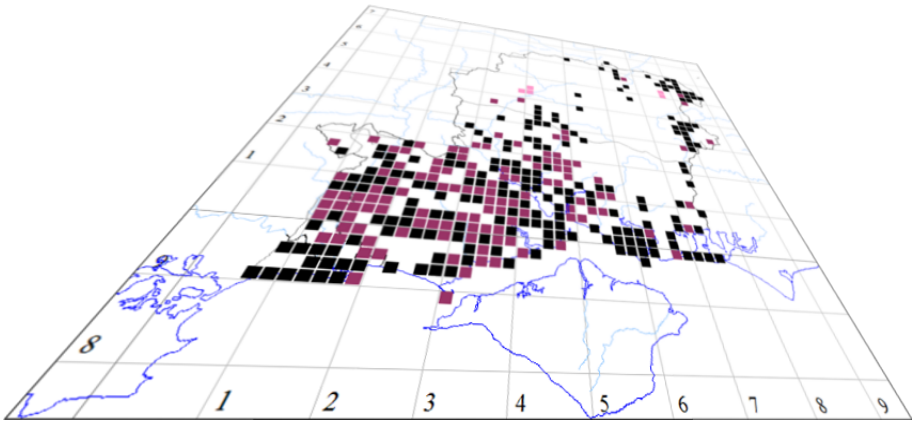
Appendix A The MapMate Data Model



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:

vc11recorder@hantsplants.org.uk

APPENDIX A: THE MAPMATE DATA MODEL

Introduction

This appendix is for those who want to develop their own queries and filters, understanding and exploiting the full potential of the MapMate database. It draws on the material published by MapMate on their web site in the paper *The Data Model*, bringing it up to date for more recent versions and amplifying some of the detail.

MapMate uses Microsoft Access as its database engine. This does not make a clear-cut physical distinction, as many databases do, between a 'server' domain which holds and maintains the data and delivers data on request, and a 'client' domain which makes requests of the server to satisfy the requirements of users. All data, and the functionality to make use of them, are customarily packaged into a single database file (with an **mdb** file name extension), and SQL queries are executed within this. In fact, it is possible to make a partial separation between client and server, because Access allows one database file to make cross-references to tables in another database file. This doesn't make any difference to the fact that SQL queries are run on the requesting side rather than by a server dedicated to the purpose.

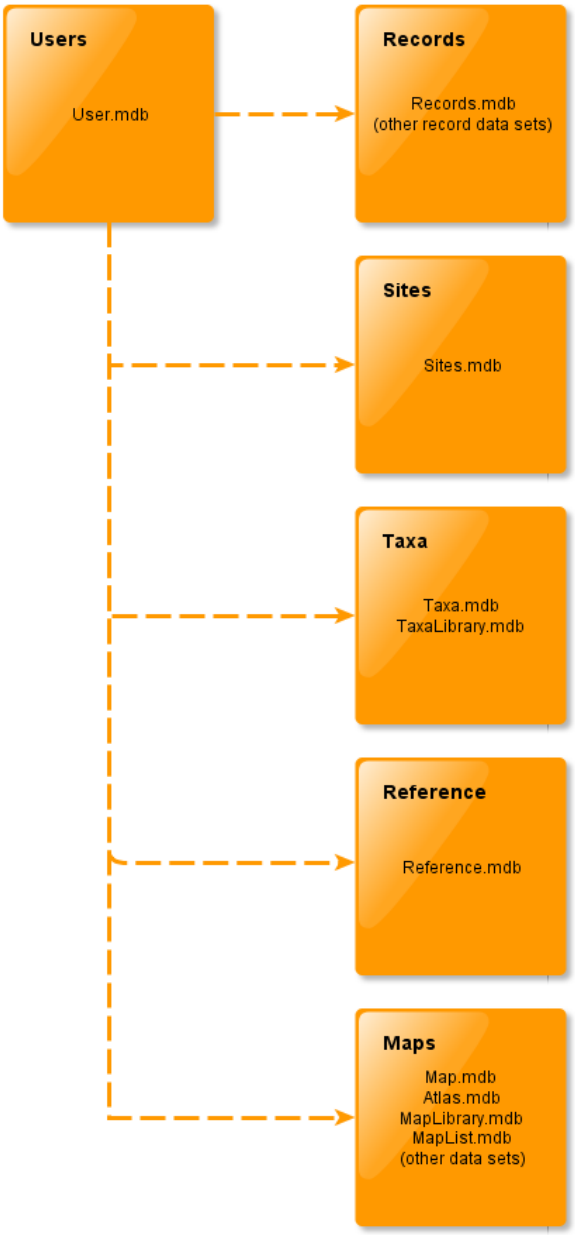
MapMate doesn't make quite this sort of separation, but it does make extensive use of the ability to cross-reference between different databases. MapMate data is packaged up into several database files, and the next part of this appendix explains the way they are organised. When you are writing queries, this isn't important, because one of the database files, **User.mdb**, brings all the cross-references together in one place so that the entire data repository looks like one database to SQL.

A word of warning, should you consider using Access to inspect your data. MapMate uses an old version of Access (Access 97) which is no longer supported by Microsoft. It's possible to use later versions of the Access package in Microsoft Office to display and modify the data in Access 97 files, but you can't change the construction of the database (which is a good thing!) However, if you open any of them directly, Access will offer you the chance to convert the file to the later format. **On no account do this**. If you do, MapMate will stop working; and you will be faced with a long and difficult process to recover the data to its original state. The best strategy is to make a copy of the file **User.mdb** under a different name in the same folder, and, if you wish, convert this to the later format. All other database tables are cross-referenced from this, so you can now safely browse the whole data set without being asked to convert again (and running the risk of giving the wrong answer in a moment of inattention). Note that if you do this, there are certain tables that are held in **User.mdb**, so you will effectively be taking a snapshot of those when you make the copy. If you come back at a later time to do further investigations, you'll need to make a new copy.

Unless you really know what you're doing, it's also a bad idea to try and modify MapMate data other than through MapMate itself. You must be quite clear how record generation numbers work, otherwise syncing and other operations will stop working properly. Most importantly, don't try to add new records to any table outside of MapMate's own procedures; you won't be able to create a Global Unique Key for the record which you can guarantee will never clash with one allocated by MapMate itself.

Packaging of MapMate Data

MapMate data is actually organised into several separate database files, with cross-references between them. The diagram below shows this physical organisation within the **My Mapmate / Data** folder. The **User.mdb** database brings references to all these databases together in one place through **linked tables**. This means that you don't need to cross-refer to the individual databases when writing queries (no "**IN**" clause needed!).



The **Maps** package is not documented further in this Appendix; in general it yields very little of use for reporting purposes, and it is not safely edited outside the facilities provided by MapMate itself.

Table Relationships

For queries, it's important to know how the individual tables relate to each other in the databases. MapMate has a few conventions when it comes to the way in which fields are named, and when a row in one table refers to a row in another (in database jargon, a **foreign key**).

Preliminaries: Data Conventions

Every table has a unique identifier (its **primary key**), and for most tables this is the **Global Unique Key** or **GUK** generated by MapMate. It comprises a sequence of 5 alphanumeric characters followed by the **Computer Unique Key** or **CUK** of the MapMate license where the record was generated. This combination guarantees the uniqueness of every record throughout a network of MapMate users exchanging data (as long as the same MapMate license is not used on more than one computer), and provides a capacity of over 60 million records of all types (Records, Sites, Recorders etc.) for each user.

A few tables that are supplied built into the system and are used as reference lookup lists have much simpler codes (for instance, one- or two-letter codes) as their primary key.

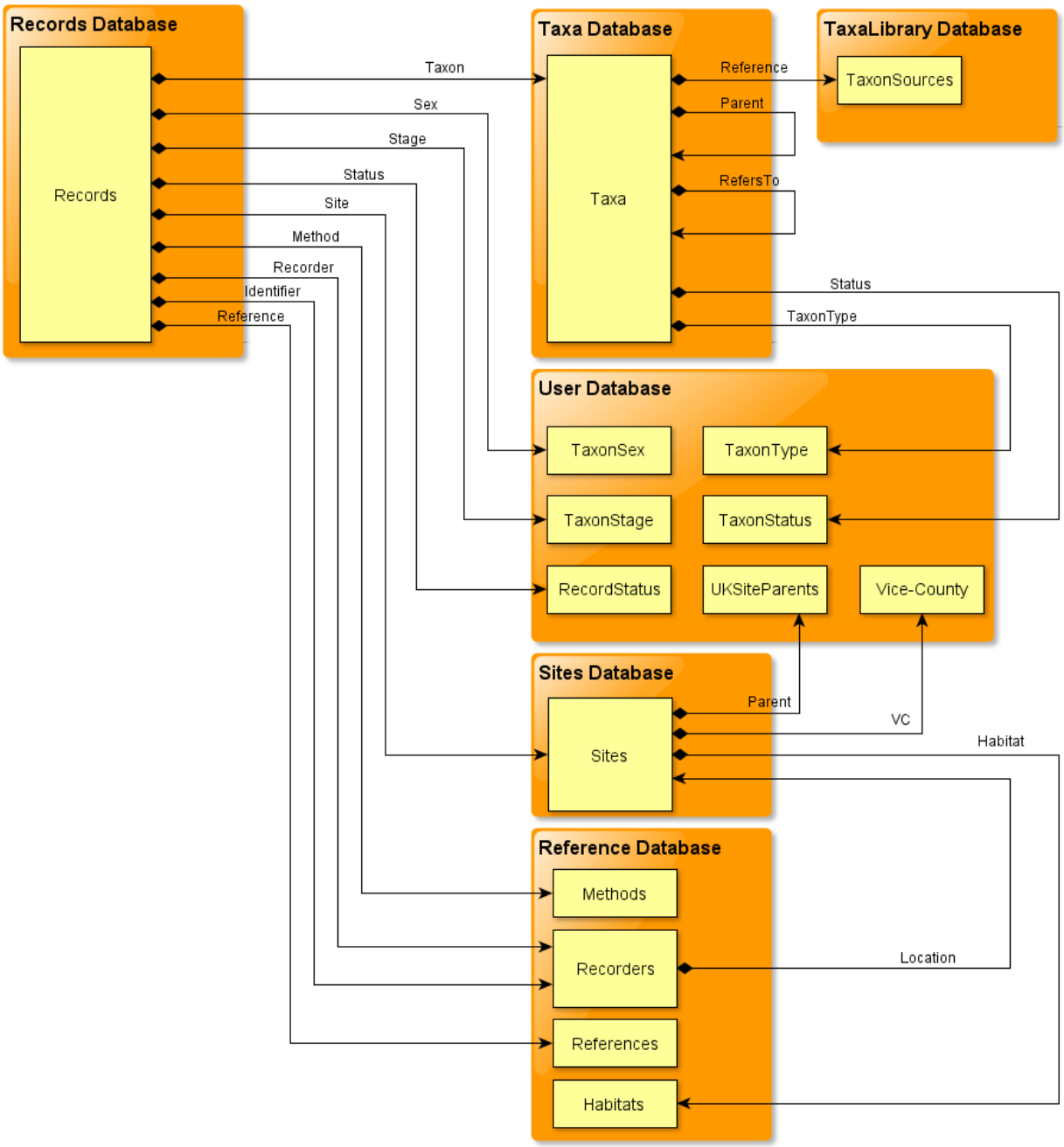
All database fields that are maintained automatically by the system start with an underscore (_). In general, you can retrieve these for display in queries but it is a bad idea to modify them except in very limited circumstances, discussed in Appendix B. In particular, **never** change the **_guk** field of any record! By the way, in queries you will always see these names enclosed in square brackets; Access requires this for any name with punctuation characters in it, including the space character. Punctuation characters are anything other than alphabetic and numeric characters; square brackets are also required if the field name starts with a numeric digit.

Database fields that are used as foreign keys have names starting with an asterisk (*). These will be primary keys of records in the table referred to; so in most cases, such a field will hold the GUK of a record in another table. In some tables, a record will have a reference back into its own table – for instance, to identify its parent in a hierarchy such as Taxa; such fields have a double asterisk (**) at the front of their name.

In the following diagrams, the separate database packages are shown in case you want to use an external database tool such as Access to find and view their constituent tables. You don't need to refer to them in queries: all queries directly reference the tables (or in a few cases, as we shall see later on, other queries that select from the tables), through the linkage in **User.mdb**. I've left the asterisks off the foreign key field names for the sake of brevity and readability.

APPENDIX A: THE MAPMATE DATA MODEL

Records and Related Data



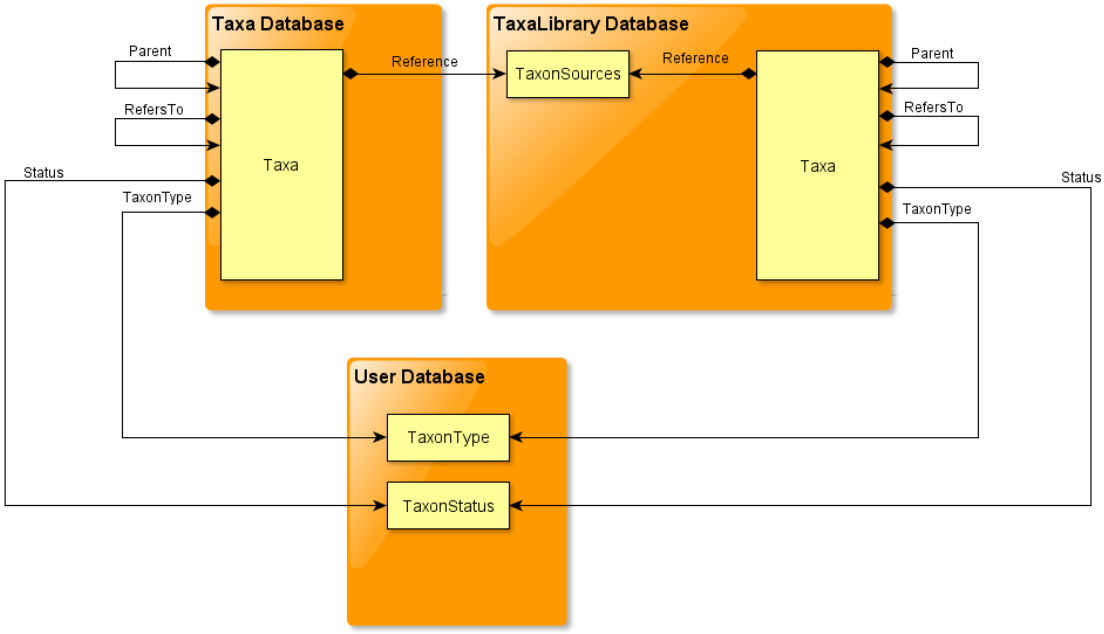
This diagram shows all the data that is directly available to you by explicit table joins when querying Records (in the strict sense). Most of these will be reasonably self-explanatory once we have reviewed table content, but a few deserve comment.

- **Records** has two separate references into **Recorders**, **Recorder** and **Identifier**. The first is the person who made the Record, the second the person who identified the specimen. If there is no independent verifier, these two both point to the same row in the **Recorders** table.
- **Taxa** has two separate links back into itself. **Parent** is a link to the next highest level in the taxonomic hierarchy; the taxonomic level of any row in **Taxa** is given by the

link to **TaxonType**. **RefersTo** is used for deprecated taxon names, and in such cases points to the currently accepted name. If the name is the currently accepted one, the row points to itself.

- The **Sites** table has a **Parent** reference to **UKSiteParents**, and normally this merely serves for MapMate to build a Site association internally with an administrative area (district, county etc.) within the British Isles and Ireland. However the Site data entry form does allow users to define their own one-level hierarchy of Site Parents and Site Types, provided they can accept that the parent will always be referred to as an 'Admin Area' on the data entry form. Since the link is not back into the **Sites** table itself, all the other attributes of a Site can't be attached to a Site Parent.
- Note that the Sites table is used to hold both Sites for biological records, and the locations of Recorders and people in other roles.

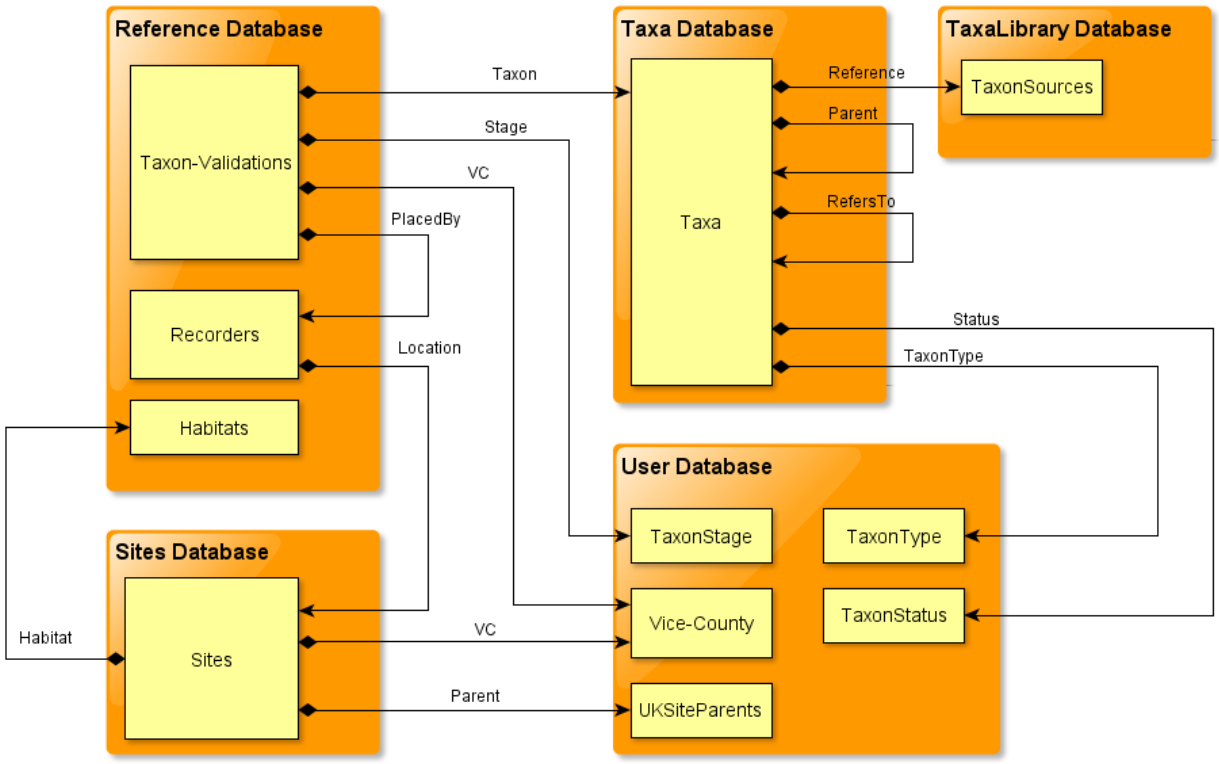
Taxa



There are two instances of the **Taxa** table, and they are held in separate databases. The **TaxaLibrary** instance holds the comprehensive set of taxon definitions supplied with MapMate. The **Taxa** instance is the optimised list which is recreated when a user sets up or reconfigures their taxonomic interests in **My Configuration...** Structurally they are exactly the same, and there is no hard-built database linkage between the two instances. However **User.mdb**, in its 'portal' role for the whole database, provides a link to **Taxa** under the same name, and a link to **TaxaLibrary\Taxa** under the alias **TaxaLib**.

For comments on other relationships on this diagram, see the preceding section.

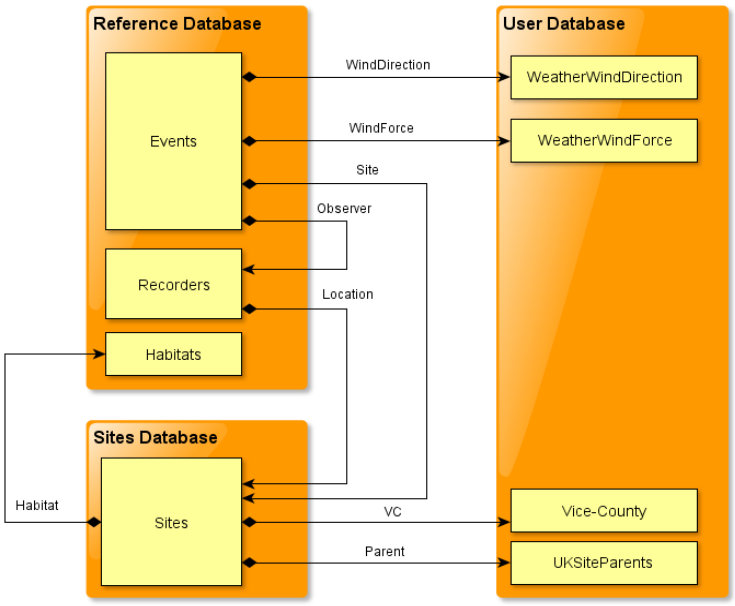
Taxon Validations



Taxon validation is invoked when a Record is entered, and user-defined validations are held in the table **Taxon-Validations**. All direct linkages are shown here for to show what is easily accessible to queries, although not all are used in the validation process.

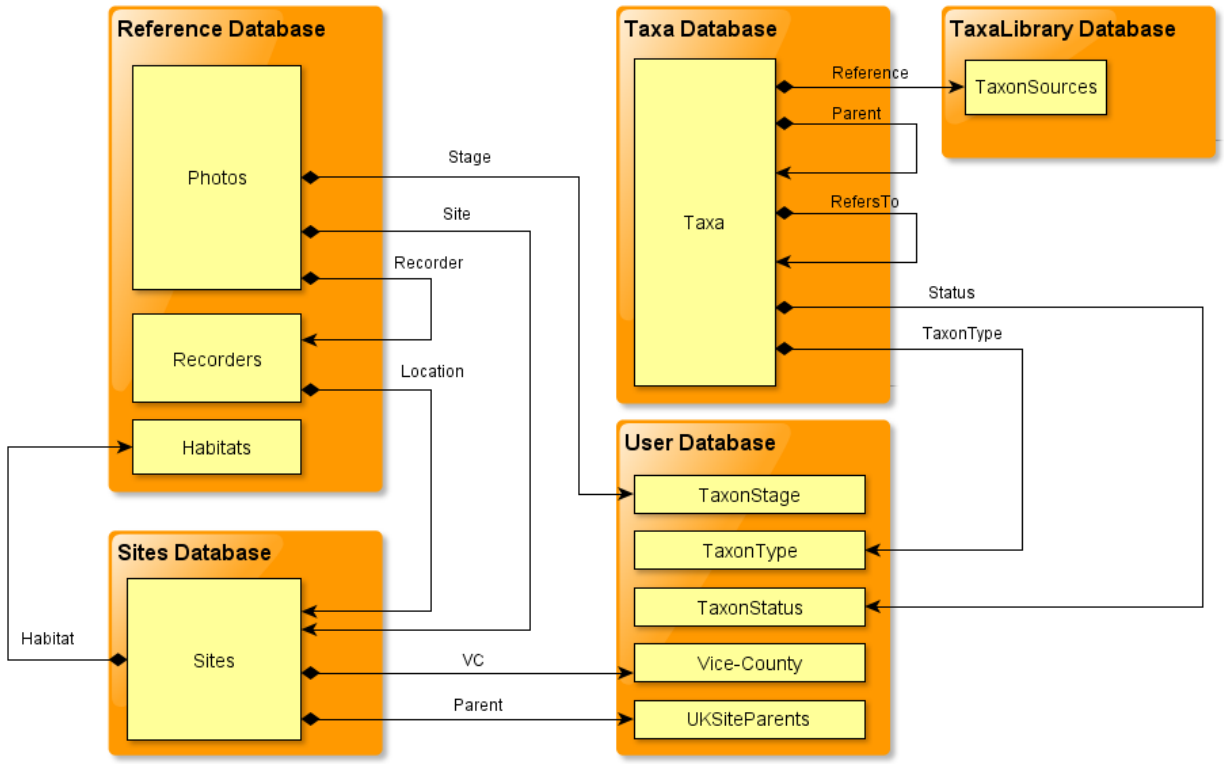
PlacedBy defines the person who set up the validation rule.

Events



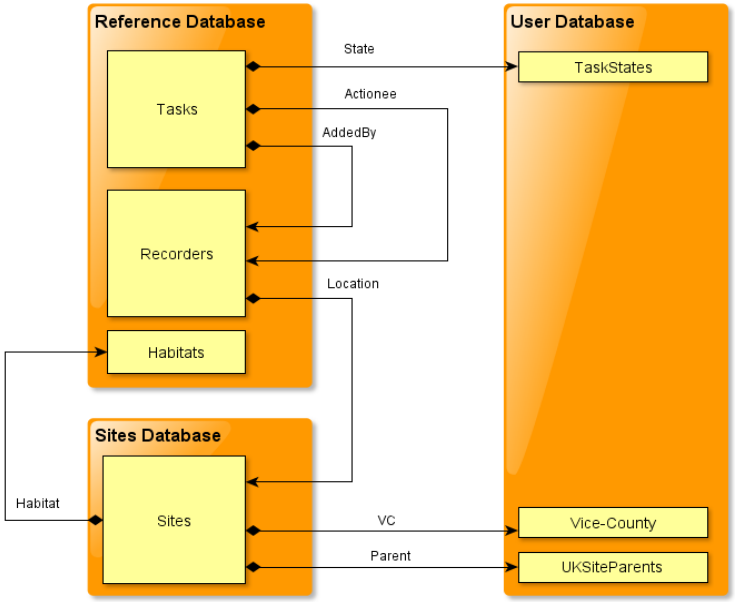
Events are a feature of MapMate that are not fully integrated with other aspects of the MapMate database. They relate to a Site, so they can be viewed as a calendar of recording incidents at a Site. There is no way in MapMate by which Records can be directly related to an Event; one must equate a Record's Site and an Event's Site, the date of a Record and that of a Site, and an Event Observer and a Record's Recorder.

Photos



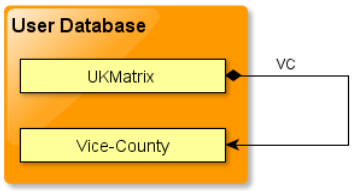
Photos are another partially implemented aspect of MapMate. They are given a Taxon, a Site, a Recorder (not specified, but presumably intended to be the person taking the photograph and not someone entering a record), and a date. But there is no scope for relating directly to a Record, so one could only try to infer this from these other pieces of information, and although it is likely, there is no guarantee that the two match up in reality.

Tasks



Tasks are a stand-alone feature in MapMate, other than using references to people in the **Recorders** table. Note that there are separate links for the person ordaining the task and the person expected to carry it out.

UKMatrix



UKMatrix is a lookup table used by MapMate internally to provide a checklist of vice-counties and administrative counties against 10km grid squares. Where a grid square embraces more than one vice-county or county there is a row for each combination in the same grid square.

Note that although the MapMate documentation on the data model describes the **County** as 'Admin Area', you should not assume from this that it provides a reference into the **UKSiteParents** table. The **County** field is a simple text field.

Table Content

Data Types

Name	Description
Text	A text field, maximum number of characters given in brackets following
GUK	An 8-character alphanumeric text field where the first 5 characters are a license-unique sequence number and the remaining 3 are the Computer Unique Key (CUK) of the licensee
Memo	A free text field allowing a practically unlimited amount of text
Integer	A whole number allowing values in the range -32768 to 32767
Long	A large whole number allowing values roughly in the range -2,000,000,000 to 2,000,000,000
Date	A Long value representing a date since 31 December 1899
Date/Time	A Long value representing a date and time of day
Y/N	A field representing 'False' as 0 and 'True' as a non-zero number (usually -1)

Special Fields

A field named **_guk** is the Global Unique Key for the record and, wherever present, always forms the primary key for the table. See the earlier section on data conventions in this Appendix for how it is composed.

The field named **_gen** is a **generation number** for the record. This number is automatically incremented by MapMate during data exchange ('syncing') processes, so it should not be thought of as a 'version number' for the individual record; since not all records are involved in all syncs, individual generation numbers will tend to go up by more or less large leaps. There is a table in MapMate (**sync**) which documents all data exchange transactions with partners, and it shows the generation number handled during that transaction. This is used to control the exchange process.

When a record is edited, its generation number is negated and set to the running "latest generation" number; this signals to the MapMate Replicator that it needs to be re-synced to partners on the next exchange. It is important that any query you write that entails an amendment to records sets the **_gen** field of those records in a manner similar to this (for Records): `SET … , Records.[_gen] = -val([thisgen]).`

When a record is "removed" (deleted) by its owner, all data fields except **_guk** and **_gen** are blanked, and **_gen** is negated and set to the running "latest generation" number. This enables deletions to be propagated to exchange partners through the Replicator. When a users remove records of which their CUK is not the owner, the records are actually deleted from the database.

In certain tables which contain system-generated data, all records are given a generation number of 2, indicating "system generated".

In general, records in all tables with a generation number of 1 are "built in" data supplied by MapMate as part of the distribution files or patches. However in the **Records** table the lowest generation numbers have a special meaning:

- **Generation 0** means that the record has been "quarantined" following a data import (something that you are given the option to do at the end of a successful import).
- **Generation 1** means that the record has been "archived": that is, it will not be exchanged in syncs with your partners.

Custom Datasets

When one makes use of a data set other than **Records** (using the drop-down at the top of the main MapMate screen), one is switched to a different database in the **Records** folder. Users create an alternative data set by making a renamed copy of the distributed **Dataset.mdb** file. This has a different structure from the main **Records** database, in that all the normally user-editable files are brought together in the **Dataset.mdb** database (and any databases derived from it). These include the following, as well as any maps and atlases the user creates.

Table	Notes
Dataset	Metadata about the data set – system generated
Events	
Habitats	
maptemplate	System-supplied template used by system to create new instances of maps and atlases
Methods	
Recorders	
Records	
References	
Sites	
Taxa	
Taxon-Validations	

As custom datasets are only a partially implemented feature in MapMate (there are no facilities for moving data subsets between data sets, and several of the MapMate functions only work from the **Records** database), I have not generally discussed them in the main chapters of this manual.

In the account that follows, I have listed only the locations where tables are found in the main **Records** data set.

If examining a MapMate database, be aware that switching to a different data set dynamically changes the linkages between **User.mdb** and its linked tables. If you want to see the main data files through **User.mdb**, be sure to switch back to the **Records** data set in MapMate first.

Special Views

The **Taxa** table gives access to the complete data sets for Taxa in the user's chosen range of species groups, as selected in **My Configuration...** Similarly, the **Sites** table presents all Sites held within the MapMate system.

Two built-in views on these tables restrict the visibility of data to the recording defaults that the user has currently specified, through the **Change Defaults** icon on the Analysis screen, through **Records / Change Defaults...** on the Data Entry form, or through the settings defined for a currently displayed Atlas. These are named **[Taxa\Default]** and **[Sites\Default]**. They can (and usually should) be substituted for the table names in most general purpose queries. They return an identical list of fields to the table from which they are derived.

Main tables and user-editable lookups

All these tables are explicitly designed to have data added and subsequently edited by the general user. Those marked with a † contain MapMate-supplied data which should not be deleted from them.

Records table (Records.mdb)

Name	Type	Notes
_guk	GUK	
***Taxon**	GUK	Foreign key to **Taxa** table
***Sex**	GUK	Foreign key to **TaxonSex** table
***Stage**	GUK	Foreign key to **TaxonStage** table
***Status**	GUK	Foreign key to **RecordStatus** table
***Site**	GUK	Foreign key to **Sites** table
Date	Date	Start date of recording event
DateTo	Date	End date of recording event (set equal to **Date** when not covering more than a single day, differs from **Date** for inexact dates)
Quantity	Long	Quantity recorded (negative for abundance codes, 0 for 'Present')
***Method**	GUK	Foreign key to **Methods** table
***Recorder**	GUK	Foreign key to **Recorders** table (maker of record)
***Identifier**	GUK	Foreign key to **Recorders** table (identifier of taxon; set same as ***Recorder** if no independent identifier)
Comment	Memo	
***Reference**	GUK	Foreign key to References table
_gen	Long	

APPENDIX A: THE MAPMATE DATA MODEL

Sites table (Sites.mdb)

Name	Type	Notes
_guk	GUK	
Name	Text (64)	Note fairly severe restriction on name length
****Parent**	GUK	Foreign key to **UKSiteParents** table specifying a containing region. Usually, but not inevitably, specifying an administrative area.
ViceCounty	Long	Vice-county number (Irish vice-counties, H1 to H40, are specified as 201 to 240). Foreign key to **Vice-County** table.
OSGridRef	Text (32)	OS grid reference in textual form (i.e. two-letter prefix for UK 100km x 100km squares, single-letter prefix for Irish 100km squares).
Xpos	Long	Eastings ordinate on OS National Grid, to nearest metre
Ypos	Long	Northings ordinate on OS National Grid, to nearest metre
_precision	Long	Precision of location, in metres
10kSquare	Text (4)	OS grid reference in textual form for containing hectad (10km x 10km square). Blank if not contained.
2kSquare	Text (5)	OS grid reference in textual form for containing tetrad (2km x 2km square). Blank if not contained.
_xo	Long	Eastings ordinate on OS National Grid of SW corner of Site extent, to nearest metre
_yo	Long	Northings ordinate on OS National Grid of SW corner of Site extent, to nearest metre
_xl	Long	Eastings ordinate on OS National Grid of NE corner of Site extent, to nearest metre
_yl	Long	Northings ordinate on OS National Grid of NE corner of Site extent, to nearest metre
***Habitat**	GUK	Foreign key to **Habitats** table
Description	Memo	
_gen	Long	

AutoFixList table (Reference.mdb)

This table is not explicitly edited by the user, but is populated by using the **Remember this** facility in Data Entry and removed via the **Change Defaults...** form in Data Entry.

Name	Type	Notes
UserEntry	Text (64)	The abbreviation or mnemonic provided by the user
_TableID	Text (64)	The name of the database table supplying the substitution
_FieldID	Text (64)	The name of the table column supplying the substitution
_Equivalence	Text (64)	The text of the table item comprising the substitution

Name	Type	Notes
_RequiredGuk	Text (10)	The GUK of the table row supplying the substitution

Events table (Reference.mdb)

Name	Type	Notes
_guk	GUK	
_owner	Text (8)	CUK of person who defined the Event (so in effect, only 3 characters long)
***Site**	GUK	Foreign key to **Sites** table
Date	Date	Start date of Event
DateTo	Date	End date of Event
***Observer**	GUK	Foreign key to **Recorders** table (person making the observation during the Event)
StartTime	Date/Time	Start time of Event. May be null.
EndTime	Date/Time	End time of Event. May be null.
***WindDirection**	GUK	Foreign key to **WeatherWindDirection** table
***WindForce**	GUK	Foreign key to **WeatherWindForce** table
Temperature	Long	To nearest degree Centigrade. May be null.
CloudCover	Long	To nearest whole number percentage. May be null.
Notes	Memo	
_gen	Long	

Habitats table (Reference.mdb)†

Name	Type	Notes
_guk	GUK	
Habitat	Text (64)	Name / description of the Habitat
HabitatCode	Text (16)	A Classification code (such as Broad Habitat Level 1 code). May be blank or null.
NBNKey	Text (16)	The NBN classification code. May be blank or null.
_gen	Long	

Methods table (Reference.mdb)†

Name	Type	Notes
_guk	GUK	
Method	Text (255)	Name or description for the Method for lookup
Explanation	Memo	A fuller account

Name	Type	Notes
**Parent	GUK	Currently unused, therefore may be blank or null
_gen	Long	

Photos table (Reference.mdb)

Name	Type	Notes
_guk	GUK	
_owner	Text (8)	CUK of person who defined the Event (so in effect, only 3 characters long)
Caption	Text (255)	
*Taxon	GUK	Foreign key to Taxa table
File	Text (255)	File path or URL
Image	Memo	To hold image
Date	Date	Starting date of recording window
DateTo	Date	End date of recording window (differs from Date for inexact dates)
*Photographer	GUK	Foreign key to Recorders table
*Site	GUK	Foreign key to Sites table
*Stage	GUK	Foreign key to TaxonStage table
_gen	Long	

Recorders table (Reference.mdb)

Name	Type	Notes
_guk	GUK	
Name	Text (64)	
*Location	GUK	Foreign key to Sites table
Comment	Memo	Additional descriptive information
_gen	Long	

References table (Reference.mdb)

Name	Type	Notes
_guk	GUK	
Author	Text (255)	Author or collator of Reference
Year	Integer	Year of publication / recording / deposition
Title	Text (255)	Title of publication, or conventional name
PageRef	Text (16)	Reference to page of publication or other indexing

Name	Type	Notes
		reference. May be blank or null
**SeeAlso	GUK	Currently unused
Location	Text (255)	Currently unused
Comment	Memo	Further descriptive matter
_gen	Long	

Tasks table (Reference.mdb)

Name	Type	Notes
_guk	GUK	
_owner	Text (8)	CUK of person who defined the Event (so in effect, only 3 characters long)
Task	Text (255)	Name or description of Task
Date	Date	Start date of Task
DateTo	Date	End date of Task
*Actionee	GUK	Foreign key to Recorders table (person to whom Task execution is allotted)
Notes	Memo	
*State	Text (8)	Foreign key to TaskStates table. Note, not a GUK
*AddedBy	GUK	Foreign key to Recorders table (person who defined the Task)
_gen	Long	

Taxon-Validations table (Reference.mdb)†

Name	Type	Notes
_guk	GUK	
*Taxon	GUK	Foreign key to Taxa table. Taxon to which validation applies
*Stage	GUK	Foreign key to TaxonStage table. Stage of development to which the warning is applied
Date	Date	Start date on which rule was defined (**not** the start of a date range to which the validation should be applied)
DateTo	Date	End date on which rule was defined (**not** the end of a date range to which the validation should be applied)
*VC	Long	Foreign key to Vice-County table. Note, not a GUK
*PlacedBy	GUK	Foreign key to Recorders table. Person defining the validation rule
Warning	Text (255)	Text displayed when validation is made
Comment	Text (255)	Explanatory notes (not displayed)

Name	Type	Notes
_gen	Long	

Filters table (User.mdb)†

Name	Type	Notes
_guk	GUK	
_filterClass	Text (16)	Must be one of 'Records', 'Sites', 'Taxa'
_filterName	Text (50)	Identifying name of Filter
_filterSql	Memo	Source code of SELECT SQL query
_gen	Long	

TaskStates table (User.mdb)†

In theory this is a table that can be extended by the user, as there is a facility to **Add New...** attached to the **State** field in the Data Entry form for Tasks. But it currently brings up a blank editing form. This means that Tasks can only be given one of two states at present ('To do' and 'Complete') without extending the table through a database tool like Access.

Name	Type	Notes
_guk	Text (8)	Note, not a true GUK but a short text mnemonic
Item	Text (64)	The displayed name / description for the State
_gen	Long	

UKSiteParents table (User.mdb)†

Although there is no independent facility to add new **UKSiteParent** records, it can be done by using **Add a New Admin Area...** on the **Admin Area** field of the **Sites** data entry form.

Name	Type	Notes
_guk	GUK	
Name	Text (64)	Name or description of region
Type	Text (64)	Categorisation of type of region. Note, free text, not looked up against any other table
_gen	Long	

System lookups

These tables are system-supplied and not intended to be amended in any way by the user.

Taxa table (Taxa.mdb, TaxaLibrary.mdb)

Name	Type	Notes
_guk	GUK	
Taxon	Text (64)	Scientific name
***TaxonType**	GUK	Foreign key to **TaxonType** table. (Most terms define position in the taxonomic hierarchy but there are a few outside the hierarchy altogether.)
****Parent**	GUK	Foreign key to parent taxonomic entity in **Taxa** table
Vernacular	Text (64)	Vernacular name
Authority	Text (64)	Naming authority
***Status**	GUK	Foreign key to **TaxonStatus** table
Comment	Text (255)	
***Reference**	GUK	Foreign key to **TaxonSources** table. Identifies where the names were compiled from (not the original typification document)
Code	Text (64)	Coding according to the most frequently used coding system for the species group
****RefersTo**	GUK	Foreign key to **Taxa** table. In the case of a deprecated name it refers to the preferred alternative in the table. Otherwise it refers to itself.
_specificIndex	Text (64)	Scientific specific name (indexed, for faster searches)
_gen	Long	

TaxonSources table (TaxaLibrary.mdb)

Readers will notice that this is a straight clone of the **References** table used for Records.

Name	Type	Notes
_guk	GUK	
Author	Text (255)	Author or collator of Taxon Source
Year	Integer	Year of publication deposition
Title	Text (255)	Title of publication, or conventional name
PageRef	Text (16)	Reference to page of publication or other indexing reference. May be blank or null
****SeeAlso**	GUK	Currently unused
Location	Text (255)	Currently unused
Comment	Memo	Further descriptive matter
_gen	Long	

RecordStatus table (User.mdb)

Name	Type	Notes
_guk	Text (1)	Note, not a GUK; actually a character representation of a short binary code (so limited to at most 255 values)
Status	Text (64)	The displayed term
Description	Text (255)	
Breeding	Y/N	Whether this status indicates breeding
UsedBy	Text (255)	A comma-separated list of species groups that use this term
_gen	Long	

TaxonSex table (User.mdb)

Name	Type	Notes
_guk	Text (1)	Note, not a GUK; actually a character representation of a short binary code (so limited to at most 255 values)
Sex	Text (64)	The displayed term
Description	Text (255)	
UsedBy	Text (255)	A comma-separated list of species groups that use this term
_gen	Long	

TaxonStage table (User.mdb)

Name	Type	Notes
_guk	Text (1)	Note, not a GUK; actually a character representation of a short binary code (so limited to at most 255 values)
Stage	Text (64)	The displayed term
Description	Text (255)	
Mature	Y/N	Whether the stage indicates maturity
UsedBy	Text (255)	A comma-separated list of species groups that use this term
_gen	Long	

TaxonStatus table (User.mdb)

Name	Type	Notes
_guk	GUK	In fact a mixture of genuine GUKs and other character codes implemented for special recording schemes
Status	Text (64)	The displayed term
Description	Text (255)	
_gen	Long	

TaxonType table (User.mdb)

Name	Type	Notes
_guk	GUK	
Classification	Text (64)	The displayed name
Level	Integer	The level in the taxonomic hierarchy, 7 being the highest (Domain) and 0 the lowest (Species). This entails a certain amount of "levelling off" for anything between the 8 main taxonomic ranks or below species level
_gen	Long	

UKMatrix table (User.mdb)

Name	Type	Notes
_guk	GUK	
OS10k	Text (4)	OS grid reference text for hectad (10km x 10km square)
County	Text (64)	County name
VC	Long	Vice-county number (Irish vice-counties, H1 to H40, are specified as 201 to 240). Foreign key to **Vice-County** table
_gen	Long	

Vice-County table (User.mdb)

Name	Type	Notes
_guk	Long	Not a GUK, but Vice-county number (Irish vice-counties, H1 to H40, are specified as 201 to 240).
Name	Text (64)	
VC	Text (3)	Vice-county code, with Irish vice-counties represented as "H1" to "H40" and "Any" represented as "0".
_gen	Long	

WeatherWindDirection table (User.mdb)

Name	Type	Notes
_guk	Text (8)	Not a GUK, but a short coded name (in fact, an identical name to Item in all existing cases)
Item	Text (64)	The displayed term
_gen	Long	

WeatherWindForce table (User.mdb)

Name	Type	Notes
_guk	Text (8)	Not a GUK, but either the Beaufort Scale number or "Unknown"
Item	Text (64)	A descriptive term
_gen	Long	

YesNo table (User.mdb)

Name	Type	Notes
_guk	Long	Not a GUK; values 0 or -1
Item	Text (64)	"Yes" or "No"
_gen	Long	

System internal files

These tables, and a number of others not documented, are used by the software for internal processes. They should never be amended in any way by the user. Some are only populated temporarily, and these are not documented in full.

Sync table (Reference.mdb)

This table is used by the system to keep a history of data exchange activity with sync partners, in both directions.

Name	Type	Notes
_guk	GUK	
_host	Text (8)	The CUK of the MapMate licensee sending the sync
_target	Text (8)	The CUK of the MapMate licensee intended to receive the sync
_prime	Text (64)	The identified name of the sync partner (may be blank)
_address	Text (255)	The email address of the sync sender, or "Unknown"
_location	GUK	A foreign key to the **Sites** table. Other identifying details for the sender

Name	Type	Notes
_lastsync	Long	The generation number of the last sync between these partners in this direction
_lastsyncon	Date/Time	The date and time of the last sync between these partners in this direction
_synrecords	Long	Number of records transmitted
_syncedits	Long	Number of changed records sent on last sync
_syncs	Long	Number of sync exchanges that have taken place between these partners in this direction
_ulf	Memo	The name of the Site filter currently applied to exchanges between these partners in this direction
_dlf	Memo	The name of the Taxa filter currently applied to exchanges between these partners in this direction. May be blank
_born	Date/Time	The date and time of the initial data exchange in the sequence of exchanges between these partners in this direction
_gen	Long	

Fields table (User.mdb)

Used to define Data Entry forms for new elements of the database. Discussion of this technique, which requires the cooperation of MapMate, is beyond the scope of this manual.

QueryHatchery table (User.mdb)

Temporary file used during development of new queries.

RecentFiles table (User.mdb)

I had assumed that this was used by the system to persist a list of recently loaded maps for the **File** menu in the main MapMate window, but as it's empty on my operational system I think it must now be redundant.

Requests table (User.mdb)

Used by the system to log requests for missing information placed on data partners through the Replicator.

TaxaBakup table (User.mdb)

Used by the system during re-creation of the Taxa table after a change to taxonomic interests in **My Configuration...** It holds an emergency backup copy of the old file against failure to create the new one. After successful reconfiguration it is emptied.