

**Database documentation: recruit**

**K. A. Mackay**

NIWA Internal Report  
2001

Revised on 30 January 2001

## Contents

<b>1</b>	<b>Database documentation series</b> .....	<b>3</b>
<b>2</b>	<b>Recruitment surveys</b> .....	<b>3</b>
<b>3</b>	<b>Data structures</b> .....	<b>4</b>
3.1	TABLE RELATIONSHIPS .....	4
3.2	DATABASE DESIGN .....	7
<b>4</b>	<b>Table summaries</b> .....	<b>8</b>
<b>5</b>	<b>recruit tables</b> .....	<b>9</b>
5.1	TABLE 1: T_SITE .....	9
5.2	TABLE 2: T_SITE_CODES .....	10
5.3	TABLE 3: T_LGTH.....	11
<b>6</b>	<b>recruit business rules</b> .....	<b>12</b>
6.1	INTRODUCTION TO BUSINESS RULES.....	12
6.2	SUMMARY OF RULES .....	13
<b>7</b>	<b>Acknowledgments</b> .....	<b>15</b>
	<b>Appendix</b> .....	<b>16</b>

## List of Figures

Figure 1:	Entity Relationship Diagram (ERD) for the recruit database.....	6
Figure 2:	Map of the coverage of the NZMS 260 map series for the North Island. ....	16
Figure 3:	Map of the coverage of the NZMS 260 map series for the South Island. ....	17

## 1 Database documentation series

The National Institute of Water and Atmospheric Research (NIWA) currently carries out the role of Data Manager and Custodian for the fisheries research data owned by the Ministry of Fisheries.

The Ministry of Fisheries data set incorporates historic research data, data collected more recently by MAF Fisheries prior to the split in 1995 of Policy to the Ministry of Fisheries and research to NIWA, and currently data collected by NIWA and other agencies for the Ministry of Fisheries.

This document provides a brief description of the recruitment database **recruit**, and is a part of the database documentation series produced by NIWA. It supersedes the previous documentation by Mackay (1998)<sup>1</sup> on this database.

All documents in this series include an introduction to the database design, a description of the main data structures accompanied by an Entity Relationship Diagram (ERD), and a listing of all the main tables. The ERD graphically shows the relationships between the tables in **recruit** and their relationships to other databases.

This document is intended as a guide for users and administrators of the **recruit** database.

Access to this database is restricted to specific nominated personnel as specified in the current Schedule 6 of the Data Management contract between the Ministry of Fisheries and NIWA. Any requests for data should in the first instance be directed to the Ministry of Fisheries.

## 2 Recruitment surveys

The abundance of several of New Zealand's important commercial species is likely to be recruitment driven. So the development of a recruitment index and its incorporation into a stock assessment model could reduce the uncertainty in the estimates of biomass and sustainable yields and might lead to a method of predicting fluctuations in abundance.

Because of this, a juvenile kahawai (*Arripis trutta*) recruitment index feasibility study was instigated<sup>2</sup>. Juvenile kahawai were sampled in three areas of New Zealand between November 1996 and September 1997 to test the feasibility of deriving an annual recruitment index for this species.

Sites were chosen in each area based on ease of access, shallow beach gradient to allow deployment of the net, no obstructions (rocks, mangroves), and a fishable area ~100m long.

---

<sup>1</sup> Mackay, K.A 1998: Database documentation: recruit. *NIWA Internal Report No. 35*. 10p.

<sup>2</sup> Gerring, P. K. & Bradford, E. 1998: Juvenile kahawai recruitment index feasibility study. *NIWA Technical Report 36*. 38p.

The net used for sampling was a beach-seine modified to catch 0+ aged kahawai. The net was 20m long by 2m deep with a 9mm mesh. Sites were sampled by beach seining with the net dragged parallel to the shore for ~100m. The net was then dragged ashore and the catch sorted by species. Juvenile kahawai were counted and measured to the nearest 1mm (fork length), while other species were only counted. For large catches of kahawai, a random sample of at least 50 juveniles was measured. All sampling was done within 2 hours either side of spring high tide.

### 3 Data structures

#### 3.1 Table relationships

This database contains several tables. The ERD for **recruit** (Figure 1) shows the logical structure<sup>3</sup> of the database and its entities (each entity is implemented as a database *table*) and relationships between these tables and tables in other databases. This schema is valid regardless of the database system chosen, and it can remain correct even if the Database Management System (DBMS) is changed. Each table represents an object, event, or concept in the real world that is chosen to be represented in the database. Each *attribute* of a table is a defining property or quality of the table. All of the table's attributes are shown in the ERD. The underlined attributes represent the table's primary key<sup>4</sup>.

Note that Figure 1 shows the main tables only. Note that most tables contain foreign keys<sup>5</sup>. These foreign keys define the relationships between the tables in **recruit**.

The **recruit** database is implemented as a relational database; i.e., each table is a special case of the mathematical construct known as a *relation* and hence elementary relation theory is used to deal with the data within tables and the relationships between them. There are three types of relationships possible between tables, but only one exists in **recruit**: one-to-many<sup>6</sup>. This is shown in the ERD by connecting a single line (indicating 'many') from the child table (e.g., *t\_lgth*) to the parent table (e.g., *t\_site*) with an arrow-head (indicating 'one') pointing to the parent. For example, consider the relationship between the tables *t\_site* (the parent table) and *t\_lgth* (the child table). Any one sample in *t\_site* can generate one or more length frequency records in *t\_lgth*, but any one length frequency record can only come from one sample. Note that the word 'many' applies to the possible number of records in one table that one record in another table is associated with. For a given instance, there might be zero, one, two, or more associated records, but if it is ever possible to have more than one, we use the word 'many' to describe the association.

Every relationship has a mandatory or optional aspect to it. That is, if a relationship is mandatory, then it has to occur at least once, while an optional relationship might not occur at all. For example, in Figure 1, consider again that relationship between the table *t\_site* and its child table *t\_lgth*.

---

<sup>3</sup> Also known as a database *schema*.

<sup>4</sup> A primary key is an attribute or a combination of attributes that contains an unique value to identify that record.

<sup>5</sup> A foreign key is an attribute or a combination of attributes that is a primary key in another table.

<sup>6</sup> A one-to-many relationship is where one record (the *parent*) in a table relates to one or many records (the *child*) in another table; e.g., one sample in *t\_site* can have many length frequency records in *t\_lgth* but one length frequency records can only come from one sample site.

The symbol ‘O’ by the child *t\_lgth* means that a sample record may have zero or many length frequency records, while the bar by the parent *t\_site* means that for every length frequency record there must be a matching sample record.

These relationships are enforced in the database by the use of referential constraints<sup>7</sup>. Constraints do not allow *orphans* to exist in any table; i.e., where a child record exists without a related parent record. This may happen when: a parent record is deleted; the parent record is altered so the relationship is lost; or a child record is entered without a parent record. All constraints in **recruit** prevent the latter from occurring. Constraints are shown in the table listings by the following format:

```
Referential:      constraint name (attribute[, attribute]) |INSERT|
                                                           |DELETE|
                   parent table (attribute[, attribute])
```

For example, consider the following constraint found in the table *t\_lgth*:

```
Referential:      invalid id (id) INSERT t_site (id)
```

This means that the value of the attribute *id* in the current record must already exist in the parent table *t\_site* or the record will be rejected and the following message will be displayed:

```
*** User Error: insert constraint 'invalid id' violation
```

For tables residing in external databases, the parent table name will be prefixed by the name of the database.

Section 5 details a listing of all the **recruit** tables as implemented by the EMPRESS DBMS. These table show that a table’s primary key has an unique index on it. Primary keys are generally listed using the format:

```
Indices:      UNIQUE index_name ON (attribute [, attributes])
```

where the attribute(s) make up the primary key (the key attributes), and the index name is the primary key name. Note that the typographical convention for the above format is that square brackets [] may contain more than one item or none at all.

The unique index prevents records with duplicate key values from being inserted into the table; e.g., a new sample being inserted with an existing id number, and hence ensures that every record can be uniquely identified.

The **recruit** database is implemented as a relational database. That is, each table is a special case of the mathematical construct known as a *relation* and hence elementary relation theory is used to deal with the data within tables and the relationships between them. All relationships in **recruit** are of the type *one-to-many*<sup>8</sup>.

---

<sup>7</sup> Also known as integrity checks.

<sup>8</sup> A one-to-many relationship is where one record (the *parent*) in a table relates to one or many records (the *child*) in another table; e.g., one sample in *t\_site* can have many length frequency records in *t\_lgth* but any one length frequency record can only come from one sample.

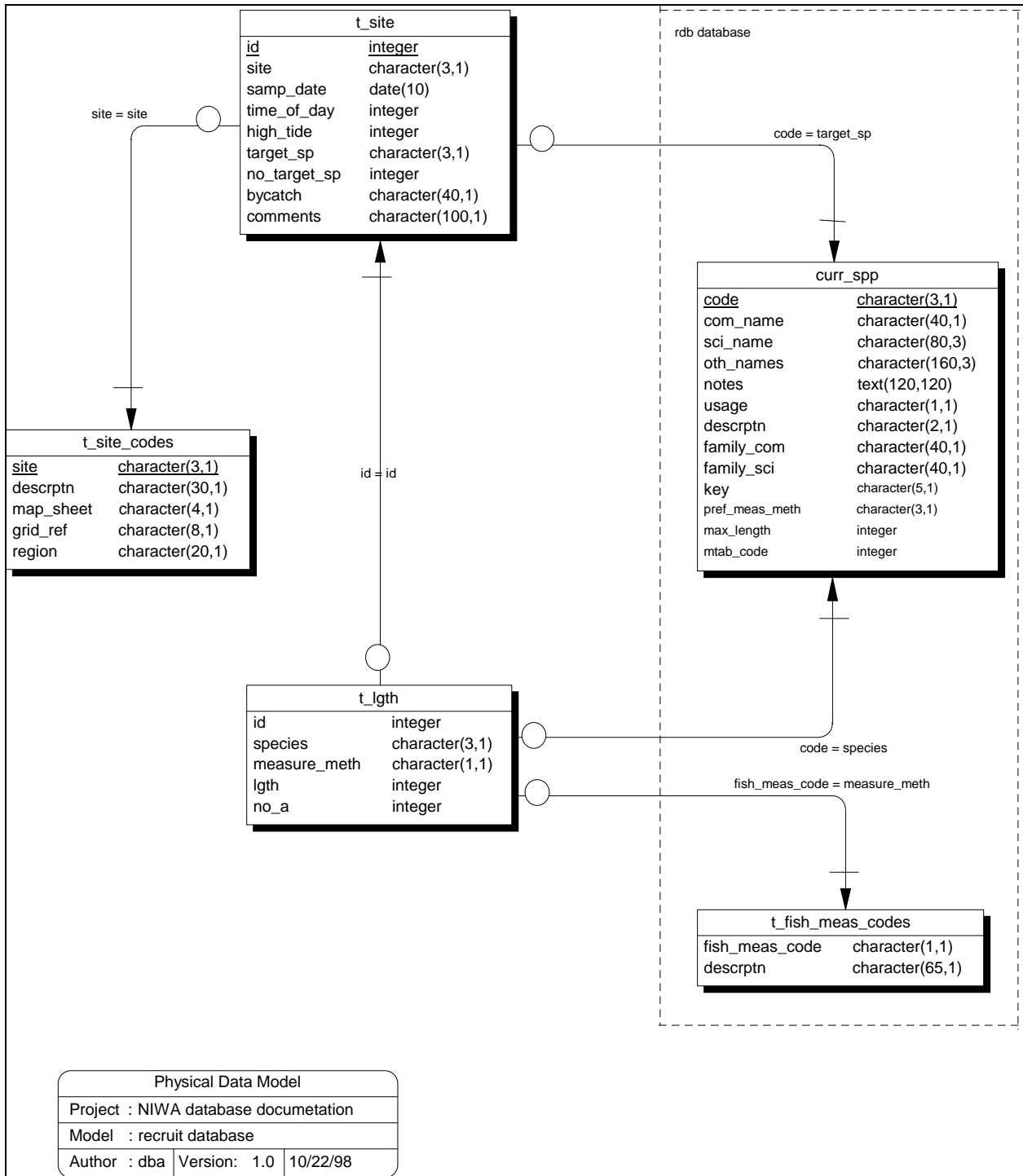


Figure 1: Entity Relationship Diagram (ERD) for the recruit database.

All tables are indexed. That is, attributes that are most likely to be used for searching, such as *id*, have like values linked together to optimise search times. Such indices are shown in the table listings (Section 5) by the following syntax:

**Indices:**      NORMAL (2, 15) index\_name ON (attribute{, attributes})

Note that indices may be *simple*, pointing to just one attribute, or *composite*, pointing to more than one attribute. The numbers ‘...(2, 15)...’ are EMPRESS default values relating to the amount of space allocated to index storage.

### 3.2 Database design

The details for each sample undertaken are recorded in the table *t\_site* (Table 1). Each sample is allocated an identification number, *id*, which is unique for every sample. The site sampled is recorded as a 3-character code and stored in the attribute *site*. Codes are used, rather than full spelling of the site names, to prevent problems caused by spelling mistakes and case differences. The attribute *site* is also a foreign key to the table *t\_site\_codes*. The actual catch of the target species is recorded in the attribute *no\_target\_sp*, with the target species code recorded in the attribute *target\_sp*. Any bycatch is recorded in the attribute *bycatch* in a semi-descriptive text format, usually describing bycatch species by their 3-character species codes. For example, the value of *bycatch* of “22 YEM 1 PUF 2 PIL” means 22 yellow-eyed mullet, 1 pufferfish, and 2 pilchards. Other information recorded in this table includes sampling date and time, and the time of the nearest high tide.

Details for individual sites used during sampling are stored in the table *t\_site\_codes* (Table 2). Each record in this table is uniquely identified by a 3-character code in the attribute *site*. Each site record contains a full description of the site and the general geographic region, and is accurately located by a NZMS 260 map number and grid reference.

Length frequencies of fish caught are stored in the table *t\_lgth* (Table 3). Each record contains: *no\_a*, the number of fish (frequency) caught at a length class; *lgth*, the length class (mm); and the 1-character code *measure\_meth* to identify the method used to measure the fish (e.g., total length, fork length, etc).

## 4 Table summaries

The following is a listing and brief outline of the tables contained in **recruit**:

1. **t\_site** : contains details for each sample taken at a site.
2. **t\_site\_codes** : contains descriptions and NZMS 260 map references to the sites used for sampling.
3. **t\_lgth** : contains length frequency data of the major species caught during a sample.



## 5 recruit tables

The following are listings of the tables in the **recruit** database, including attribute names, data types (and any range restrictions), and comments.

### 5.1 Table 1: t\_site

**Comment:** Details of samples at recruitment survey sites.

Attributes	Data Type	Null?	Comment
id	integer	No	Sequential unique number to identify each sample.
site	character(3,1)	No	3 character code for survey site, refer t_site_codes.
samp_date	date(4)		Date of sample.
time_of_day	integer		Time (24hr, NZST) of sample.
high_tide	integer		Time (24hr, NZST) of nearest high tide to the sample.
target_sp	character(3,1)		3 character code for target species, refer rdb:curr_spp.
no_target_sp	integer		Number of target species caught.
bycatch	character(40,1)		Description of any bycatch caught.
comments	character(100,1)		General comments about the sample.

**Creator:** dba

**Referential:** invalid site code (site) INSERT t\_site\_codes (site)  
invalid target species (target\_sp) INSERT rdb : curr\_spp  
(code)

**Indices:** UNIQUE t\_site\_PK ON (id)

## 5.2 Table 2: t\_site\_codes

**Comment:** Descriptions and NZMS 260 grid references for site codes.

Attributes	Data Type	Null?	Comment
site	character(3,1)	No	3 character code for survey site.
descrptn	character(30,1)		Description of site code.
map_sheet	character(4,1)		NZMS 260 map sheet number.
grid_ref	character(8,1)		NZMS 260 map sheet grid reference for the site.
region	character(20,1)		General geographic region.

**Creator:** dba

**Indices:** UNIQUE site\_codes\_pk ON (site)

### 5.3 Table 3: t\_lgth

**Comment:** Length frequency data from samples.

Attributes	Data Type	Null?	Comment
id	integer	No	Sequential unique number to identify each sample.
species	character(3,1)	No	3 character code for species, refer rdb:curr_spp.
measure_meth	character(1,1)		Code of method used to measure fish lengths, refer rdb:t_fish_meas_codes.
lgth	integer		Fish length (mm).
no_a	integer		Number of species at this length.

**Creator:** dba

**Referential:** invalid id (id) INSERT t\_site (id)  
invalid species (species) INSERT rdb : curr\_spp  
(code)  
invalid meas method (measure\_meth) INSERT  
rdb : t\_fish\_meas\_codes (fish\_meas\_code)

**Indices:** NORMAL (2, 15) LGTH\_FK ON (id)

## 6 recruit business rules

### 6.1 Introduction to business rules

The following are a list of business rules applying to the **recruit** database. A business rule is a written statement specifying what the information system (i.e., any system that is designed to handle market sampling data) must do or how it must be structured.

There are three recognised types of business rules:

<b>Fact</b>	Certainty or an existence in the information system.
<b>Formula</b>	Calculation employed in the information system.
<b>Validation</b>	Constraint on a value in the information system.

Fact rules are shown on the ERD by the cardinality (e.g., one-to-many) of table relationships. Formula and Validation rules are implemented by referential constraints, range checks, and algorithms both in the database and during validation.

Validation rules may be part of the preloading checks on the data as opposed to constraints or checks imposed by the database. These rules sometimes state that a value should be within a certain range. All such rules containing the word 'should' are conducted by preloading software. The use of the word 'should' in relation to these validation checks means that a warning message is generated when a value falls outside this range and the data are then checked further in relation to this value.

## 6.2 Summary of rules

### Recruitment survey sites table (**t\_site**)

<b>id</b>	Must contain an unique integer greater than 0..
<b>site</b>	Must have a value entered that is a valid survey site code as listed in the table <i>t_site_codes</i> .
<b>samp_date</b>	The sample date must be a legitimate date on or after 1 October 1996.
<b>time_of_day</b>	Time of the sample must be a valid 24-hour time and fall within the range of 0 – 2359. Also, time of the sample should be within the reasonable daylight hour's range of 600 – 1900.
<b>high_tide</b>	Time of high tide at the sample site must be a valid 24-hour time and fall within the range of 0 – 2359.
<b>target_sp</b>	Must contain a valid species code as listed in the <i>curr_spp</i> table of the <b>rdb</b> database.
<b>no_target_sp</b>	Must be an integer greater or equal to 0. The number of target species caught should not exceed 2000.
<b>bycatch</b>	Should consist of a space-separated list of bycatch species code (as listed in the <i>curr_spp</i> table of the <b>rdb</b> database) followed a integer greater than 0 that represents the number of that species caught.
<b>comments</b>	Can have any combination of up to 100 ASCII characters

### Recruitment survey site codes table (t\_site\_codes)

<b>site</b>	Must contain a unique 3-character uppercase alphabetic code.
<b>descriptn</b>	Can have any combination of up to 30 ASCII characters.
<b>map_sheet</b>	Must be a valid NZMS 260 map sheet number. Valid map sheet prefixes range from A to Z and numbers from 1 to 50. See Appendix 2 for a complete map of the NZMS 260 map series.
<b>grid_ref</b>	Must be a valid combination of NZMS 260 grid northings and eastings Both northings and eastings range 0 to 999.
<b>region</b>	Must be a New Zealand geographic region name.

### Recruitment length frequency table (t\_lgth)

<b>id</b>	Must contain a valid sample identification number as listed in the <i>t_site</i> table.
<b>species</b>	Must contain a valid species code as listed in the <i>curr_spp</i> table of the <b>rdb</b> database.
<b>measure_meth</b>	Must contain a valid fish measurement method code as listed in the <i>t_fish_meas_codes</i> table of the <b>rdb</b> database.
<b>lgth</b>	Must be an integer greater than 0 and should be within the reasonable range of 30 - 200.

#### Multiple columns check on species and length:

The fish length should be less than the maximum-recorded fish length for the species as recorded in the *curr\_spp* table in the **rdb** database.

#### Reasonable ranges for length by species.

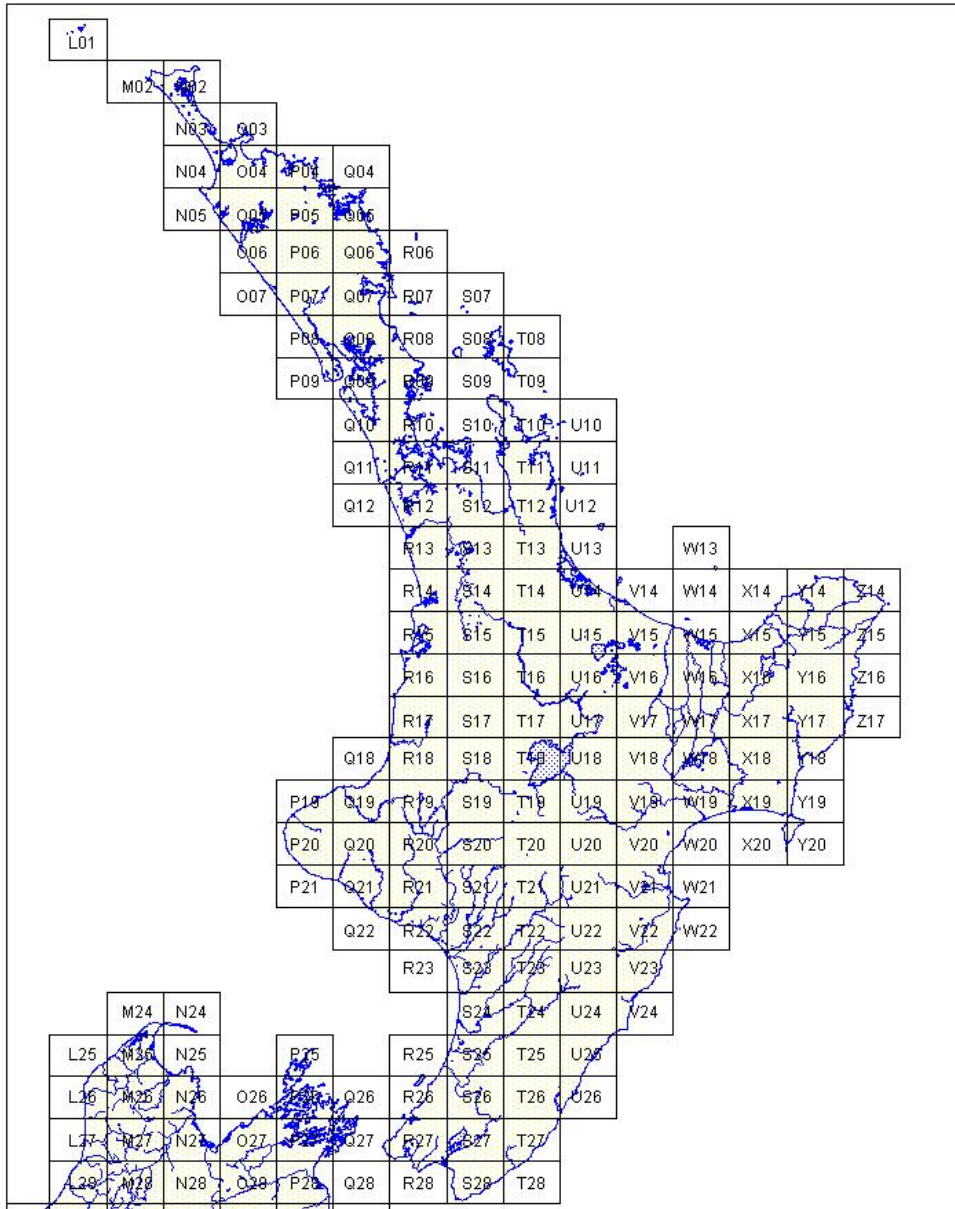
species code	minimum length	maximum length
KAH	35	170

<b>no_a</b>	Must be an integer greater than or equal to 0 and should be within the reasonable range of 1 - 50.
-------------	--

## **7 Acknowledgments**

The author would like to thank Dave Banks for his review and editorial comment for this document and Elizabeth Bradford for her technical input.

# Appendix



**Figure 2: Map of the coverage of the NZMS 260 map series for the North Island.**



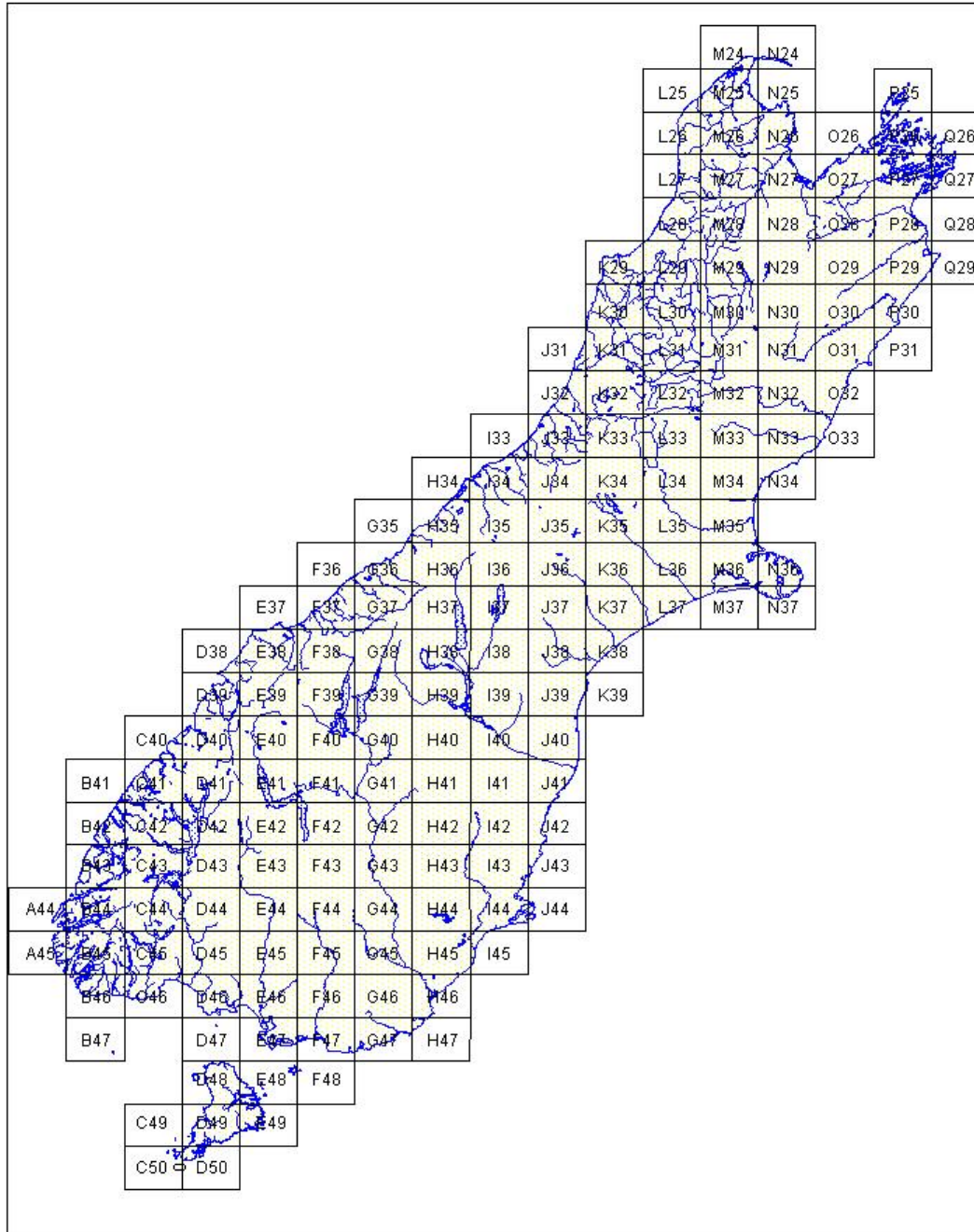


Figure 3: Map of the coverage of the NZMS 260 map series for the South Island.