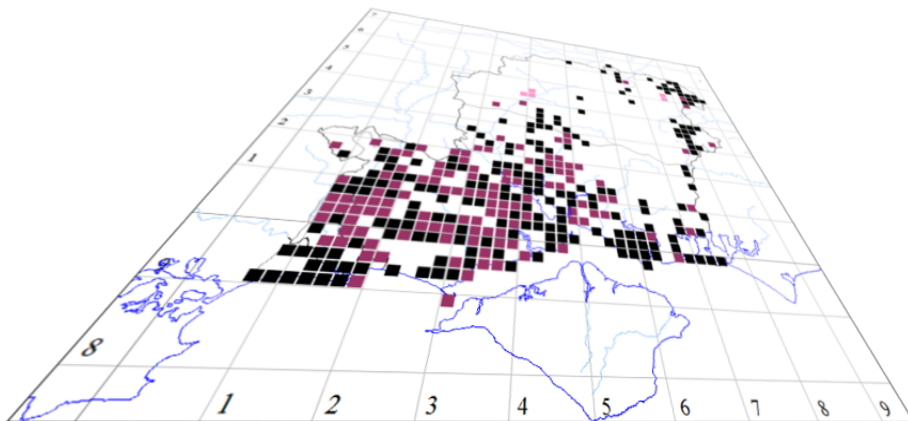


# *Chapter 2*

## *Preparing for Action*



***The author of this handbook is pleased to have errors pointed out and to receive suggestions for improvement and other comments.***

*Please send all communications to:*

[vcirecorder@hantsplants.org.uk](mailto:vcirecorder@hantsplants.org.uk)

### Things You Need to Know

This handbook assumes that you already know something about using Microsoft Windows. MapMate is a Windows-based program, and most of the facilities it offers are accessible through completely standard Windows ways of interacting with your computer. It does have a few quirks, most of which are good ideas to help you do certain tasks efficiently or to stop you doing risky things too easily, and some of which are – well, just quirks. In either case, when these crop up I will mention them. I am not going to describe in detail all the standard features of Windows programs; if you need to acquire this proficiency, there are plenty of books to help you, and you should look for one that matches the version of Windows that you are using.

Here is a brief list of the topics that you need to have absorbed before you go past this chapter.

- Running a program from desktop icons or the **Start** menu.
- Using Windows Explorer (also invoked through **Start / Computer**, **Start / Documents** etc.) to view folders and files on your system.
- Using a mouse to highlight and select information and to activate buttons; being aware that you have left and right buttons on your mouse that usually fulfil different roles.
- Using drop-down menus and toolbars to invoke actions.
- Using the keyboard to enter data and to navigate between different items on a form.
- Knowing that there are typing shortcuts to many features, using either the function keys (**F1** to **F12**) built into your keyboard or using letter keys in combination with other keys, usually **Ctrl** or **Alt**.
- Copying and pasting data from one place to another via the Windows Clipboard.

## CHAPTER 2 PREPARING FOR ACTION

### Working Through This Book

The general approach taken in this book is to lead you through a series of task-related topics, roughly in the order you are most likely to undertake them when starting to use MapMate. In that sense, it isn't a reference manual. I don't exhaustively document all options available on every menu, exactly in the order they appear in the menus; that's the job of the MapMate Help file, which you should certainly use if you are stumped, and which includes lots of useful information to supplement what you will find here. Instead, I introduce just enough of the facilities to let you do the task in hand. Others may be introduced later in another topic; some are left for you to explore on your own once you have acquired the basic knowledge.

I also include quite a lot of discussion in the early chapters about ways of doing things. MapMate gives you quite a lot of flexibility, and you will need to make informed judgements from time to time about the way that suits you best. You don't have to follow my suggestions for the best way; you may be using MapMate for different purposes, or come up with a genuinely better idea. I hope that at any rate you will have been made aware of the implications of taking one approach rather than another.

For these reasons, I recommend that you work through Chapters 2 to 4 in order, carrying out practical tasks on your own data or the training data provided. Most of the chapters from then on will depend on concepts, techniques and issues that have first been aired in these earlier ones.

### An Introduction to Databases

The name suggests that the mapping aspects of MapMate are the most important part of the program. Invaluable as those are, the key feature is one you may not give much thought to on a daily basis – and that's as it should be, because it concerns the underlying means by which MapMate stores your data.

However, if you want to get the most out of MapMate and avoid difficulties, it's worth taking a few minutes to understand how your data will be organised.

I'm sure we've all put together tables of species records like the following: either on paper, or in a computer spreadsheet.

Taxon	Vernacular	Site	Grid Ref	Date	Recorder	Method
<u>Acer campestre</u>	Field Maple	U171 Sarum Road, Sparsholt	SU4129	14/06/2010	Joel Miller	Field observation
<u>Acer campestre</u>	Field Maple	Little West Wood	SU4129	19/08/2004	Ian Ralphs	Field observation
<u>Acer campestre</u>	Field Maple	Pitt Down (East)	SU4129	19/08/2004	Ian Ralphs	Field observation
<u>Acer platanoides</u>	Norway Maple	Beacon Hill Plantation	SU4029	22/06/2005	Joel Miller	Field observation
<u>Acer platanoides</u>	Norway Maple	Farley Mount Country Park (W of Beech Clump)	SU410293	14/04/2011	Rand, M W	Field record / observation
<u>Acer platanoides</u>	Norway Maple	Farley Mount Country Park	SU4029	29/05/2000	Selby, Mr P.J.	Field record / observation
<u>Acer platanoides</u>	Norway Maple	Farley Mount	SU4028	09/10/1999	Bassett, Miss A.M., Martin, Miss L & Hartland M	Field record / observation
<u>Acer platanoides</u>	Norway Maple	Beacon Hill Plantation	SU4029	22/06/2005	Joel Miller	Field observation
<u>Acer pseudoplatanus</u>	Sycamore	Pitt Down (West)	SU4129	16/05/2002	Helen Lancaster	Field observation
<u>Acer pseudoplatanus</u>	Sycamore	Beacon Hill Plantation	SU4029	22/06/2005	Joel Miller	Field observation
<u>Acer pseudoplatanus</u>	Sycamore	Farley Mount Country Park	SU4029	29/05/2000	Selby, Mr P.J.	Field record / observation
<u>Acer pseudoplatanus</u>	Sycamore	Farley Mount	SU408293	14/04/2011	Rand, M W	Field record / observation
<u>Acer pseudoplatanus</u>	Sycamore	Beacon Hill Plantation	SU4029	22/06/2005	Joel Miller	Field observation

This is obviously quite a good way to look at data, but is it the best way to store it and to maintain it? For one thing, there is a lot of **repeating** information. The same site appears many times with the same grid reference. The same recorders' names occur often. And there are just a couple of standard methods for saying how the data was acquired. If this were a spreadsheet or a word processing document, all that information would indeed be repeated throughout the document.

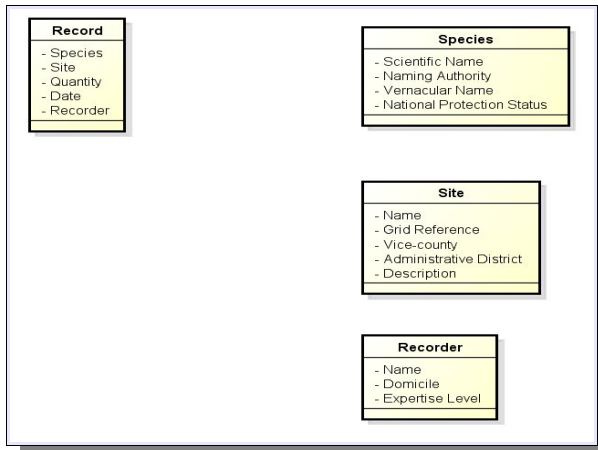
Now consider what would happen if you realise that the grid reference for Beacon Hill Plantation is wrong and should be SU3929. You must go through all your records and change the grid reference against that site name. Easily done in a spreadsheet, but if you have a very large data set, perhaps held in various files, a little less trivial. There's always the possibility of missing a few of the records or making a clerical error, and ending up with inconsistency where consistency should reign. And for each site, recorder, and every other type of data you use to flesh out the details of a record, there may be a host of other information needing to be maintained.

To deal with this issue, MapMate uses a technology known as a **relational database**. That's not something to research your ancestry in, but a technical term for a way to organise storage of data. It's not the only way, but it's the most common.

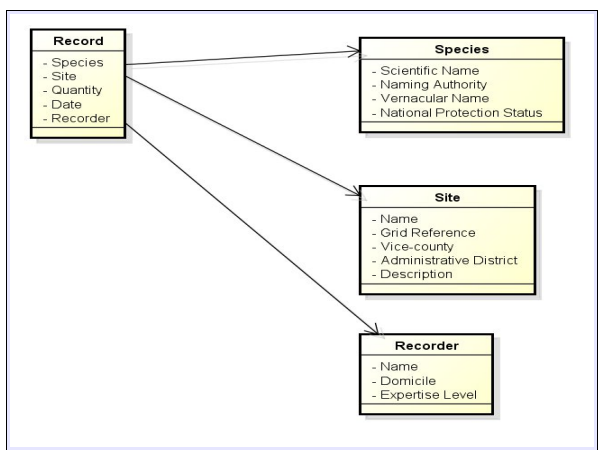
Clearly, some of the details of an individual plant record are unique to that record. The date it was made, the actual species recorded, the location and the recorder all need to be uniquely identified. But behind most of those items, there is a "back story" of associated information that doesn't change for each separate record. For practical purposes of recording, a species with a given scientific name has the same naming authority and

## CHAPTER 2 PREPARING FOR ACTION

vernacular name associated with it, and always belongs to the same plant family (we won't go into taxonomic niceties here!). And if we are truly recording at the same site location, then the same grid reference will apply, and perhaps other details we might want to record like habitat and ownership. (That's not to say that different sites can't have the same name – look at “Farley Mount” in the list above.)



A relational database, rather than organising data into a single big table like that illustrated above, partitions the data into several tables. Each holds just those data attributes that pertain to the thing it represents and can be defined uniquely for that given instance. For instance, suppose that we want to record details for recording events, including species recorded, place recorded, and recorder. Here's an example breakdown – for simplicity, details are not exactly as in MapMate, by the way.



Clearly the Record still has to be able to indicate which Species, Site and Recorder were involved in the recording event. Instead of repeating all those details, it keeps a reference to where they can be found in the other tables of the database – a pointer or index. For this, it needs something in the referenced table that uniquely identifies which instance, out of the many Species, Sites or Recorders, it refers to. Once it has that, it has access to all the other information that goes with that Species, Site or Recorder.

### Guaranteeing Data Uniqueness

What makes a suitable unique identifier? We might be tempted to use the scientific name for the Species. But scientific names change. If a name changes, the real-life taxon it refers to is still in most cases the same. We would be back to changing every Record in the database to use the new name – not much of a step forward. Or in the case of a Site, we might consider using the Site name; but because the identifier has to be unique, that would prevent us from ever having two site definitions at slightly different locations sharing the same familiar name for the neighbourhood.

To avoid these and other pitfalls, most software uses an arbitrary identifier for each record that is manufactured by the database management system and guaranteed to be unique for that database.

Database systems are designed to allow users to share a common pool of data, and MapMate is no exception. But the way it does this is not typical. The usual arrangement, with some simplification, is for the database to have a single home on a computer network

## CHAPTER 2 PREPARING FOR ACTION

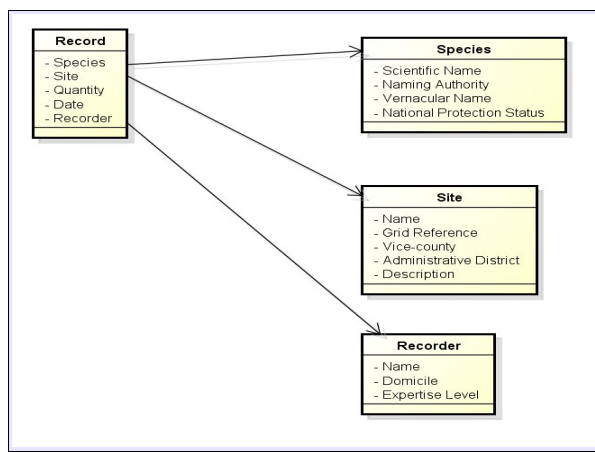
(or on the Internet, which is just another big computer network), and for the software running on users' computers to send requests across the network for the data, which is then posted back. (If you read up on this, you might see it referred to as a 'client-server architecture'. ) MapMate has no single big database for everyone's records; instead, your records live on your own computer system, and they only get to someone else's computer when you choose to send them there. Most of the time your copy of MapMate has no idea of what is going on in everyone else's copy of MapMate. That's fine, because it means that you can work without being connected to the outside world. With a rugged enough hand-held computer, you could even do your recording straight into MapMate in the field.

However it presents an interesting challenge when it comes to generating unique identifiers for each record in your database. (If you want a bit more jargon, these things are often referred to in the literature as 'primary keys'. ) Since MapMate is also designed for exchanging data periodically with other people, how can it know that the unique identifier it produces on one computer is different from all other unique identifiers on the computers of any other MapMate user you might possibly want to exchange data with, now or in the future?

The trick it uses is to build in the three-character license code of your MapMate license (what MapMate calls the 'Centre Unique Key') as part of the unique identifier. For instance, if your license code is 'ihb', then a typical identifier on your data would be '3x57a1hb' - made up of an arbitrary unique value followed by this license code. This guarantees that any unique identifier that MapMate produces will be unique across all MapMate users - provided MapMate is used in the way intended.

We shall explore the implications of this in more detail in Chapters 4 and 5, but there's one very important one to raise now. You may be tempted to install a single MapMate license on two or more computers, use them in more than one location, and then try to move data between them. Don't do this. In fact, MapMate has built-in safeguards to stop you exchanging your own data with yourself; if it didn't, you could end up with a complete mess. If you want to run MapMate, say, on both a desktop and a laptop, MapMate Ltd. have a special deal to provide a second license, and you should contact them.

### The Joy of Unique Identifiers



Now, if you followed this diagram from earlier on, you will probably have guessed what the pointers represent. The item called 'Species' in the Record table is just the unique identifier of a record in the Species table; and so on for the other tables. Given such an identifier, databases have efficient ways of looking up all the items of data associated with it in a table.

Let's consider a recent taxonomic name change: Man Orchid (formerly *Aceras anthropophorum*) is now *Orchis anthropophora*. Any records that were put into MapMate before the change will

## CHAPTER 2 PREPARING FOR ACTION

have referenced *Aceras anthropophorum* at that time.

Then the new name is adopted, and any reports of Man Orchid records need to use the new name. Records entered before and after the name change need to be brought together for listing or summarisation.

This will all happen automatically for us if we change the Scientific Name (and, of course, the Naming Authority!) in the Species table. The unique identifier doesn't change, either in the Species table or in the Records table. So any Record that refers to Man Orchid will automatically “get” the new detail associated with it from that moment on, because it's still pointing at the same Species record.

Species name changes like this aren't actually something you can do, by the way. They are issued in batches by MapMate Ltd., usually when the national society that coordinates recording for the species group asks for the changes; so, for example, requests for changes for vascular plants come from BSBI. You don't get to pick and choose the ones that you like.

You may be thinking that this is useful most of the time, but not in all circumstances. Suppose that you wanted to pull out a list of herbarium records, using the name recorded at the time of the collection or determination. Assuming that this name refers to a taxon that's still recognised but has a new name, that really won't be possible. Or supposing there is some ambiguity over the relationship of the old and new names, and you want to know what the original recorder's intentions were. A lot of database systems that have to deal with historical data face these problems, and usually tackle them with a system of 'versioning'; in other words, there are multiple versions of what might be considered the same entity, only one of which is 'current'; and a record that refers to this entity also keeps account of which version it refers to.

MapMate does use versioning for various purposes, as we shall see in later chapters; but not for dealing with archives of historical records in this way. It does allow synonyms to be declared, so that more than one identity can be kept going for a given taxon. But this really only relates to data entry; reporting is always done under the current approved name. What synonyms are available is again down to MapMate Ltd. and, ultimately, the national recording body.

When you come to Chapter 3 on data entry, I hope that this introduction will help you make sense of the way the various features fit together. When entering a new Record, you have two options to declare (for example) its Site details: either you can elect to set up a new Site right then, in which case you are diverted onto defining the Site before you can continue with the Record; or else you can look up a list of Sites that have already been entered and select one. If you choose to work with a predefined set of Site definitions (as you might if you were recording for a single reserve or estate, for instance) you can set up all the Sites in advance and then just select them into Records as you need. All this works because Sites have an existence independent of Records in a separate table.

If you go into the details of defining your own reporting in Chapter 4, you will find that much of the complexity in the process comes from linking up the data items from separate tables again, so that it looks to the observer as if it was all coming from one big table – the most readable format in most instances.



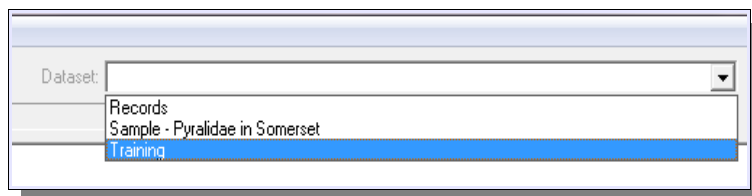
### The Training Dataset

Having done the mental preparation, there's one more practical step that you need to take before embarking on the next chapters. If you want somewhere to experiment with data entry in Chapter 3, and if you want to follow the examples in Chapter 4 onwards, there is a separate data set available called 'Training'.

You can download the Training data set from *<add URL here>*. You should then install it into the **Records** subfolder of your MapMate data folder; as discussed in Chapter 1, this will be either under **Documents\My MapMate\Data** if you are using a newer version of Windows (and as long as you agreed to move any legacy data here), or under the **Program Files\MapMate** folder if you are still running from a legacy installation.

Once this is done, you should close down MapMate if it is open, and re-open it.

Now, on the main MapMate screen, place your cursor over the box to the right of the caption that reads **Dataset:** and click. You should see a drop-down list like the following. If you don't see **Training**, check that you put the file in the right place.



Select the **Training** dataset from the list. You will then have a set of about 50,000 records for South Hampshire (vice-county 11) to play with.

When you want to do work 'for real' on your own data, you will need to revert to using the **Records** data set. This is the only data set on which you can do a number of important operations like exchanging data with other people; you can treat other record sets as useful for illustration purposes and some experimentation, but don't assume that you can keep several fully operational sets running in parallel. You will also find that a number of functions in MapMate (as well as closing and re-opening the program) automatically take you back to **Records**, so take care to check what is in the box before doing anything too experimental.