

**MARVL system analysis**

**Phil Carter**

**State Forests of NSW**

**1998**

---

**Table of contents**

Table of contents	2
Introduction	3
Overview of forest inventory	3
Overview of the MARVL process	4
The MARVL system	6
Components of the MARVL system	6
Inventory specification	6
Data collection and entry	9
Data import (and export)	13
Analysis	13
View Designer	16
Report Designer	19
Function Set Designer	21
Strategy Designer	23
User-variable Designer	23
Dictionary Designer	23
The MARVL database	23
Linking MARVL and GIS	24
References	27
Appendix 1. List of standard formulas used by MARVL	28
Notation	28
Appendix 2. List of standard function forms.	30
Stem breakage	30
Height/age relationships	30
Height/DBHOB relationships	31
Taper functions	32
Volume functions	34
Appendix 3. MARVL variables	36
Log level variables	36
Piece level variables	37
Tree level variables	37
Plot level variables	38
Appendix 4 Example report section definitions	39
Appendix 5. MARVL database schema	42

## Introduction

MARVL is a forest inventory method designed to provide detailed information on the potential yield and log size distribution likely to result from felling a stand of trees. The method was first described by Deadman and Goulding (1978). The acronym MARVL is derived from “**M**ethod for **A**ssessment of **R**ecoverable **V**olume by **L**og Types”.

The purposes of this document are to:

- briefly describe the forest inventory process
- identify the role of MARVL in forest inventory process, and,
- describe in detail the MARVL method in relation to the forest inventory process.

Much of the MARVL method is based on standard forest mensuration techniques. What is emphasised in this report are operations which are unusual, are poorly documented, or are unique to MARVL.

## Overview of forest inventory

Husch *et al.* (1982) define forest inventory as:

“...the procedure for obtaining information on the quantity and quality of the forest resource and many of the characteristics of the land area on which the trees are growing.”

Husch *et al.* give the following checklist of items which may need to be considered in the planning of a forest inventory:

1. Purpose of the inventory
2. Background information
  - Past surveys, maps, reports etc
3. Description of the area
  - Location, size, terrain, accessibility
4. Information required for the final report
  - Tables, graphs, maps, narrative report
5. Inventory design
  - Estimation of area, determination of timber quantity, size and shape of sampling units, sampling method, precision
6. Procedures for aerial photograph interpretation
7. Procedures for field work
  - Location and establishment of sampling units, current stand information, recording of observations, data conversion and editing
8. Compilation and calculation procedures
  - Instructions for reduction of field measurements
9. Final report
10. Maintenance
  - Storage and retrieval of data

Item 8 in this list is directly provided for by MARVL; MARVL has indirect bearing on items 4, 5, 7, 9 and 10.

### **Overview of the MARVL method**

MARVL was developed in New Zealand in response to perceived deficiencies in the existing plantation inventory procedure. Deadman and Goulding (1978) listed the principal deficiencies as:

1. an inability to provide detailed product breakdown, particularly when it was necessary to differentiate the merchantability of individual stands for diverse markets,
2. the inability of current inventory to provide detailed information at the stand level.

MARVL differs from other inventory systems in that it separates the field assessment of size and quality of stems from the actual cross-cutting. When the stand is cruised, no attempt is made to divide the stem into logs or estimate merchantable limits at any point on the tree. (Lawrence, 1986).

Use of MARVL involves 3 basic steps (NZFRI, 1995):

1. inventory design,
2. sampling of stand(s) to assess tree size, structure and quality,
3. analysis of the sample data to determine potential product yield.

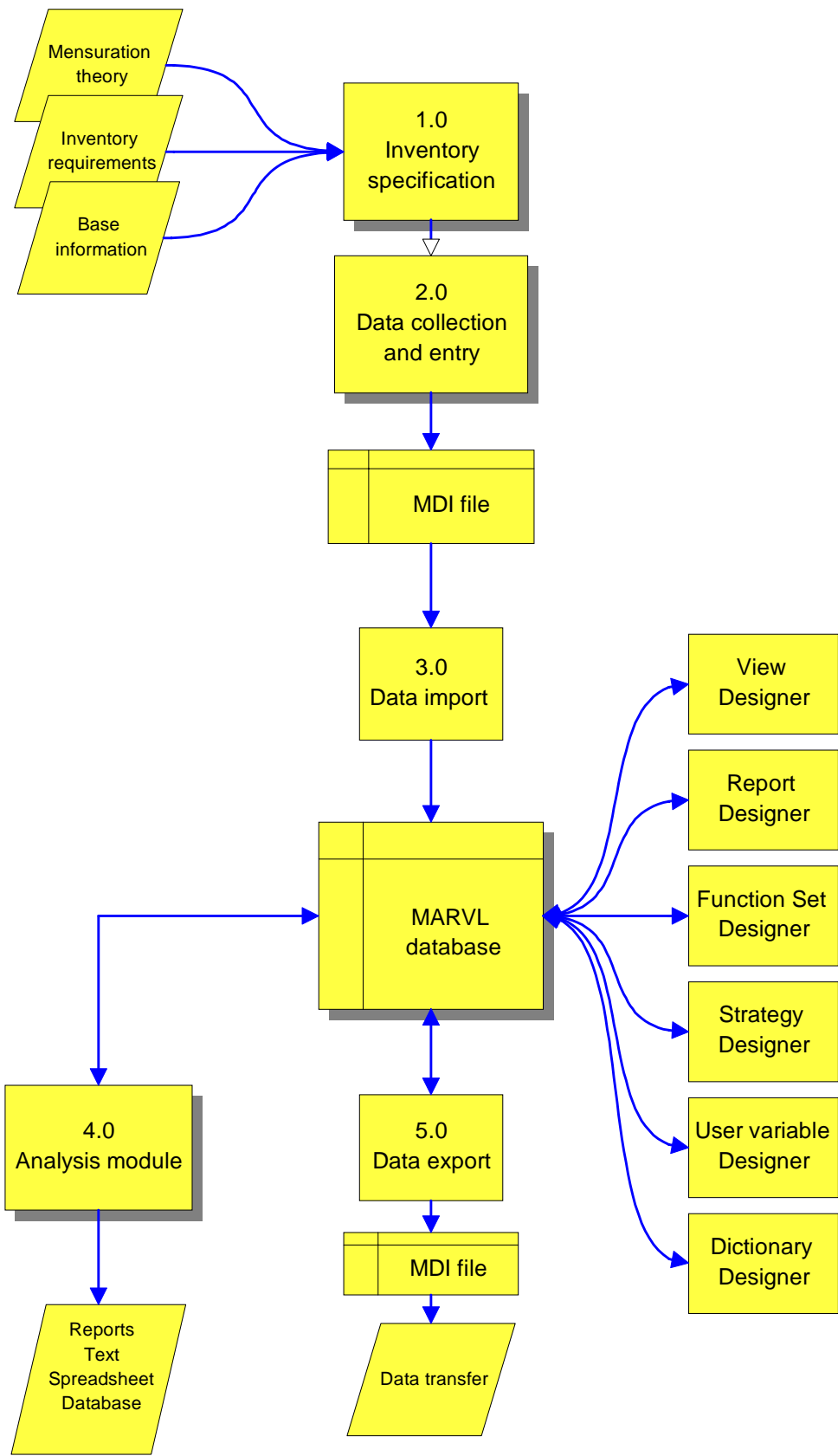
At step 1, MARVL supports the use of fixed area plots (“bounded” plots in MARVL literature), horizontal point and horizontal line samples, which may be used in simple or stratified designs. There is a “double sampling” option at the plot level.

Step 2 is referred to in MARVL literature as “cruising”. Standard tree size indices (such as DBHOB, height) are measured in this step. There is the facility for the inclusion of “user-defined” variables. In addition, each tree is described by structural and quality codes.

Step 3 is accomplished with the Analyse module of MARVL. This module enables the user to produce 1 or more reports, using 1 or more views as input, with 1 or more cutting strategies, to 1 or more projection dates.

Reports may be either standard, or custom. A “view” is a named list of plots (NZFRI, 1995). A group of views is called a “plan”. Inventory data may be projected to a specified date if suitable growth models are available. A “cutting strategy” is a list of rules which define products in terms of permissible quality codes, minimum/maximum dimensions, value and acceptable species.

Figure 1. MARVL system flow chart



## **The MARVL system**

The MARVL system is described in detail in this section. Processes and the flow of data in the MARVL system is illustrated in Figure 1. This diagram shows only the basic outline of MARVL; many interactions between the components have been omitted from this diagram for the sake of clarity. These interactions are explained in the following text.

## **Components of the MARVL system**

### ***Inventory specification***

Process 1.0 of Figure 1 (which corresponds to item 5 of the Husch *et al.* (1982) list) is discussed in this section.

### **Sampling issues**

The issue of inventory design is, to a certain extent, external to MARVL. MARVL understands a limited number of inventory design types, inventory designs outside this subset can't be processed. The issue of whether a particular design is more suitable for a particular job than some other design will not be solved by MARVL.

MARVL samples may be either fixed area plots, horizontal point or horizontal line samples, in either a simple or stratified design. Fixed area plots within the same stratum must be of the same area.

Double, or 2-phase, sampling relies on a close correlation between some easily-measured variable, the “auxiliary variate”, and a variable of interest that is not so easy to measure. The first phase involves measuring the auxiliary variate, (basal area, for instance) on a large sample. The second phase involves measuring the variable of interest, volume for example, (and the auxiliary variate if the second phase sample is not drawn from the first phase sample) on a smaller sample. (Cochran, 1977; De Vries, 1986).

MARVL application of double sampling is explained as follows: Fully-measured plots are termed “primary plots”. “Secondary plots” are measured for basal area only. Statistics, such as volume, are adjusted by the ratio of basal area per hectare over the entire sample (secondary plots plus primary plots) to the basal area per hectare of the fully measured plots (primary plots only). Because tree volume is a function of basal area and height, basal area can be expected to be a good predictor of volume if average tree size is reasonably uniform within a stratum (NZFRI, 1995).

Use of the terms of “primary” and “secondary” samples in the context of double sampling does not accord with terms used in standard texts. This is confusing. The first phase of this 2-phase sample is primary plus secondary plots. The second phase is the sample of primary plots.

Decisions about what variables need to be measured to achieve the aims of the inventory are an essential part of inventory design. MARVL provides a set of default inventory variables, which are adequate for simple inventories. For more complex jobs, user defined variables may be added. These variables are discussed in following sections.

### Default variables

MARVL has a set of default variables at both the plot and tree level. These are listed in Table 1.

Table 1. MARVL default variables

Level	Variable
Plot	Unique plot key
	User who last modified the data
	Date and time the data were changed
	Plot number
	Inventory name
	Dictionary name
	Year plot was planted
	Year plot was measured
	Month plot was measured
	Number of trees in the plot (live and dead)
	Slope
	Plot type (fixed area, horizontal point or line)
	Plot area (if fixed area)
	Plot length (if horizontal line)
	BAF (if horizontal point or line)
	Live basal area
	Live stocking
	Mean top diameter
	GIS link key
	Description of plot
Tree	Unique plot key
	Tree number
	Stocking represented by this tree
	DBH
	Height
	Live/dead
	Species
	Collapsed stem description

The meaning of most of these variables will be apparent; comments on some of the less obvious variables follow.

The distinction between unique plot key and plot number should be remembered. The unique key is generated by MARVL to ensure that all plots in the database are uniquely identified. The plot number is the plot identifier assigned by the inventory designer. 2 separate inventories could both have a plot number 1, their identity in the database is maintained with the unique key.

Year plot was planted can be safely ignored for regrowth forest; a default value can be entered here. This field is useful for plantations.

Mean top diameter is defined as the root mean square (RMS) of the 100 largest DBH live trees per hectare. Mean top height is obtained by solving the height/diameter function for a tree of mean top diameter (NZFRI, 1995).

GIS link key is yet another unique identifier for the plot, used in conjunction with template, notification or query files (see section titled “Linking MARVL and GIS”).

Stocking represented by this tree is generated by MARVL, and is the frequency used in conversion of the plot data to a per hectare basis. Its value is set by the system. It is not normally accessible to the user, it may be modified by growth models to account for mortality. (Lawrence, pers. comm.)

#### User defined variables

Default plot and tree level variables may be supplemented by user defined variables, of which there are 3 types (NZFRI, 1995):

1. tree user variables, associated with trees in a plot,
2. plot user variables, associated with each plot,
3. plot extra variables.

Tree and plot user variables may be either numeric, or “ordinal”. Interval or ratio variables are handled by numeric user variables. Ordinal variables would be better called “categorical” variables, because, in practice, they may be either nominal or ordinal (see Husch *et al.* (1982) for a discussion of scales of measurement.)

Minima, maxima, measurement units, and display formats may be specified for numeric variables.

A lists of legal values may be specified for each ordinal variable.

Plot extra variables are, in effect, user defined default variables. While the definitions for both plot user variables and plot extra variables are stored in the same table, the values of plot extra variables are stored in the plot table itself. (Values for plot user variables are stored in a separate table.) Plot extra variables may be of the string type, plot user variables may not. Plot extra variables may not be used in a cutting strategy, whereas (beginning with MARVL V3.3) plot user variables may be (Mein, pers. comm.). NZFRI (1995) recommends that the creation of plot extra variables not be undertaken lightly.



### ***Data collection and entry***

An expanded version of process 2.0 of Figure 1, data collection and entry, is shown in Figure 2. This process is a sub-item of item 8, with elements of item 7 of the Husch *et al.* (1982) list. In brief summary, this process involves:

- creation of a template data interchange file (process 2.1)
- field data collection (process 2.21)
- data entry (process 2.22)

The end result of this process is an interchange file which contains the data for the entire inventory. The process described above, is relevant to a manual system, where inventory data are hand written on plot sheets. If electronic field data capture is used, field data collection and data entry are collapsed into a single step (process 2.3).

Aspects of process 2.0, including tree description methods, are discussed in this section.

#### **Field measurement**

The field measurement process of an inventory is normally described in a document specific to each inventory. This document should state what is to be measured, and give precise instructions as to how each variable is to be measured, and how it is to be recorded.

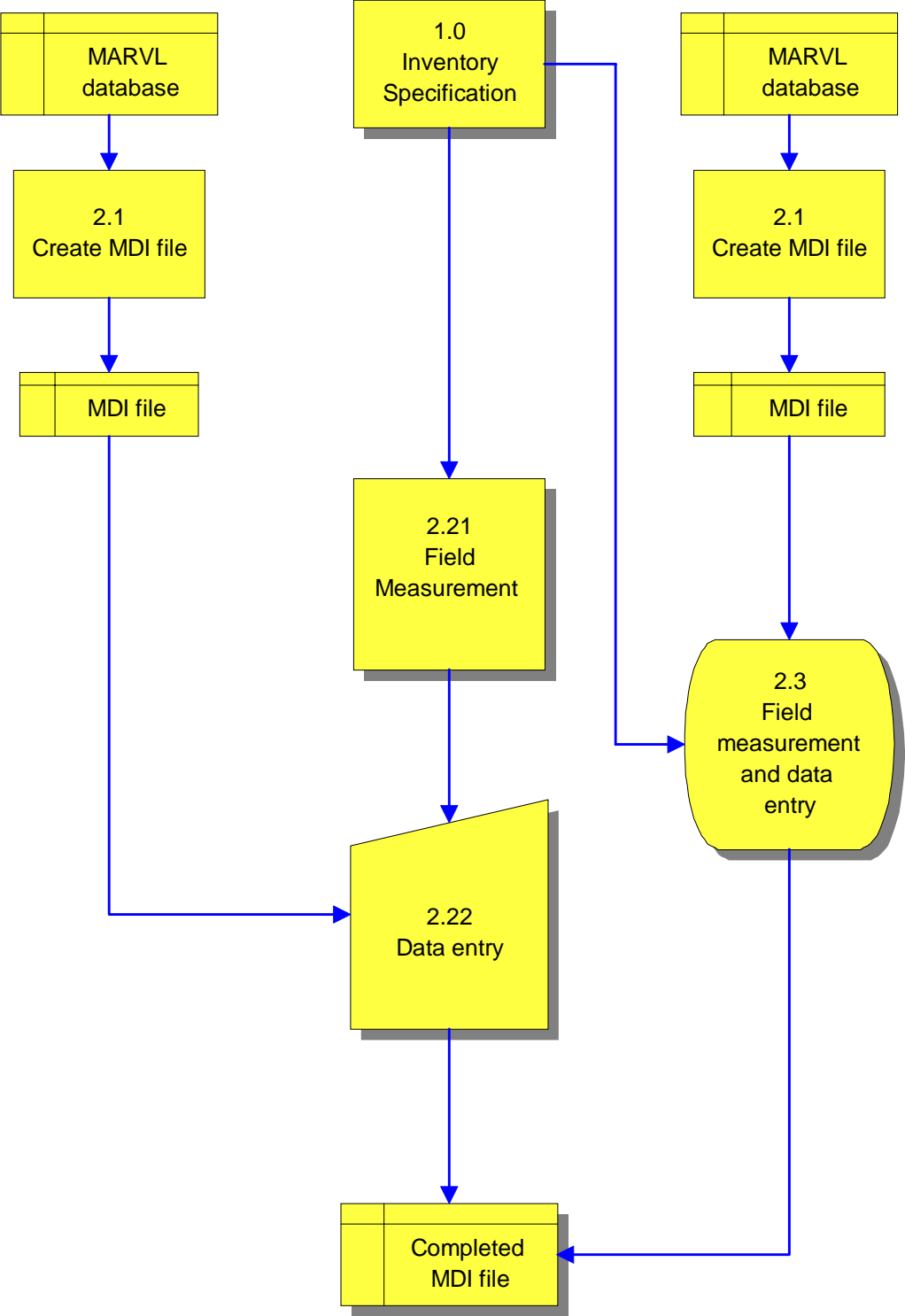
#### **Data entry**

The method by which collected data are transferred to (and from) the MARVL database is outlined in this section. The method is described in detail in NZFRI (1997).

The MARVL data capture program MARVLDE3 is used to collect MARVL inventory data, either on a PC in the office, entering data from hand written plot sheets, or in the field, entering data directly into a DOS-based portable data recorder (PDR). A series of menus and data entry screens (forms) allows the user to:

1. enter data at the inventory, stratum and plot level,
2. edit existing data,
3. transfer data from one computer to another, typically to or from a PDR.

Figure 2. MARVL data collection and entry



MARVLDE3 can be configured for different screens, default file locations, file transfer settings and range checks (among others). Range checking is particularly important for data quality. 2 levels of range check are available. The first level merely requests confirmation of suspicious values from the operator. Values outside second level ranges are not accepted by the program.

Data entry, editing and transfer is based on files known as MARVL Data Interchange (MDI) files. MDI files are used to:

1. store inputs from a data capture program, for transfer to the MARVL database,
2. transfer data from 1 MARVL installation to another (different database, site or company).

The “data” referred to here includes such objects as views, dictionaries, cutting strategies, function sets, species code lists and user defined variables, as well as measured plot and tree data.

An MDI file must exist before any data can be physically entered. At a minimum, this file must contain a quality code dictionary, 1 or more function sets (see note below) and a species code list, and optionally, may contain other objects noted above. This file is created using the Export utility of the View Designer.

Function sets may be assigned to individual strata; this assignment is made during data entry. If more than 1 function set is required, they will need to be present in the MDI file.

When transferring data, the MDI file may contain 1 or more entire inventories (with their associated dictionaries, function sets and user-defined variables). 1 or more plans (with their associated views), or individual objects such as cutting strategies. These files are also created using the Export utility of the View designer.

The MARVL system requires that all data for an inventory be present in a single MDI file for import. This is easy enough to manage for a small inventory, but for larger inventories, or inventories where there is more than 1 crew entering data, there is currently no formal method for merging individual MDI files. This problem can be circumvented by editing MDI files using a text editor, but care is needed when using this method because of the complex format of MDI files. Another work-around is to use the View Designer (see below), which can be used to import multiple data sets from a single field inventory into the MARVL database as separate inventories (Pont, pers. comm.).

The contents of MDI files are loaded into the MARVL database using an Import utility, a function of the View Designer. Both the View designer and the MARVL database are described in later sections.

### Tree description codes

Each tree in a MARVL inventory is completely described by a series of codes. The description method is discussed in this section.

There are 2 types of description code: structural codes and quality codes.

Structural codes describe the morphology of the tree, and are listed in Table 2.

Table 2. MARVL structural codes

Code	Meaning
*	Broken or dead top
>	Diameter reduction
<	Fork (or reduction and forced cut)
%	Forced cut
+	Merchantable branch

“Quality” is a classification of all or part of a tree stem in terms of those factors which have been found to have a significant impact on log value. Deadman and Goulding (1978) noted that the product yield of a [radiata pine]stand [in New Zealand] is influenced by both the characteristics of the stand and by the method of cross-cutting the stems. Gordon and Lawrence (1995) noted that the main external stem features which affect log quality [in New Zealand] are pruning, branch size and sweep, other features that may be important include fluting/buttrressing, out-of round, nodal length, nodal swelling and wood damage.

Quality classification is fundamental to the MARVL method, and is implemented by defining 1 or more “quality codes”. Quality codes are specified in a “quality code dictionary”, which serves 2 purposes (NZFRI, 1995):

1. it tells MARVL which letters of the alphabet will be used as quality codes,
2. it provides the user with an opportunity to document the exact meaning of each code, and to describe the intended area of application of the dictionary.

In the formulation of a quality code dictionary, it is necessary to assign a single letter of the alphabet to a log quality class, and to fully describe that quality class. Any letters may be assigned to any quality class. Parts of the tree coded with letters not found in the quality code dictionary are discarded as waste.

Robust quality codes are fundamental to MARVL. Quality codes should be kept as independent as possible from particular log product specifications, to fully exploit the ability of MARVL to re-analyse inventory data to a different set of log product specifications. In particular, length or diameter constraints must not be included in the definition of quality codes (Gordon and Lawrence, 1995).

A number of New Zealand authors recommend the use of a decision tree to implement quality codes.

***Data import (and export)***

Completed MDI files are transferred to the MARVL database using an Import facility (process 3.0 in Figure 1). This facility is part of the View Designer, and is described in a following section.

***Analysis***

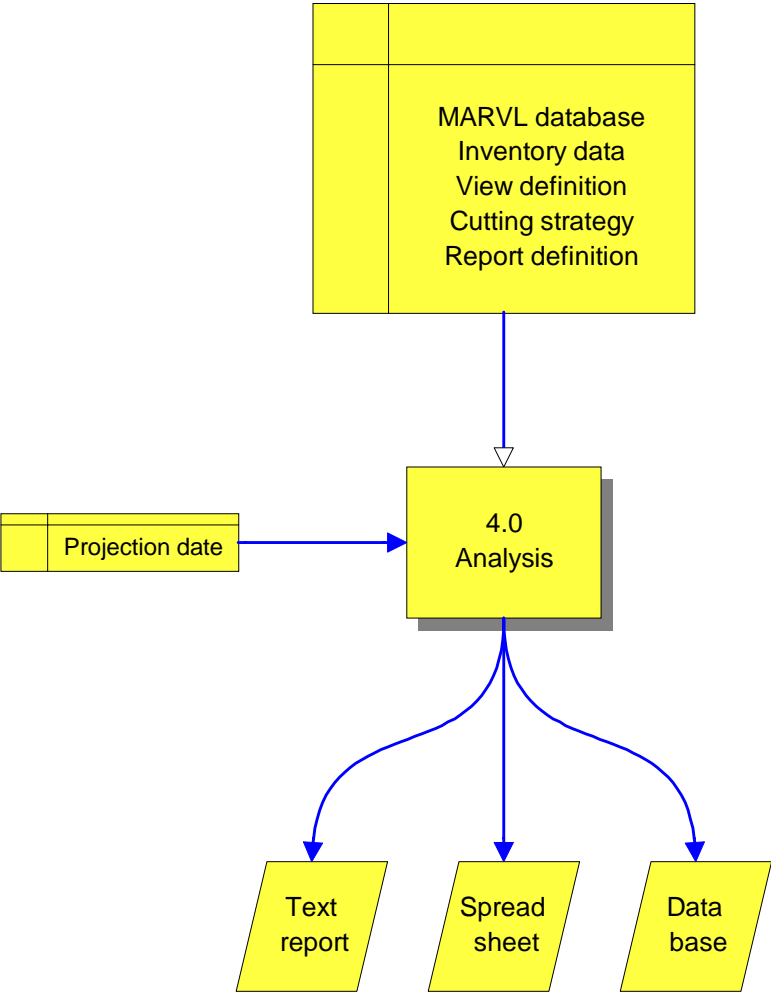
The Analyse module is the heart of MARVL. The analysis module uses data from the database to produce reports. This process is described in Figure 3, and represents item 8 in the Husch *et al.* (1982) list.

An analysis may be done on a single view, or a plan, which is a named group of views. One or more projection dates may be selected; a date of 0/0 processes the data as measured. One or more cutting strategies may be selected. One or more reports may be selected. The analysis module produces 1 report for each combination of projection data, cutting strategy. The optimisation program is run for each requested report; the process can be time-consuming if many reports are requested.

Analyses may be run interactively, or by using a batch file. Batched analyses, perhaps run overnight, may be the answer for large and/or complex jobs.

The optimisation process is described in the next section. The reporting process and cutting strategies are described in following sections.

Figure 3. MARVL Analyse module



### Optimisation

As noted previously, what distinguishes MARVL from other inventory systems is the separation of the field assessment of size and quality of stems from the actual cross-cutting. What this means in terms of the MARVL system is that, in the analysis phase, each tree is converted into logs using the constraints specified by the cutting strategy. The process is (NZFRI, 1995):

1. Each tree is cut at a uniform stump height.
2. Depending on the breakage function used, the tree may be considered to have broken on falling. If this is the case, the part of the tree above the predicted break point is discarded as waste.
3. The tree is trimmed by making cuts at all forks (structural code = "<") and forced cuts (structural code = "%"). The results of this trimming are termed "pieces".
4. Each of the pieces that remains is analysed using the cutting strategy to find the combination of log types and lengths that gives the greatest total value, while satisfying all of the size and quality constraints of the log types. This step is called "optimisation".

The optimisation step is based on a dynamic programming method originally conceived by Pnevmticos and Mann (1972) (Deadman and Goulding, 1978). The method actually used in MARVL 3 is described in detail in NZFRI (1995); major points are repeated here.

MARVL attempts to cut logs at a limited number of (equally-spaced) points along the stem. It further assumes that any log that may be cut will fit exactly between two of these positions. The distance between adjacent stem positions under this model is called the "round-off" length, because all log lengths and stem heights (relative to the stump) are rounded-off to multiples of this length.

It is important, in selecting a round-off length, to balance precision and system performance. Very small round-off lengths increase running time. An analysis that uses a round-off length of 0.25m would take about 4 times as long to run as an analysis that uses 1.0m. If you wish to cut 3m and 4m logs and use a round-off length of 0.7m, MARVL would actually cut 2.8m and 4.2m logs, these being the nearest multiples of the round-off lengths.

Random log lengths are far slower to analyse than fixed lengths, because, at every position along a stem, MARVL has to consider every possible log length that could be cut there. An option here may be to represent the random lengths as a few fixed lengths.

You may supply a value for the cost of making a saw cut. The main reason for doing this is to encourage cutting of a long log in preference to 2 short logs of the same value.

Stem diameters and volumes are calculated with stored or programmed taper and volume functions. Functions to be used are specified by the user.

### ***View Designer***

MARVL analyses are based on 1 or more “views”; a view is a named list of plots. Views are managed with the “View Designer”.

The View Designer performs several functions:

- data import/export
- view definition
- view validation
- view management
- pilot survey (estimation of required number of samples)
- configuration

#### **Data import/export**

The data import/export facility is primarily used for importing data into the MARVL database from an MDI file. Typically, all the data from a single inventory will be in the MDI file, and this will be imported into the database when data entry is complete. Data export is also managed with MDI files.

Other data import functions include:

- import MQF (GIS query) file (see “Linking MARVL and GIS” below)
- import SQL query (using standard SQL commands)
- import a database file

Other export functions are;

- export inventory template file (with GIS-located points)
- export MicroMARVL file (export V2 files)

#### **View definition**

As noted above, a “view” in MARVL is a named list of plots. The plots in a view may comprise all or some of the plots from an entire inventory, or some or all of the plots from more than 1 inventory. For analysis convenience, a group of views may be gathered into a “plan”, a named list of views.

Views are presented in the View Designer as a series of plot, group and stratum lines. The group referred to here is a growth group, a group of plots measured on the same date, and which are grown forward together. A sample section of a typical view is shown in Table 3. In Table 3, stratum lines are indicated with “===”; group lines are indicated with “---” and plot lines are indicated with the word “Plot”.



Table 3. Typical MARVL View Designer window

```

=== 100.00          Coastal Species
--- 7/1997
      Plot14017 Strategic Inventory Bounded 39
      Plot14017 Strategic Inventory Bounded 32
--- 6/1997
      Plot14017 Strategic Inventory Bounded 39

```

Stratum, group and plot lines may be customised to meet user requirements. This is discussed in “Configuration”, below.

#### View validation

All views must be validated to ensure that they are capable of analysis (NZFRI, 1995). Validation rules confirm that:

- within a growth group,
  - the number of plots is within the limits set by PlotsInGroup<sup>1</sup>,
  - PLEs for basal area and stocking are within limits set by GroupPLEBA and GroupPLESPH,
  - there are no duplicate plots,
- within a stratum,
  - all secondary plots have the same type and same size,
  - there is at least 1 primary plot, and that all primary plots have the same type and size,
  - there are no duplicate plots,
  - the function set name is valid,
  - the area is within limits set by StratumAreaRange,
- within a view,
  - there are no plots duplicated between strata (except for height only plots, which may be shared to create a common height regression).

#### View management

At the base level, all the plots from a single inventory (referred to as a design inventory in MARVL literature) form a default view. New views are created by copying the required parts from 1 or more views, and pasting them to a new view. Parts may be copied from more than 1 inventory. New views must be validated before they can be analysed.

Views may be saved or deleted.

#### Pilot survey

The Pilot Survey facility may be used to estimate the required number of samples. PLE calculations are based on data from a subsample of plots.

---

<sup>1</sup> Values for PlotsInGroup, GroupPLEBA, GroupPLESPH and StratumAreaRange are set in the configuration file, MARVL.INI.

### Configuration

Plot, group and stratum lines may be customised to meet user requirements. Fields to be displayed, and the order in which they are displayed, are selected in a dialog box.

Information fields, and formatting fields available for customising are shown in Table 4.

Table 4. Customising field available in the View Designer

Line type	Variables	Formatting
Stratum	Area	---
	Function set	====
		comma
		space
		'Stratum'
Group	BA CI/PLE	---
	BA Mean	====
	DBH Mean	comma
	Double sampling ratio	space
	Established year	tab
	Measure date	'Group'
	No of primary plots	
	No of secondary plots	
	Stocking CI/PLE	
	Stocking Mean	
Plot	Area	---
	BA	====
	BAF	comma
	Database plot key	space
	DBH Mean	tab
	Design inventory	'Plot'
	Dictionary name	
	Established year	
	Heights in regression	
	Heights in regression (Ab)	
	Horizontal line length	
	Last modified (by user)	
	Last modified (date)	
	Mean top diameter	
	Measured date	
	Plot number	
	Plot type	
	Plot type (Ab)	
	Primary/secondary	
	Primary/secondary (Ab)	
	Stocking	
	Trees (count)	

Note: "Ab" in Table 4 stands for "abbreviated".

### ***Report Designer***

Results of inventories processed by MARVL are obtained as reports. Reports may be output in the form of text files or in spreadsheet or database form. The Report Designer is used to create new reports, or to modify existing reports.

### **Report Characteristics**

A MARVL report consists of one or more “sections”. All reports are specified in their own “Report Definition File”, which have the extension RPD, and which contain the following information:

- report description,
- output file name, with append/overwrite option,
- output format, may be text, worksheet, or database,
- list of included sections, with codes to describe the level (plot, stratum or population) and to show/not show titles and totals.

### **Report sections**

Report sections may be either “built-in”, which can’t be modified, or “custom”, which can be modified (NZFRI, 1995). Built-in report sections are listed in Table 5.

Table 5. MARVL built-in report sections

Name	Description
LogGroup	Allows log types to be grouped together to calculate statistics
LogTrace	Lists each log produced by the bucker
Strategy	Population level only. Prints the strategy used to produce this set of results
Dictionary	Population level only. Prints the dictionaries needed by the strategy
Functions	Stratum level only. Prints the function set used to produce this set of results
InventoryDescription	Population level only. Prints the description field from the inventory table, and some other useful information
PlotDescription	Plot level only. Description of the plot including plot type, area and number
StratumDescription	Stratum level only. Stratum area, number of primary and secondary plots and the double sampling ratio

Custom report sections are defined in a file titled REPORT.INI. This file may contain 1 or more section definitions. A text editor may be used to add new section definitions to this file, or to modify existing report sections.

Custom report sections are defined with a report definition language, that is described in detail in NZFRI (1995); a summary of this information is provided here.

Lists of allowable report analysis variables (just called “variables”) are given in NZFRI (1995). There are 4 variable lists, for logs, pieces, trees and plots. (See Appendix 3 for complete lists.) For any 1 report section, variables may be selected from only 1 list. Generally, variables form the column headings of the new report section (but see below for modifications). Class variables form the row headings, these must also be selected from the lists of allowable variables. Class widths, a start-point and a number of classes may be specified for class variables. Variables may be grouped using “By variables” (allowable variables also listed). A given variable may not be available for use in all categories (analysis, class or by variable). Statistics such as errors may be included for some analysis variables. User-defined variables may be used in reports definitions, subject to some restrictions.

Samples of custom report section definitions are shown in Appendix 4.

MARVL reports may be thought of, somewhat arbitrarily, as belonging to 1 of 2 categories, standard reports or custom reports. MARVL is supplied with 5 reports already installed, these are the standard reports. Standard reports are very general in nature, so will cover a wide range of possible output requirements. Custom reports are all other possible reports.

### Standard reports

Standard reports may be composed of both built-in and custom report sections; custom sections of standard reports may be modified as required. Care is needed when modifying report section definitions; any single section definition may be used in more than 1 report. If modifications to a standard report are required, the best approach may be to copy the entire definition to a new report definition file, and edit the new file, and/or the REPORT.INI file, as required.

Standard MARVL reports include:

- standard population report
- standard population and stratum report
- standard population, stratum and plot report
- Log trace
- Yield table

Standard population reports, standard population and stratum reports and standard population, stratum and plot reports contains the following report sections:

---

InventoryDescription  
Strategy  
Dictionary  
Functions  
StandingResource  
StandardResource  
LiveStandingTrees  
CutoverResidue  
ExtractableStems  
ProductAnalysis  
CuttingWaste  
LogGroup

The difference between these 3 reports is the level at which results are reported.

The Log trace report is intended for diagnostic or investigative use, it contains a single built-in section only, LogTrace. This report writes the results of the optimisation process for each tree in the selected view to the selected output form.

The Yield table report contains a single custom section, YieldTable. A yield table presents anticipated yields from a stand at various points in time (Vanclay, 1994). Because there are currently no native forest growth models that are compatible with MARVL, yield tables for native forests can't be produced with MARVL as yet.

#### Custom reports

As noted above, a custom report consists of 1 or more custom report sections, with the option of 1 or more built-in sections.

Custom report sections may be created from scratch by the user, using a text editor to modify the REPORT.INI file, creating a new report section using the report definition language described above. Existing report sections may be modified by the same method.

#### ***Function Set Designer***

Functions to be used in the calculations for a given inventory are selected by the user with the Function Set Designer. A Function Set is a named list of functions.

MARVL has been built around the use of functions for:

stem breakage,  
growth models,  
growth adjustments,  
height/age relationships,  
height/DBHOB relationships,  
taper and volume.

The use of a taper and volume function and either a height/age or a height/DBHOB relationship - supplied by the user or fitted by MARVL - is mandatory. The use of other functions is optional.

Breakage models predict the height at which a tree stem will break on felling. That part of the tree above the break point is classed as waste. There is a “no breakage” option, which is the option currently used for regrowth forests in NSW.

Growth models may be used to project inventory data in time. Growth models are currently all one-offs, each has a unique functional form, and coefficients are built into the model form (Mein, pers. comm.).

Growth adjustment tables specify the proportion of annual growth that has occurred on a monthly basis. (NZ growth models for softwood species are based on monthly time increments; Lawrence, pers. comm.)

Height/age and height/DBHOB relationships are used to estimate a height for unmeasured trees, so that taper and volume functions, which use height as an independent variable, can be applied. Height/age functions tend to be one-offs, there are some models that share functional forms (Mein, pers. comm.). Height/age relationships are of no relevance to regrowth forest where age is not known, but may be relevant to native species plantations.

Height/DBHOB relationships may be specified as a function, or as a conversion table specified by the user. If a function is specified, a function is fitted to the measured data for each stratum in the inventory.

Taper and volume functions are used to predict stem diameters and volumes. Taper and volume functions must be “compatible”, that is, total volume estimates, based on integration of a taper equation, should be identical to those given by a tree volume equation (Demaerschalk, 1972). The reason for the requirement of compatibility is so that volume estimated by summing volume estimates of parts of the stem exactly equals the estimated volume of the entire stem.

In a mixed-species forest, there are often groups of species, which have similar characteristics, but which are different from other groups. Similar species may be grouped using “function groups” (NZFRI, 1995).

MARVL software comes with numerous functions already installed. Most of these are applicable only to exotic species in New Zealand. Currently, the only functions of relevance to regrowth forests in NSW are taper and volume functions for blackbutt, flooded gum, and “coastal species” (blackbutt plus flooded gum), based on taper models developed by Muhairwe (1995).

MARVL recognises a couple of standard forms of each function; functions forms for breakage, height/DBHOB, taper and volume are listed in Appendix 2. If relationships in one of these forms are known, coefficients can be entered into a text file in standard form, and installed in the software using a translate utility. If relationships are not in standard form, they must be programmed.

MARVL assigns functions sets at the stratum level, it is possible to use different function sets for different strata within the 1 inventory. This assignment is made in the data entry phase.

### ***Strategy Designer***

A “cutting strategy” is a list of product types, each of which is specified by required lengths, minimum and maximum small- and large-end diameters, permitted quality codes, value, acceptable species (Anon, 1990). Beginning with V3.3 of MARVL, user variables may be used in a cutting strategy. Cutting strategies are created or modified with the Strategy Designer.

A cutting strategy is based on a quality code dictionary, but, unlike a plot, a cutting strategy can refer to more than 1 dictionary, so that plots cruised to different sets of quality codes can be analysed as a single inventory (NZFRI, 1995).

It is important to understand that the “value” specified in a cutting strategy does not need to be an actual dollar amount. It may be an actual market value, an internal transfer price, or an artificial price. If an artificial price is used, its magnitude should accurately reflect relative value to the user (NZFRI, 1995).

Any number of cutting strategies may be applied to a single inventory.

### ***User-variable Designer***

The significance of user defined variables was discussed in as earlier section. Plot and tree user variables are created or modified with the User-variable Designer.

Plot extra variables are created by modifying the database structure itself (NZFRI, 1995).

### ***Dictionary Designer***

Quality code dictionaries are created or modified with the Dictionary Designer. Quality codes and the Dictionary are discussed in a previous section “Tree description codes”.

### ***The MARVL database***

Plot and inventory data are stored in a database to provide safe access to the data by multiple users, and to formalise the relationships between the data entities that MARVL uses (NZFRI, 1995). Data are transferred into the MARVL database from MDI files using an import utility, as described elsewhere.

MARVL was developed and tuned to run best with Borland Paradox database software, but other database software may be used. Borland SQL Links for Windows is a set of drivers supported by MARVL that will directly connect to Oracle, SQL Server, Sybase, Interbase or Informix. MARVL can use other database products that have an ODBC driver, but with a reduction in performance.

The MARVL database may be installed either locally, or on a network server, so it is available to multiple users. MARVL runs fastest when using its database on a local hard disk.

The database schema is shown in Appendix 5.

#### Other important files

There are a number of files crucial to the running of MARVL that are not stored as part of the MARVL database. These are:

**Configuration files.** System-wide defaults are stored in a file called MARVL.INI. Settings for each user are stored in a separate file, MARVLUSR.INI. Individual user preferences will override system settings (NZFRI, 1995). The data entry program, MARVLDE3, also has its own configuration file, MARVLDE3.INI.

**Reports.** Each report is defined in a file with the extension RPD. Report sections are defined in a file called REPORT.INI.

**Data interchange.** Files with the extension MDI are used to import/export data.

**Templates.** Plot points generated by GIS are stored in a file with an MDF extension.

**Notifications.** Notification of plot locations (to GIS) are stored in files with an extension MNF.

**Query files.** Plots selected with GIS are stored in a file with the extension MQF.

The creation and use of template, notification and query files is explained in the following section.

#### ***Linking MARVL and GIS***

MARVL may be linked to GIS by a variable called GIS Link Key. This key is known to both MARVL and the GIS. There are 2 ways to assign a GIS Link Key (NZFRI, 1995):

1. Unique values can be generated by the GIS, MARVL is informed of these values by a Template (MDF) file. The MDF file is read by the view designer, which produces a template inventory file (an MDI file) with empty plots, ready for filling in.
2. Unique values can be generated by MARVL. MARVL informs the GIS of these values by a Notification (MNF) file.



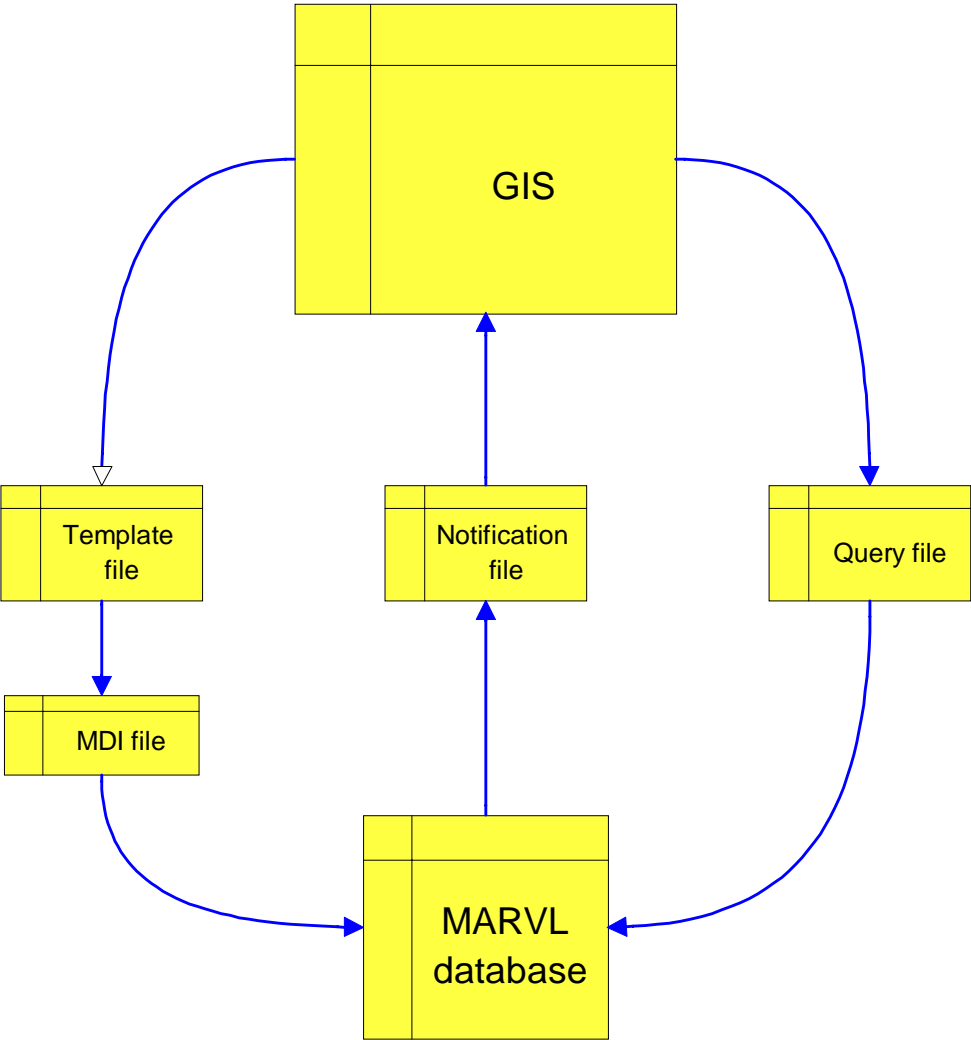
If plots are linked to a GIS as described above, is possible to:

1. select plots on the GIS,
2. use the View Designer to create new views based on this selection,
3. use the View Designer to assign function sets, validate and save the view,
4. use the Analyse module to analyse the new views.

The GIS selection is communicated to MARVL with a Query (MQF) file. Arc/Info macros to maintain and execute this link have been described by Hock (1996)

The flow of information between MARVL and GIS is shown schematically in Figure 4.

Figure 4. Flow of information between MARVL and GIS.



---

## References

- Anon (1990). MicroMARVL - versatile plantation inventory. NZ Forest Research Institute, What's new in Forest Research? Number 191
- Cochran, William G. (1977). Sampling techniques. John Wiley and Sons.
- De Vries, Pieter G. (1986). Sampling theory for forest inventory. Springer-Verlag.
- Deadman, M.W. and C.J. Goulding (1978). "A method for assessment of recoverable volume by log types." New Zealand Journal of Forestry Science **9**: 225-239.
- Demaerschalk, J.P. (1972). "Converting volume equations to compatible taper equations." Forest Science **18**: 241-245.
- Gordon, A.D. and M.E. Lawrence (1995). External stem quality assessment - MARVL. In D. Hammond Ed. 1995 Forestry Handbook. New Zealand Institute of Forestry. 190-191.
- Hock, Barbara (1996). Linking MARVL and the geographic information system Arc/Info. NZ Forest Research Institute, Unpublished Project Record Number 4996
- Husch, Bertram, Charles I. Miller and Thomas W. Beers (1982). Forest Mensuration. Wiley.
- Lawrence, M.E. (1986). The MARVL pre-harvest inventory procedure. Proceedings, "Harvesting whole trees with processing and log allocation in the forest to conventional and energy products" conference, Rotorua, Forest Research Institute, Rotorua, New Zealand.
- Muhairwe, Charles K. (1995). Taper study for *Eucalyptus* species particularly blackbutt. State Forests of NSW, Unpublished report
- NZFRI (1995). Guide to using MARVL V3. New Zealand Forest Research Institute Limited, NZ FRI Software Series Number ?
- NZFRI (1997). Guide to using MARVL data capture 3. New Zealand Forest Research Institute, Number ?
- Pnevmaticos, S.M. and S.H. Mann (1972). "Dynamic programming in tree bucking." Forest Products Journal **22**: 26-30.
- Vanclay, Jerome K. (1994). Modelling forest growth and yield. Applications to mixed tropical forests. CAB International.

## Appendix 1. List of standard formulas used by MARVL

### Notation

Symbols used in this section are listed below.

$i$	number of trees in a plot, $i = 1$ to $n$
$D_{Bi}$	DBHOB of $i$ th tree
$D_i$	DOB, $i$ th tree
$d_i$	DUB, $i$ th tree
$b_i$	double bark thickness, $D_i - d_i$
$H_i$	height of $i$ th tree
$h_i$	level above ground of a point on the stem, $i$ th tree
$l_i$	distance from the top of the tree $i$ th tree, $H_i - h_i$
$R$	$l_i / H_i$
$g_i$	basal area of $i$ th tree
$w_i$	frequency of $i$ th tree
$X$	per hectare characteristic
$x_i$	any characteristic of $i$ th tree
$A$	plot area, ha (fixed area plots)
$L$	plot length (horizontal line plots)
$F$	basal area factor (horizontal point or line sample)
$N$	Plot stocking density (stems/ha)
$G$	Plot basal area ( $m^2/ha$ )
$K$	$\pi / 40\,000$
$T_0$	Year of planting (plantation)
$T_1$	Year of measurement

Per hectare estimates

In general, the value of any per hectare stand characteristic,  $X$ , is estimated as follows:

$$\hat{X} = \sum_{i=1}^n w_i x_i$$

where  $w_i = 1 / A$  (fixed area plot)

$w_i = F / g_i$  (horizontal point sample)

$w_i = 1000 \frac{\sqrt{F}}{D_{Bi} L}$  (horizontal line samples)

Stocking density

$$\hat{N} = \sum_i w_i$$

Plot basal area

$$\hat{G} = \sum_i w_i g_i$$

In the case of point samples, this formula simplifies to:

$$\hat{G} = F n$$

Per tree estimates of the characteristic X are found by:

$$\begin{aligned} \hat{x} &= \frac{\hat{X}}{\hat{N}} \\ &= \frac{\sum_i w_i x_i}{\sum_i w_i} \end{aligned}$$

## Appendix 2. List of standard function forms.

### *Stem breakage*

Source: BREAKTAB.TXT, symbols are defined in Appendix 1

Equation 1

$$h_{bi} = b_1 + b_2 H_i + b_3 H_i^2 + b_4 s_p + b_5 s_p^2 + b_6 s_p H_i$$

Equation 2

$$h_{bi} = H_i \left[ b_1 + (1 - b_1) \left( 1 - e^{(-b_2 H_i)} \right)^{b_3} \right]$$

### *Height/age relationships*

Function types:

1 (approximate height/age curve)

2 (no growth)

3 (percentage growth)

4 (explicit height/age curve)

5 - 22 (normal height/age curve)

All currently installed models are of the form:

$$H_i = a \left[ 1 - \exp(-b(T_i - T_0)) \right]^c$$

***Height/DBHOB relationships***

Source: HTDBHTAB.TXT, symbols defined in Appendix 1.

Pettersen 1:

$$\frac{D_{Bi}}{(H_i - bh)^{0.4}} = \alpha D_{Bi} + \beta$$

Pettersen 2:

$$\frac{1}{(H_i - bh)^{0.4}} = \alpha + \beta \left( \frac{1}{D_{Bi}} \right)$$

Logarithmic:

$$\ln H_i = \alpha + \beta \left( \frac{1}{D_{Bi}} \right)$$

**Taper functions**

Source: TAPERTAB.TXT, symbols defined in Appendix 1.

Taper function 01:

$$d_i^2 = \left[ \frac{V_i}{K H_i} \right] \left[ b_1 R_i^1 + b_2 R_i^2 + b_3 R_i^3 + b_4 R_i^4 + b_5 R_i^5 + b_6 R_i^{b_7} + b_8 R_i^{b_9} \right]$$

where  $d_i = DUB$  at length  $l_i$  from tree tip

Taper function 07:

$$d_i = D_i - B_i \quad (\text{sectional measurement})$$

Taper function 08:

$$D_i^2 = D_{Bi}^2 \left[ b_c R_i^{\left( \frac{b_1}{H_i^{b_2}} \right)} \right] + \left[ \frac{b_3}{(D_{Bi} H_i)^{b_4}} R_i^{b_5} \right]$$

$$\text{where } b_c = 1 - \left[ \frac{b_3}{(D_{Bi} H_i)^{b_4}} \right] \left[ 1 - \frac{bh}{H_i} \right]^{b_5} / \left[ 1 - \frac{bh}{H_i} \right]^{\frac{b_1}{H_i^{b_2}}}$$

$$d_i^2 = D_i^2 \left[ b_7 + b_8 R_i + b_9 R_i^{b_{10}} \right]$$

Taper function 09:

$$D_i^2 = \left[ \frac{D_{Bi} H_i}{H_i - bh} \right]^2 \left[ b_c R_i^{b_1} + b_2 R_i^{b_3} + b_4 R_i^{b_5} \right]$$

$$\text{where } b_c = \left[ 1 - \frac{bh}{H_i} \right]^2 - b_2 \left[ 1 - \frac{bh}{H_i} \right]^{b_3} - b_4 \left[ 1 - \frac{bh}{H_i} \right]^{b_5} / \left[ 1 - \frac{bh}{H_i} \right]^{b_1}$$

$$d_i^2 = D_i^2 \left[ b_7 + b_8 R_i + b_9 R_i^{b_{10}} \right]$$

Taper function 10:

$$D_i^2 = D_{Bi}^2 \left[ b_c R_i^{b_1} + b_2 R_i^{b_3} \right]$$

$$\text{where } b_c = \left[ 1 - b_2 \left( 1 - \frac{bh}{H_i} \right)^{b_3} \right] / \left[ 1 - \frac{bh}{H_i} \right]^{b_1}$$



$$d_i^2 = D_i^2 \left[ b_7 + b_8 \frac{D_i}{D_{Bi}} \right]^2$$

Taper function 11:

$$D_i^2 = \left[ \frac{D_{Bi} H_i}{H_i - b h} \right]^2 \left[ b_1 + b_2 R_i^{(b_3 D_{Bi})} \right]$$

$$d_i^2 = D_i^2 \left[ b_7 + b_8 R_i \right]$$

Taper function 13:

2-segment model. Quadratic (upper) and hyperbolic (lower) functions, with a variable joint point. (Equation?)

Taper function 14:

$$d = b_1 D_{Bi}^{b_2} b_3^{D_{Bi}} \left[ 1 - \sqrt{\frac{h_i}{H_i}} \right] \left[ b_4 \left( \frac{h_i}{H_i} \right) + b_5 \left( \frac{h_i}{H_i} \right)^2 + b_6 \left( \frac{h_i}{H_i} \right)^3 + b_7 \left( \frac{H_i}{h_i} \right) + b_8 D_{Bi} + b_9 H_i + b_{10} \left( \frac{D_{Bi}}{H_i} \right) \right]$$

Taper function 15:

$$D_i^2 = b_1 D_{Bi}^{b_4} + b_2 e^{(-b_3 h_i^{0.25})}$$

$$\text{where } b_2 = \left[ D_{Bi}^2 - b_1 D_{Bi}^{b_5} \right] / e^{(-b_3 b h^{0.25})}$$

$$d_i^2 = b_7 + b_8 D_i^2$$

**Volume functions**

Source: VOLUMTAB.TXT, symbols defined in Appendix 1.

Volume table 01:

$$V_i = D_{Bi}^{b_1} \left[ \frac{H_i^2}{H_i - bh} \right]^{b_2} e^{b_3} + b_4$$

Volume table 02:

$$V_i = b_1 + b_2 D_{Bi}^2 \frac{H_i}{10000} + b_3 \frac{D_{Bi}^2}{10000} + b_4 H_i + b_5 \frac{D_{Bi} H_i}{10000}$$

Volume table 03:

$$\log(V_i) = b_1 \log(D_{Bi}) + b_2 \log(H_i) + b_3 \frac{D_{Bi}^2}{10000} + b_4$$

Volume table 04:

$$V_i = b_1 D_{Bi}^{b_2} H^{b_3}$$

Volume table 05:

$$V_i = \left[ b_1 D_{Bi}^2 \frac{H_i^2}{H_i - bh} + b_2 \right] / 1000$$

Volume table 06:

$$V_i = b_1 (D_{Bi}^2 H_i)^{0.9} / 10000$$

Volume table 07:

$$V_i = \sum (\text{sectional volumes})$$

Volume tables 08 - 11

Integral of taper function types 08 - 11

Volume table 12:

$$V_i = [b_1 H_i + b_2 D_{Bi}^2 H_i + b_3 H_i^2 + b_4 D_{Bi}^2 H_i^2] / [H - bh]$$

Volume tables 13, 14, 15

Integral of taper function types 13, 14, 15

### Appendix 3. MARVL variables

Lists of variables available for the definition of report sections follow. Variables in any one report section may be selected from only 1 list. Variable characteristics are:

If there is a V in column 1, this variable can appear in a “Variables=” line.

If there is a C in column 2, this variable can appear in a “Classvariable=” line.

If there is a B in column 3, this variable can appear in a ByVariable=” line.

If there is an E in column 4, this variable can have a +ERROR after it on a “Variables=” line

If there is a P in column 5, this variable can have a +PERCENT.

#### *Log level variables*

Variable name	1	2	3	4	5
LOG_SPECIES		C			
LOG_VALUE	V	C		E	P
LOG_SED	V	C		E	P
LOG_SED_MIN	V				
LOG_SED_MAX	V				
LOG_LED	V	C		E	P
LOG_LED_MIN	V				
LOG_LED_MAX	V				
LOG_LENGTH	V	C		E	P
LOG_LENGTH_MIN	V				
LOG_LENGTH_MAX	V				
LOG_VOLUME	V	C		E	P
LOG_SIZE	V				
LOG_COUNT	V				P
LOG_TYPE	V	C	B		
LOG_TYPE_USER	V	C	B		
LOG_KIND	V	C	B		
LOG_GROUP		C	B		
LOG_TAPER	V			E	P

*Piece level variables*

Variable name	1	2	3	4	5
PIECE_SPECIES		C	B		
PIECE_VALUE	V	C		E	P
PIECE_SED	V	C		E	P
PIECE_SED_MIN	V				
PIECE_SED_MAX	V				
PIECE_LED	V	C		E	P
PIECE_LED_MIN	V				
PIECE_LED_MAX	V				
PIECE_LENGTH	V	C		E	P
PIECE_LENGTH_MIN	V				
PIECE_LENGTH_MAX	V				
PIECE_VOLUME	V	C		E	P
PIECE_SIZE	V				
PIECE_COUNT	V				P
PIECE_NUM_LOGS	V				P
PIECE_IS_EXTRACTED	V	C	B		P

*Tree level variables*

Variable name	1	2	3	4	5
TREE_SPECIES		C	B		
TREE_DBH	V	C		E	P
TREE_VOLUME_TOTAL	V	C		E	P
TREE_VOLUME_RECOV	V	C		E	P
TREE_VOLUME_EXTR	V	C		E	P
TREE_VOLUME_CROWN	V	C		E	P
TREE_VOLUME_STUMP	V	C		E	P
TREE_VOLUME_BREAK	V	C		E	P
TREE_VALUE	V	C		E	P
TREE_IS_EXTRACTED	V	C	B		P
TREE_IS_ALIVE	V	C	B		P
TREE_HEIGHT_TOP	V	C		E	P
TREE_HEIGHT_BREAK	V	C		E	P
TREE_COUNT	V				P
TREE_BA	V	C		E	P
TREE_IS_MALFORMED	V	C	B		P
TREE_IS_BROKEN	V	C	B		P
TREE_SIZE	V				
TREE_SIZE_EXTR	V				
TREE_SIZE_RECOV	V				

*Plot level variables*

Variable name	1	2	3	4	5
PLOT_MEAN_DBH	V			E	P
PLOT_DBH_MEAN_TOP	V			E	P
PLOT_HEIGHT_MEAN_TOP	V			E	P
PLOT_HEIGHT_MEAN	V			E	P
PLOT_VOLUME_TOTAL	V			E	P
PLOT_VOLUME_RECOV	V			E	P
PLOT_VOLUME_EXTR	V			E	P
PLOT_VOLUME_CROWN	V			E	P
PLOT_VOLUME_STUMP	V			E	P
PLOT_VOLUME_BREAK	V			E	P
PLOT_BA	V			E	P
PLOT_SPH	V			E	P

---

## Appendix 4 Example report section definitions

```
[StandingResource]
Name=Standing Resource
ClassVariable=TREE_IS_ALIVE
Variables=TREE_VOLUME_TOTAL +ERROR TREE_COUNT +PERCENT TREE_BA +ERROR
+PERCENT TREE_DBH
Levels=PLOT STRATUM POPULATION

[StandardResource]
Variables=PLOT_HEIGHT_LIVE PLOT_DBH_MEAN_TOP PLOT_HEIGHT_MEAN_TOP
ShowTitles=TRUE
ShowTotals=FALSE
Levels=POPULATION

[VolumeAnalysis]
Name=Volume Analysis
Variables=TREE_VOLUME_TOTAL +ERROR TREE_VOLUME_RECOV +ERROR TREE_COUNT
TREE_SIZE
ClassVariable=TREE_IS_EXTRACTED
Levels=PLOT STRATUM POPULATION
ShowTotals=FALSE
ShowTitles=TRUE

[LiveStandingTrees]
Name=Live Standing Trees
Variables=TREE_VOLUME_TOTAL +ERROR TREE_COUNT TREE_SIZE
ClassVariable=TREE_IS_RECOVERED
Levels=PLOT STRATUM POPULATION
ShowTotals=TRUE
ShowTitles=TRUE

[CutoverResidue]
Name=Cutover Residue
Variables=LOG_VOLUME +ERROR
ClassVariable=LOG_KIND
ShowTotals=FALSE
ShowTitles=TRUE
FirstClass=0
NumberOfClasses=2
Levels=PLOT STRATUM POPULATION

[ExtractableStems]
Name=Extractable Stems (stump and breakage removed)
Variables=TREE_VOLUME_EXTR +ERROR TREE_COUNT TREE_SIZE_EXTR
ClassVariable=TREE_IS_RECOVERED
Levels=PLOT STRATUM POPULATION
ShowTotals=TRUE
ShowTitles=TRUE

[CuttingWaste]
Variables=LOG_VOLUME_EXTR +ERROR
ClassVariable=LOG_KIND
ShowTotals=FALSE
ShowTitles=FALSE
FirstClass=2
NumberOfClasses=1
Levels=PLOT STRATUM POPULATION
```

---

```
[ProductAnalysis]
Name=Product Analysis
Variables=LOG_VOLUME +ERROR +PERCENT LOG_VALUE +PERCENT LOG_COUNT
LOG_SIZE
ClassVariable=LOG_TYPE_USER
Levels=STRATUM POPULATION
ShowTotals=TRUE
ShowTitles=TRUE

[StockTable]
Name=Stock Table
Variables=LOG_COUNT LOG_VOLUME +ERROR +PERCENT LOG_VALUE
ByVariable=LOG_TYPE_USER
ClassVariable=LOG_SED
Levels=POPULATION
ShowTitles=TRUE
NumberOfClasses=25
ClassWidth=40
FirstClass=0

[StandTable]
Name=Stand Table
Variables=TREE_COUNT TREE_IS_MALFORMED TREE_IS_EXTRACTED TREE_HEIGHT_TOP
TREE_HEIGHT_BREAK TREE_VOLUME_TOTAL TREE_VOLUME_RECOV
ClassVariable=TREE_DBH
Levels=POPULATION
ShowTitles=TRUE
NumberOfClasses=25
ClassWidth=40
FirstClass=0

[LengthDist]
Name=Log Length Distribution
Variables=LOG_VOLUME LOG_VALUE LOG_COUNT LOG_SED
ByVariable=LOG_TYPE
ClassVariable=LOG_LENGTH
Levels=PLOT STRATUM POPULATION
ShowTitles=TRUE
NumberOfClasses=40
ClassWidth=0.25
FirstClass=1

[PieceVolume]
Name=Piece Volume Distribution
Variables=PIECE_COUNT PIECE_SED PIECE_LED PIECE_LENGTH
ByVariable=PIECE_IS_EXTRACTED
ClassVariable=PIECE_VOLUME
ShowTitles=TRUE
Levels=PLOT STRATUM POPULATION

[YieldTable]
Name=Yield Table
Variables=LOG_VOLUME +ERROR +PERCENT LOG_VALUE +PERCENT LOG_COUNT
ClassVariable=LOG_TYPE
ShowTitles=TRUE
Levels=PLOT STRATUM POPULATION

[LogTrace]
Name=Log Trace
ShowTitles=TRUE
Levels=PLOT STRATUM POPULATION
```



---

```
[StratumDescription]
Levels=STRATUM POPULATION

[CentreDiameterUnderBark]
Levels=POPULATION

[DominanceDistribution]
Name=Dominance distribution
Variables=TREE_COUNT
ClassVariable=TREE_DBH
NumberOfClasses=40
ClassWidth=50
FirstClass=200
ByVariable="Dominance"
```

## Appendix 5. MARVL database schema

### ----- Import -----

```
Extra :      Extra data for MicroMARVL v2.x import
Inventory *    Text(20)    Inventory Name
Stratum  *    Int(10)     Stratum Number
PlotNum  *    Int(5)      Plot Number within Forest/Cpt/Stand/Meas
EstabYr   Int(4)      Established Year: when plot was planted
MeasYear  Int(4)      Year plot was measured
MeasMnth  Int(2)      Month plot was measured
Species   Text(5)      Tree Species
GISLink   Long(10)     Connection to unique Plot ID in GIS
```

```
ExtrDict  :      Extra Data: Inventory --> Dictionary link
Inventory *    Text(20)    Inventory Name
Dictnary  Text(20)    Dictionary name
```

```
ExtrFunc  :      Extra Data: Inventory --> Function Set link
Inventory *    Text(20)    Inventory Name
Stratum  *    Int(10)     Stratum Number
FuncName  Text(20)    Function Set Name
```

### ----- Misc -----

```
Forest    :      Valid Forest codes
Forest    *    Text(4)     Forest Name
LongName   Text(30)    Long name
Owner      Text(40)    Owner of forest
Region     Text(2)     Location of forest

Species    :      List of all legal tree species codes
Species    *    Text(5)    Tree Species
Botanic    Text(25)    Botanical name
Common     Text(25)    Common name
```

### ----- Function -----

```
Function   :      Set of volume, taper, breakage, growth, height
equations
FuncName   *    Text(20)    Function Set Name
LastUser   Text(10)    Name of user who last modified this
DateUser   Timestamp   Date & Time this was last changed
IsLocked   Int(1)      Function Set cannot be changed
GmodType   Text(1)     Growth model type
GmodNum    Int(4)      Growth model number
HmodType   Text(1)     Height model type
HmodNum    Int(4)      Height model number
Adjust     Int(4)      Monthly Growth adjustment table
HtType     Text(1)     Height/Diameter type L=log P=Pett1 Q=Pett2
T=Table
HtTable    Int(4)      Height/Diameter Curve table (HtType='T')
Descript   Memo   Description of Function Set

FunGroup   :      Groups of table / model numbers in a function set
FuncName   *    Text(20)    Function Set Name
FuncGrp    *    Text(20)    Function Group Name
Volume     Int(4)      Tree Volume table number
Taper      Int(4)      Tree Taper table number
Breakage   Int(4)      Tree breakage table number
```

---

```

FunSpecy      :      Which species belong to which function group
FuncName      *      Text(20)      Function Set Name
FuncGrp       *      Text(20)      Function Group Name
Species       *      Text(5)       Tree Species

----- Strategy -----
Strategy      :      Cutting Strategy definition
CutStrat      *      Text(20)      Cutting Strategy name
LastUser      Text(10)      Name of user who last modified this
DateUser      Timestamp     Date & Time this was last changed
IsLocked      Int(1)        Strategy cannot be changed
RoundLen      Real(4,1)     Round off length
StumpHt       Real(4,1)     Stump height (metres)
CutCost       Real(9,7)     Cost of making a cut
Descript      Memo  Description of Strategy

LogType       :      Log type's value and acceptable end diameters
CutStrat      *      Text(20)      Cutting Strategy name
LogType       *      Text(20)      Log Type name
LogKind       *      Int(2)        Stump, Waste, Above-Break, User-Defined
Dollars       Real(8,2)     Log dollar value
MinSED        Real(4,0)     Minimum small end diameter
MaxSED        Real(4,0)     Maximum small end diameter
MaxLED        Real(4,0)     Maximum large end diameter
Descript      Memo  Description of Log Type

LogDict       :      Valid quality codes for each log type in a cutting
strategy
CutStrat      *      Text(20)      Cutting Strategy name
LogType       *      Text(20)      Log Type name
Dictionary    *      Text(20)      Dictionary name
QCodes        Text(26)      Valid qualities
MinLeng       Real(4,1)     Minimum length of MinQCode
MinQCode      Text(26)      Must have at least MinLeng of these
qualities (composite log types)

LogLeng       :      Valid length ranges for Log Types in a cutting strategy
CutStrat      *      Text(20)      Cutting Strategy name
LogType       *      Text(20)      Log Type name
MinLeng       *      Real(4,1)     Minimum log type length
MaxLeng       *      Real(4,1)     Maximum log type length

LogSpecy      :      Log type's allowed species
CutStrat      *      Text(20)      Cutting Strategy name
LogType       *      Text(20)      Log Type name
Species       *      Text(5)       Tree Species

LogGroup      :      LogGroup: definition
CutStrat      *      Text(20)      Cutting Strategy name
LogGroup      *      Text(20)      Log Group name
AvgSED        Real(4,0)     Minimum average small end diameter
MinSED        Real(4,0)     Minimum allowable log type diameter
SpecCut       Int(1)        Cut to SED spec during analysis

LogGrpTy      :      The log types in a log group
CutStrat      *      Text(20)      Cutting Strategy name
LogGroup      *      Text(20)      Log Group name
LogType       *      Text(20)      Log Type name

```

## ----- Plan -----

LPlan : Group of Views  
 LPlan \* Text(20) Groups a set of inventories  
 LastUser Text(10) Name of user who last modified this  
 DateUser Timestamp Date & Time this was last changed  
 IsLocked Int(1) True if this plan cannot be changed  
 Descript Memo Description of Plan

LPlanInv : List of Views in the Plan  
 LPlan \* Text(20) Groups a set of inventories  
 Inventory \* Text(20) Inventory Name

## ----- View -----

Inventory : View / Inventory definition  
 Inventory \* Text(20) Inventory Name  
 LastUser Text(10) Name of user who last modified this  
 DateUser Timestamp Date & Time this was last changed  
 IsLocked Int(1) True if this view is locked  
 IsDesign Int(1) True if this is a design inventory  
 IsValid Int(1) True if this view can be analysed  
 Descript Memo Description of View

StrPlot : Link view strata to plot keys  
 Inventory \* Text(20) Inventory Name  
 Stratum \* Int(10) Stratum Number  
 PlotKey \* Long(10) Unique plot key  
 UsHeight Int(1) True if height trees from this plot are  
 added to regression  
 PlotUse Int(1) 0 = unused, 1 = Primary, 2 = Secondary plot

Stratum : Stratifies plots in a View  
 Inventory \* Text(20) Inventory Name  
 Stratum \* Int(10) Stratum Number  
 Area Real(6,2) Stratum area  
 FuncName Text(20) Function Set Name

## ----- Plot -----

Plot : Plot data  
 PlotKey \* Long(10) Unique plot key  
 LastUser Text(10) Name of user who last modified this  
 DateUser Timestamp Date & Time this was last changed  
 PlotNum Int(5) Plot Number within Forest/Cpt/Stand/Meas  
 Inventory Text(20) Inventory Name  
 Dictnary Text(20) Dictionary name  
 EstYear Int(4) Year plot was planted  
 MeasYear Int(4) Year plot was measured  
 MeasMnth Int(2) Month plot was measured  
 TreeCoun Int(3) Number of trees in plot (live and dead)  
 Slope Int(2) Slope value  
 PlotType Int(1) bounded, angle gauge, count, horiz-line,  
 LIS  
 Area Real(7,5) Plot area  
 Length Real(6,2) Plot length  
 Baf Real(5,2) Basal Area factor  
 BArea Real(5,1) Live Basal area  
 Stocking Real(5,0) Live Stocking  
 MTopDBH Real(4,0) Mean top diameter  
 GISLink Long(10) Connection to unique Plot ID in GIS  
 Descript Memo Description of Plot

---

```

PlotKey      :      Contains next value for a unique plot key
PlotKey      *      Long(10)      Unique plot key

Tree :      Tree data (plus stem description)
PlotKey      *      Long(10)      Unique plot key
TreeKey      *      Long(10)      Tree number
TreeWgt      Real(6,1)      Stocking represented by this tree
Dbh          Real(4,0)      Diameter at breast height
Height       Real(5,2)      Height of tallest leader
IsLive       Int(1)         True if tree is alive, False if dead
Species      Text(5)        Tree Species
StemDesc     Memo          Collapsed Stem Description

----- UserVar -----
UserVar      :      User Variable: type, measurement units & valid value
range
UserVar      *      Text(20)      User Variable name
LastUser     Text(10)      Name of user who last modified this
DateUser     Timestamp     Date & Time this was last changed
Minimum      Real(20,5)     Minimum Value
Maximum      Real(20,5)     Maximum Value
Label        Text(20)      Measurement unit
VarType      Int(1)         N(numeric) D(discrete) O(ordinal)
Binding      Int(1)         T(tree), P(plot), E(every plot)
Width        Int(2)         Printing width (including decimal places)
DPlaces      Int(2)         Number of decimal places to print
Descript     Memo          Description of User Variable

TreeUser     :      User Variable values for Trees
PlotKey      *      Long(10)      Unique plot key
TreeKey      *      Long(10)      Tree number
UserVar      *      Text(20)      User Variable name
VarValue     Real(20,5)     Value of Variable

PlotUser     :      Plot user variable values
PlotKey      *      Long(10)      Unique plot key
UserVar      *      Text(20)      User Variable name
VarValue     Real(20,5)     Value of Variable

PlotVar      :      Plot user variables (every plot has this variable)
PlotKey      *      Long(10)      Unique plot key

VarLabel     :      Ordinal user variable value labels
UserVar      *      Text(20)      User Variable name
OrdValue     *      Int(10)      User Variable ordinal value
Label        Text(20)      Label Text

VarCalc      :      Methods describing how to calculate user variable
values
UserVar      *      Text(20)      User Variable name
LastUser     Text(10)      Name of user who last modified this
DateUser     Timestamp     Date & Time this was last changed
Method       Int(10)       Calculation method number

VarDict      :      User Variable to Method to Dictionary link
UserVar      *      Text(20)      User Variable name
Dictnary     *      Text(20)      Dictionary name
QCodes       Text(26)      Valid qualities from Dictionary

```

## ----- Dictnary -----

```
Dictnary      :      Quality Code Dictionary definition
Dictnary      *      Text(20)      Dictionary name
LastUser      Text(10)      Name of user who last modified this
DateUser      Timestamp     Date & Time this was last changed
IsLocked      Int(1)        Dictionary cannot be changed
Descript      Memo      Description of Dictionary

Dictcode      :      List of valid codes and descriptions in a Dictionary
Dictnary      *      Text(20)      Dictionary name
QCode *       Text(1)      Quality Code
Descript      Memo      Description of Quality Code
```