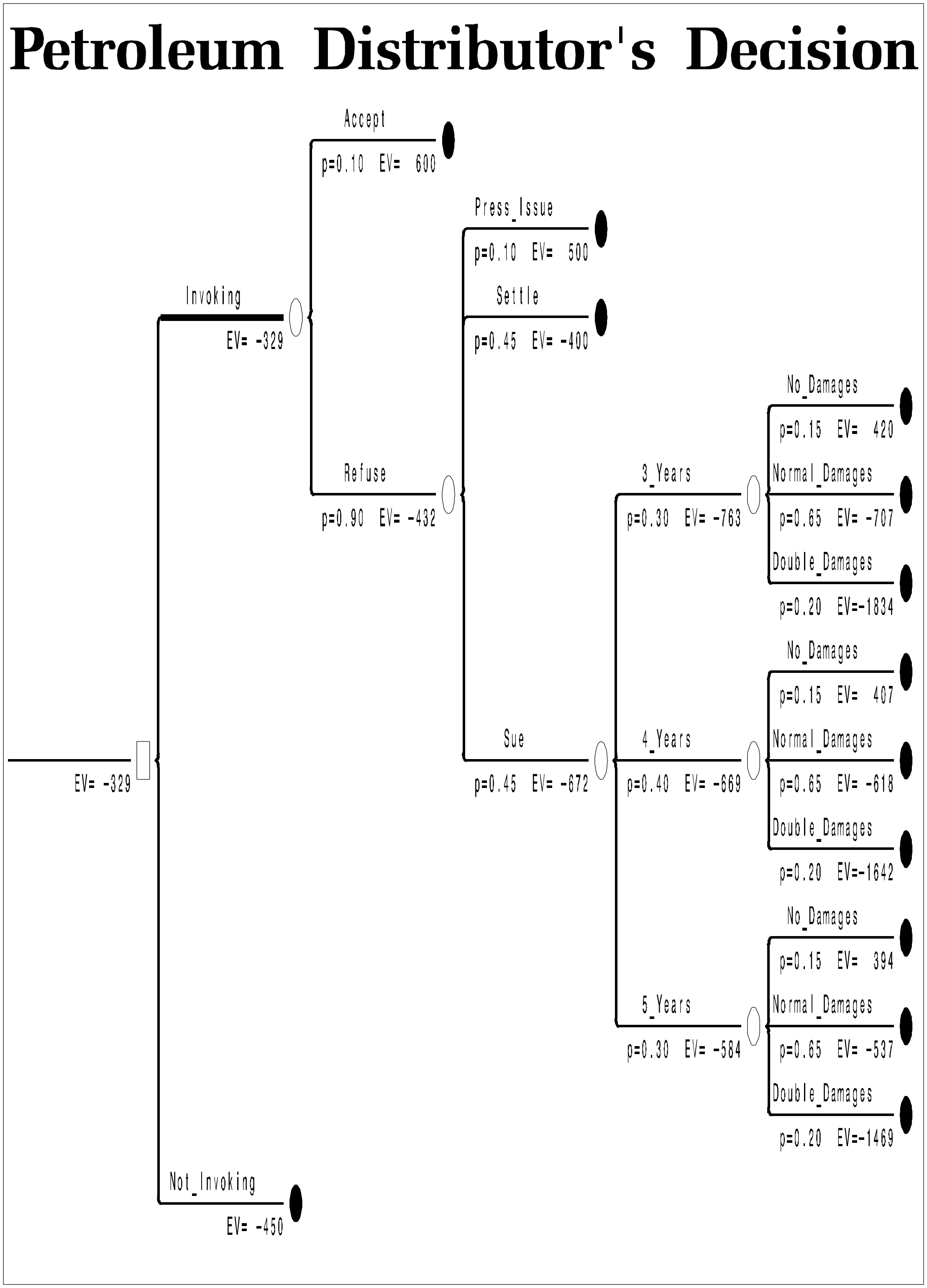
The optimal decision summary in Output 3.6.2 suggests that the president should invoke the 10% clause because it would turn a loss of $450,000 into an expected loss of $329,000 in present value.

Output 3.6.2. Summary of the Petroleum Distributor's Decision

```
                 Petroleum Distributor's Decision

                       The DTREE Procedure
                     Optimal Decision Summary

                         Order of Stages

                       Stage       Type
                       --------------------
                       Action      Decision
                       Response    Chance
                       Lawsuit     Chance
                       Last        Chance
                       Result      Chance
                       _ENDST_     End


                       Decision Parameters

          Decision Criterion:    Maximize Expected Value (MAXEV)
     Optimal Decision Yields:    -329


                     Optimal Decision Policy

                        Up to Stage Action

             Alternatives     Cumulative    Evaluating
             or Outcomes          Reward         Value
            ------------------------------------------
            Invoking                             -329*
            Not_Invoking                         -450
```

The decision tree for this problem is shown in Output 3.6.3. There you can find the expected value of each scenario.

Output 3.6.3. Decision Tree for the Petroleum Distributor's Decision



The president feels that the estimated likelihood of lawsuit outcomes is fairly reliable. However, the assessment of the likelihood of the customer's response and reaction is extremely difficult to estimate. Because of this, the president would like to keep the analysis as general as possible. His staff suggests using the symbols p and q to represent the probability that the customer will accept the invocation and the probability that the customer will decline to press the issue if he refuses the invocation, respectively. The probabilities of the other possible outcomes about the customer's response and reaction to the invocation of the 10% clause is listed in Table 3.39.

Table 3.39. Probabilities of the Petroleum Distributor's Decision

Uncertainty	Outcome	Probability
Customer's Response	Accept the Invocation	p
	Reject the Invocation	1-p
Customer's Action if the Invocation is being Rejected	Press the Issue	q
	Settle the Case	(1-q)/2
	Sue for Damages	(1-q)/2

Now from the decision tree shown in Output 3.6.3 on page 346, the expected value of the outcome '**`Refuse`**' is

$$\begin{aligned} \text{EV} &= 500q - 400(1-q)/2 - 672(1-q)/2 \\ &= 500q - 200 + 200q - 336 + 336q \\ &= 1036q - 536 \end{aligned}$$

Hence, the expected payoff if the petroleum distribution company invokes the clause is

$$\begin{aligned} \text{EV} &= 600p + (1036q - 536)(1-p) \\ &= 1136p + 1036q - 1036pq - 536 \\ &= 1136p + 1036(1-p)q - 536 \end{aligned}$$

Therefore, the president should invoke the 10% clause if

$$1136p + 1036(1-p)q - 536 > -450$$

or

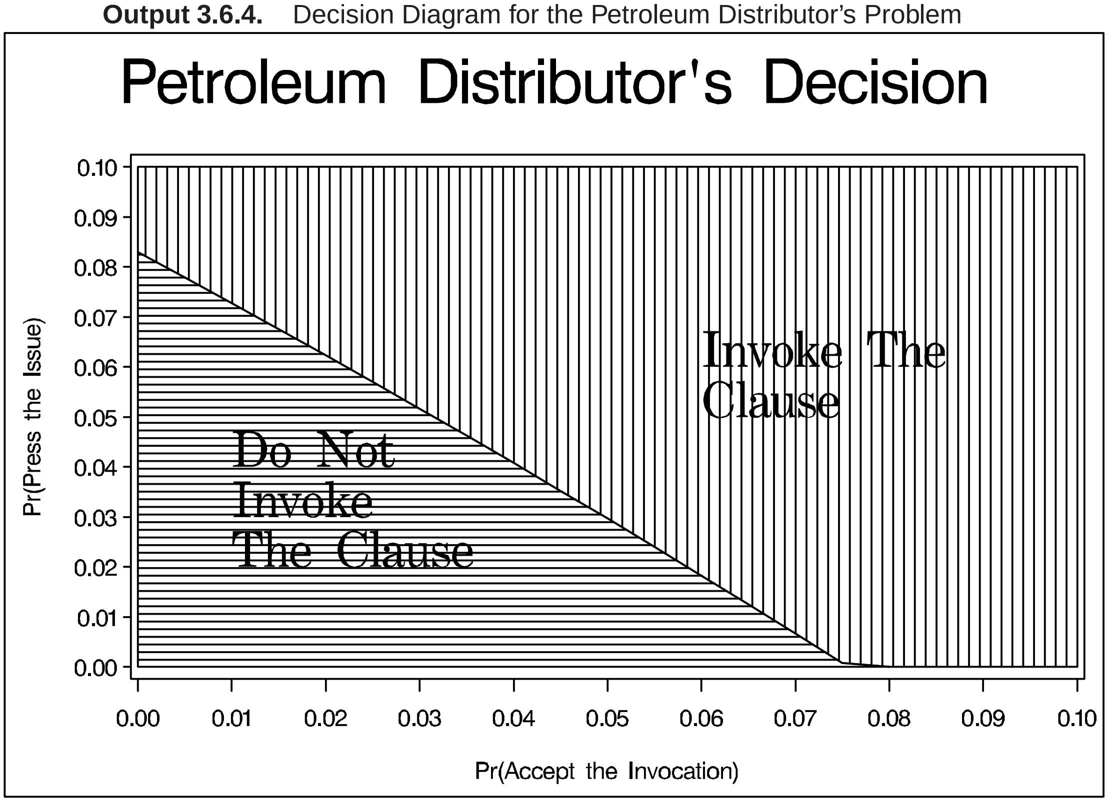
$$q > \frac{86 - 1136p}{1036 - 1036p}$$

This result is depicted in Output 3.6.4 on page 349, which is produced by the following statements:

```
   /* -- create data set for decision diagram        -- */
data Data7(drop=i);
   P=0.0;                  /* initialize P */

      /* loop for all possible values of P */
   do i=1 to 21;

         /* determine the corresponding Q  */
      Q=(86-(1136*P))/(1036*(1.0-P));
      if Q < 0.0 then Q=0.0;

         /* output this data point */
      output;

         /* set next possible value of P   */
      P=P+0.005;
   end;

run;

   /* create the ANNOTATE= data set for labels of  */
   /* decision diagram                             */
data label;
   length FUNCTION STYLE COLOR $8;
   length XSYS YSYS            $1;
   length WHEN POSITION        $1;
   length X Y                   8;
   length SIZE ROTATE           8;

   WHEN = 'A';
   POSITION='0';
   XSYS='2';
   YSYS='2';
   input FUNCTION $ X Y STYLE $ SIZE COLOR $
         ROTATE TEXT $ & 16.;
   datalines;
label   0.01    0.04    centx   2       black   .    Do Not
label   0.01    0.03    centx   2       black   .    Invoke
label   0.01    0.02    centx   2       black   .    The Clause
label   0.06    0.06    centx   2       black   .    Invoke The
label   0.06    0.05    centx   2       black   .    Clause
;

   /* -- set graphics environment             -- */
goptions lfactor=3;
```

```
   /* -- define symbol characteristics for boundary  -- */
symbol1 i=joint v=NONE l=1 ci=black;

   /* -- define pattern for area fill                -- */
pattern1 value=M2N0 color=black;
pattern2 value=M2N90 color=black;

   /* -- define axis characteristics                 -- */
axis1 label=('Pr(Accept the Invocation)')
      order=(0 to 0.1 by 0.01) minor=none;
axis2 label=(angle=90 'Pr(Press the Issue)')
      order=(0 to 0.1 by 0.01) minor=none;

   /* -- plot decision diagram                       -- */
title h=2.5 "Petroleum Distributor's Decision";
proc gplot data=Data7 ;
   plot Q*P=1 / haxis=axis1
                vaxis=axis2
                annotate=label
                name="dt6p2"
                frame
                areas=2;
run;

quit;
```

Output 3.6.4. Decision Diagram for the Petroleum Distributor's Problem



The decision diagram in Output 3.6.4 is an analysis of the sensitivity of the solution to the probabilities that the customer will accept the invocation and that the customer will decline to press the issue. He should invoke the clause if he feels the customer's probabilities of outcomes '**`Accept`**' and '**`Press_Issue`**', p and q, are located in the upper-right area marked as '**`Invoke The Clause`**' in Output 3.6.4 and should not invoke the clause otherwise. Note that the values $p = 0.1$ and $q = 0.1$ used in this example are located on the upper right corner on the diagram.

Statement and Option Cross-Reference Table

The following tables reference the statements and options in the DTREE procedure (except the PROC DTREE statement and the QUIT statement) that are illustrated by the examples in this section.

Table 3.40. Statements Specified in Examples

	Examples					
Statement	**1**	**2**	**3**	**4**	**5**	**6**
EVALUATE	X	X	X	X	X	X
MODIFY					X	
MOVE					X	
RECALL					X	
RESET	X					
SAVE					X	
SUMMARY	X	X			X	
TREEPLOT			X	X		X
VARIABLES	X					
VPC					X	
VPI					X	

Table 3.41. Options Specified in Examples

	Examples					
Option	**1**	**2**	**3**	**4**	**5**	**6**
ANNOTATE=				X		
COMPRESS			X	X		X
CRITERION=	X	X		X		
EVENT=	X					
FTEXT=			X			X
GRAPHICS			X	X		X
HSYMBOL=			X			X
HTEXT=			X			
LINKA=				X		
LINKB=				X		
LINKC=						
LSTYLEB=			X			
LWIDTH=			X			X
LWIDTHB=			X			X
NAME=			X	X		X
NOLEGEND				X		X
NOWARNING	X	X	X		X	
OUTCOME=	X					
PAYOFFS=	X	X	X	X	X	X
PROB=	X					
PROBIN=	X	X	X	X	X	X
REWARD=	X					
RT=	X	X		X		
STAGE=	X					
STAGEIN=	X	X	X	X	X	X
STATE=	X					
SUCCESSOR=	X					
SUMMARY					X	X
SYMBOLC=				X		
SYMBOLD=				X		
SYMBOLE=				X		
TARGET=	X	X			X	
TYPE=	X					
VALUE=	X					
YBETWEEN=			X			X

References

Baird, B.F. (1989), *Managerial Decisions Under Uncertainty: An Introduction to the Analysis of Decision Making*, New York: John Wiley and Sons, Inc.

Howard, R. A. (1968), "The Foundations of Decision Analysis," *IEEE Transactions on System Science and Cybernetics*, Vol. SSC–4, No. 3 (September), 211–219.

Howard, R. A. (1988), "Decision Analysis: Practice and Promise," *Management Science*, Vol. 34, No. 6, 679–695.

Raiffa, H. (1970), *Decision Analysis Introductory Lectures on Choices under Uncertainty*, Reading, MA: Addison–Wesley.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/ OR User's Guide: Project Management, Version 8,* Cary, NC: SAS Institute Inc., 1999. 806 pp.

SAS/OR User's Guide: Project Management, Version 8

ISBN 1–58025–492–6

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999