

River Macrophytes Database



Technical Manual

This document and the corresponding database is available for download from:
<http://jncc.defra.gov.uk/page-5815>

River Macrophytes Database Technical Manual

Version 2.0.0

July 2011

Produced by Conwy Valley Consultants Ltd, December 2004

Updated by Joint Nature Conservation Committee, July 2011

Contents

| | |
|---|-----------|
| Contents | 1 |
| 1 Introduction..... | 2 |
| 2 System requirements | 2 |
| 3 Installation | 2 |
| 3.1 Files provided..... | 2 |
| 3.2 Microsoft Access | 3 |
| 4 System architecture | 3 |
| 5 Data tables | 3 |
| 5.1 Non standard survey types | 4 |
| 5.2 Relationship diagrams and data types..... | 9 |
| 5.3 Indexing | 12 |
| 6 Queries | 14 |
| 7 Forms | 14 |
| 7.1 Startup page form..... | 14 |
| 7.2 View form | 14 |
| 7.3 Query form..... | 16 |
| 7.4 River form..... | 17 |
| 8 Reports | 17 |
| 9 Visual Basic modules | 17 |
| 9.1 BasCheckLinkedTables | 17 |
| 9.2 BasCheckUser..... | 17 |
| 9.3 basClassifyAll | 17 |
| 9.4 basClassifyRiver | 18 |
| 9.5 basConvertNGR..... | 18 |
| 9.6 basGetFilePathName..... | 18 |
| 9.7 basValidateNGR | 18 |
| 9.8 basValidateQueryCriteria | 18 |
| 10 River classification | 18 |
| 11 Data Export | 19 |
| 11.1 GIS output..... | 19 |
| 11.2 Excel spreadsheet output..... | 20 |
| 12 Data capture spreadsheet | 20 |
| 12.1 General notes | 20 |
| 12.2 The worksheets | 21 |

1 Introduction

The JNCC River Macrophytes Database is a Microsoft Access database constructed to house data on the plant communities of rivers in Britain and Ireland. It includes data from over 2000 sites and is the most comprehensive database of its kind. Data has been collected from all over the UK between 1977 and the present day following the methods of Holmes *et al.* (1999).

The database also holds a small amount of invertebrate and fish data.

This is the Technical Manual for the 2004 update to the database which resulted in a number of significant changes to the architecture, functionality and presentation of the database. It contains the technical information about the system architecture, data tables, security etc. It also contains details of changes made to the data during migration from the old system. This manual is accompanied by the **User Guide**. There is also a **readme.txt** file containing supplementary information. For information on the data and database structure, see the **User Guide**.

2 System requirements

| | |
|---|--|
| Processor | Pentium P400 or better |
| Memory | 128MB |
| Disk space | 200 MB to install the database (version 1.0.0). An additional 160 MB is required if you are also installing MS Access runtime and its support files. |
| Operating system | Windows NT4 (SP6), 2000 (SP4), or XP (SP1) |
| Screen resolution | Min of 1024 x 768 Optimised for 1280 x 1024 |
| Microsoft data access components (MDAC) | 2.8 or later (supplied) |
| Microsoft Excel 2000 | is required if the data loading routine or write to Excel spreadsheet routine is to be used. |

The full version of Microsoft Access 2000 is recommended

3 Installation

The database is available for download from the JNCC website. For the purpose of this documentation it is assumed that you are going to install the system into the default directory C:\River_macrophytes\. If this is not the case you will need to modify the commands that you use accordingly. The database consists of two MS Access .mdb files: macrophytes_data.mdb contains the macrophyte survey data, macrophytes_code.mdb contains a set of database queries to help extract information. These two files must be kept within the same folder on your local system.

3.1 Files provided.

| | |
|---------------------------------------|---|
| The macrophytes database | macrophytes_code.mdb |
| The macrophytes forms and queries | macrophytes_data.mdb |
| The 'read me' latest information file | readme.txt |
| The documentation files | User_Guide.pdf Technical_Manual.pdf Data_Capture_Spreadsheet_Guidance.pdf |
| The data spreadsheet | Data_Capture_Spreadsheet.xls |

You will normally need to log on as an administrator to install programs on your computer. If you are uncertain about installation you should contact your organisation's IT department for assistance.

3.2 Microsoft Access

If you do not have Microsoft Access 2000 on your computer you will need to install a runtime only version before you can use the River Macrophytes database. To install the Runtime Access 2000 go to the downloaded zip files and folder AccessRT\ and double click on the file DATA1.MSI. This will start the windows installer and you should follow the on screen instructions. Make a note of the directory that you choose to install the MSACCESS.EXE program file in, as you will need to edit the shortcut to point to it.

4 System architecture

The system is a classic 'split database' architecture consisting of an 'application' database file (code.mdb) and a 'data' database file (data.mdb). These two files must be kept in the same folder in your local directory as they are linked by a relative file path.

- The application file macrophytes_code.mdb contains the forms, queries, reports, program code modules, etc. but contains no data tables, only links to the data file.
- The data file macrophytes_data.mdb contains the data tables only.

The database can be accessed and queried using the forms and queries provided in macrophytes_code.mdb, or by creating and running queries directly in macrophytes_data.mdb.

5 Data tables

These are in the data file macrophytes_data.mdb. There are 18 data table in the database which are supported by 26 dictionary tables, 3 standalone tables containing relatively static data used for calculations and 2 temporary legacy data tables included to avoid losing any data from the old system.

To get the detailed information for each table, display the table in design view in MS Access and the field names, data types etc will be displayed. Each field has a comprehensive comment describing the information it holds.

If you do this from the application database macrophytes_code you will get a reminder that the tables are linked tables and cannot be modified. If you need to modify a table you need to open the data database macrophytes_data.mdb.

Some of the tables and/or fields have been provided to enable information to be added in the future or in anticipation of possible future enhancements to the system.

Table tbl_dict_catchment holds river catchment information and is still to be populated.

Table tbl_dict_CB_type holds dictionary information for use with any future enhancement to calculate a classification based on Callitricho-batrachion vegetation.

Tables `tbl_dict_fish_species`, `tbl_dict_invertebrate_species`, and `tbl_dict_macrophyte_species` have fields to hold the NBN ID and date allocated. This will enable links to be made to the NBN database in the future. These are still to be populated.

Table `tbl_dict_river_type` has a field for an additional description. This is to hold a full description of the type for use when the additional information button is clicked on the view data form. This is still to be populated.

Table `tbl_macrophyte_survey` has a dictionaried field to hold the CB type data (see above). This is still to be populated.

Table `tbl_river` has a dictionaried field to hold the catchment data (see above). This is still to be populated.

Table `tbl_site` has a dictionaried field to hold the soil group data (see above). This is still to be populated.

5.1 Non standard survey types

The system is only designed to handle JNCC 3 point survey data. If a survey is done using any other method e.g. JNCC 5 point, this will need to be manually translated to 3 point prior to entering it into the database. This is because all the calculations for river typing etc. are based on 3 point cover code values. It is possible to store 5 point, MTR, or other survey type data in the database, but they have to be entered directly into the tables using the Access table datasheet interface as the forms do not have fields for these other survey type fields.

The fields in question are

`tbl_macrophyte_survey.percent_cover_code_5-point`
`tbl_macrophyte_survey.original_percent_cover_code`

`tbl_macrophyte_surv_species_cvr_aquat.percent_cover_code_5-point`
`tbl_macrophyte_surv_species_cvr_aquat.original_percent_cover_code`
`tbl_macrophyte_surv_species_cvr_aquat.original_relative_cover_code`

`tbl_macrophyte_surv_species_cvr_bank.percent_cover_code_5-point`
`tbl_macrophyte_surv_species_cvr_bank.original_percent_cover_code`
`tbl_macrophyte_surv_species_cvr_bank.original_relative_cover_code`

5.2 Relationship diagrams

The following diagrams show the tables and their relationships as functional units. To get a full overview of the whole database structure, view the relationships diagram from the tools menu in MS Access.

Standalone tables

tbl_Threshold

level
threshold
tolerance

tbl_fish_site_data_temp

SiteNo
NGR
Easting
Northing

tbl_weighting

macrophyte_ID
macrophyte_name
relative_cover_code
level_1
level_2a
level_2b
level_3a
level_3b
level_3c
level_3d
level_4a
level_4b

tbl_invertebrate_site_information_temp

MacrophyteSiteNo
SiteName
GREF
Easting
Northing

tbl_NGR_conversion

NGR_code
NGR_east
NGR_north

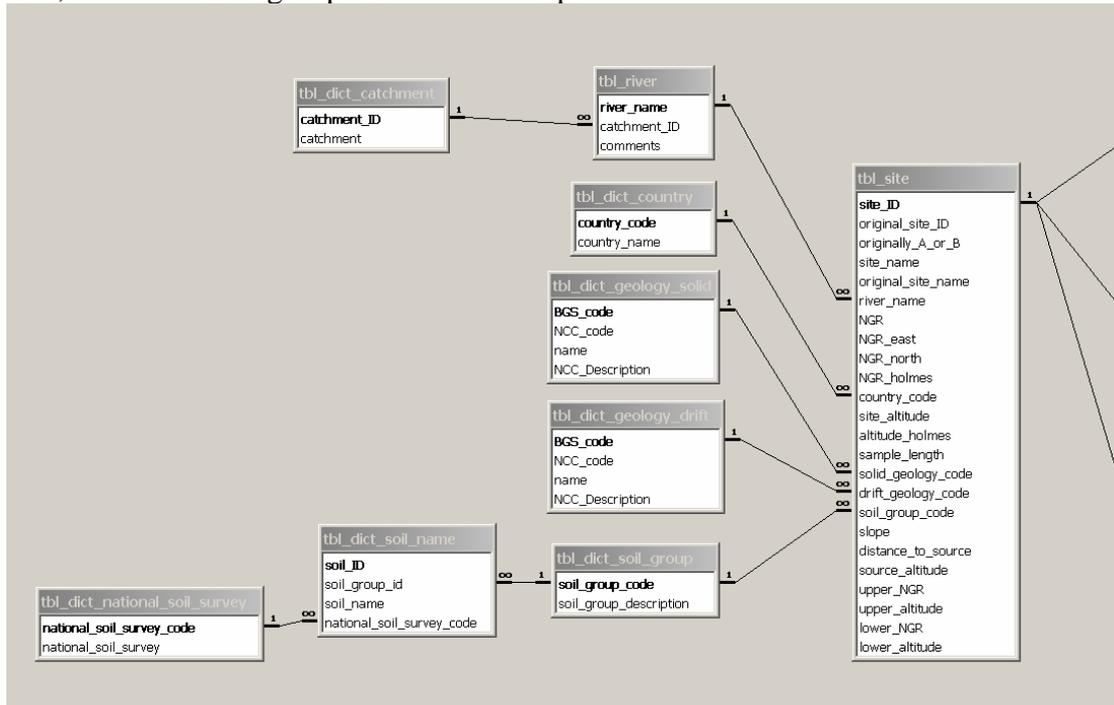
tbl_merged_species_aquat

survey_ID
old_macrophyte_ID
old_percent_cover_code_3_point
old_percent_cover_code_5_point
old_original_percent_cover_code
old_relative_cover_code
old_original_relative_cover_code
new_macrophyte_ID
new_percent_cover_code_3_point
new_relative_cover_code
comment

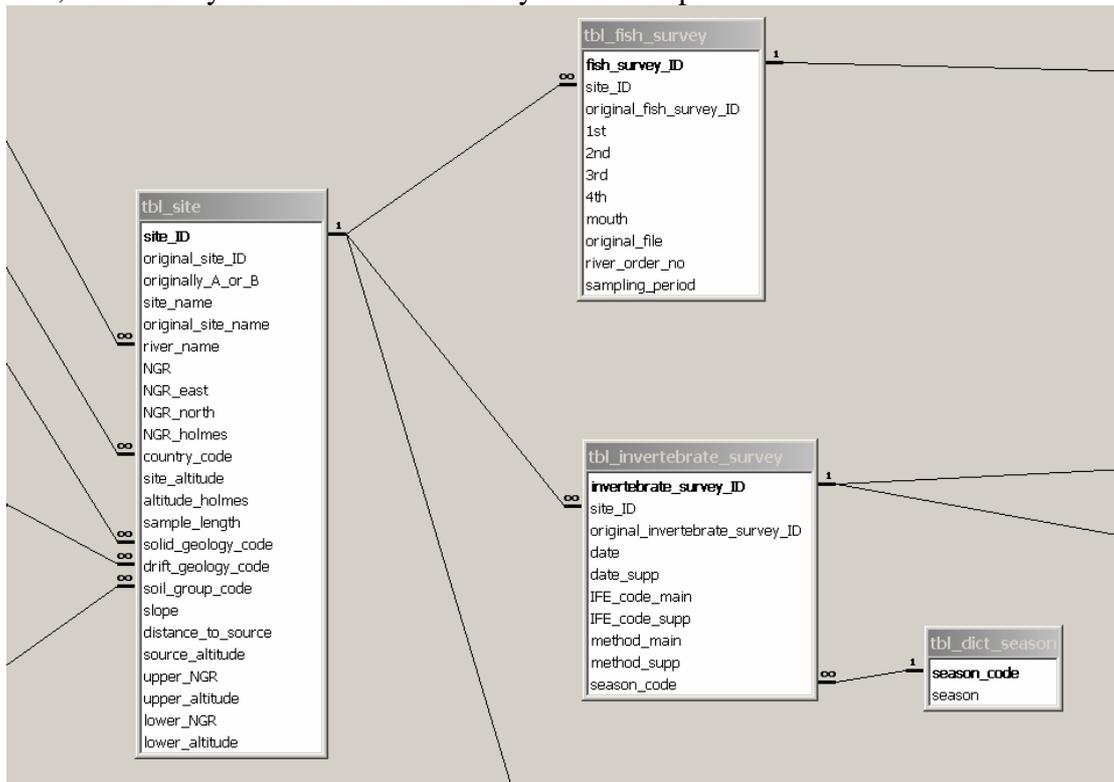
tbl_merged_species_bank

survey_ID
old_macrophyte_ID
old_percent_cover_code_3_point
old_percent_cover_code_5_point
old_original_percent_cover_code
old_relative_cover_code
old_original_relative_cover_code
new_macrophyte_ID
new_percent_cover_code_3_point
new_relative_cover_code
comment

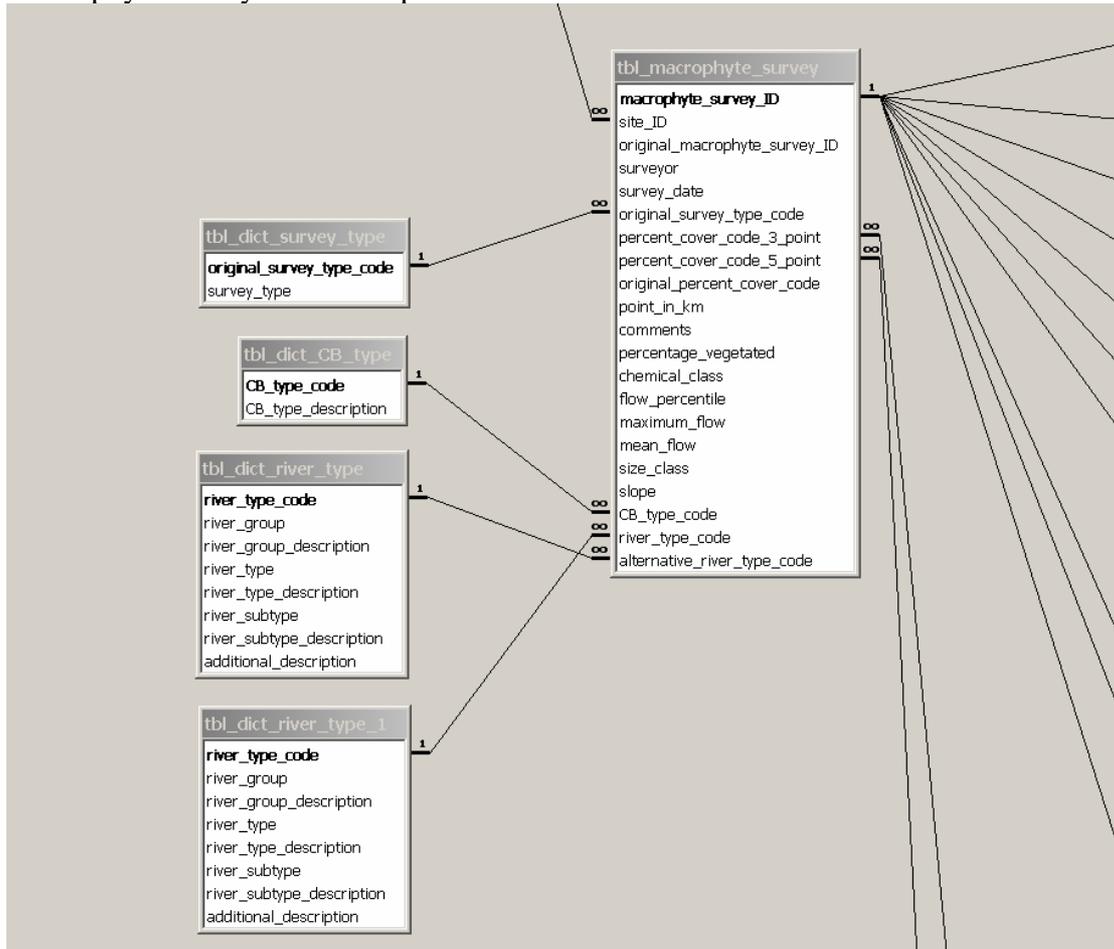
Site, River and Soil group table relationships



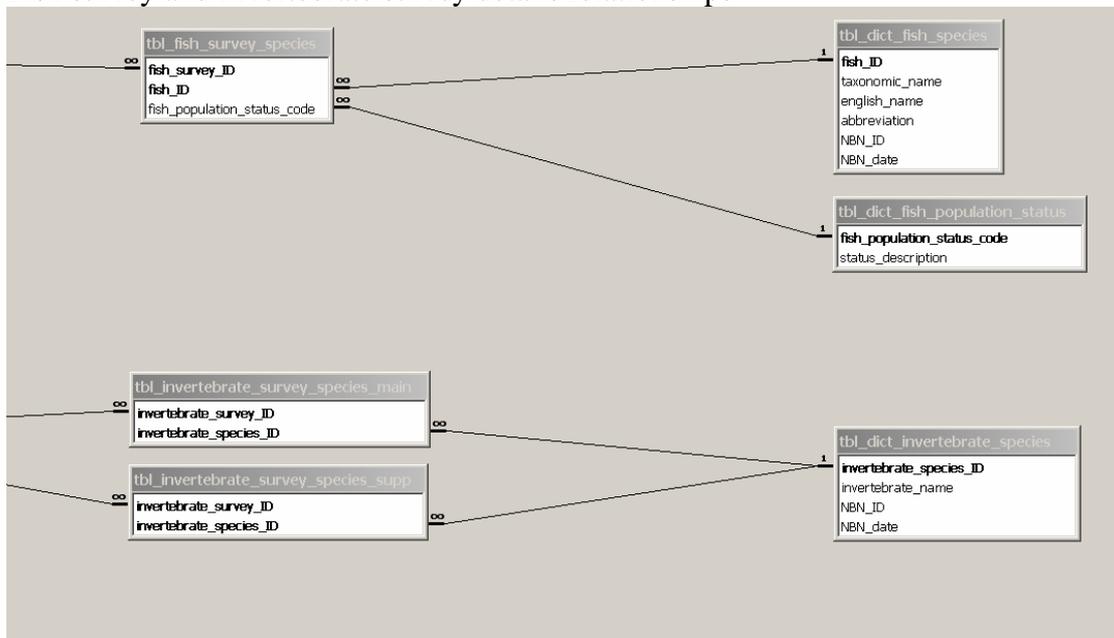
Site, Fish survey and Invertebrate survey relationships



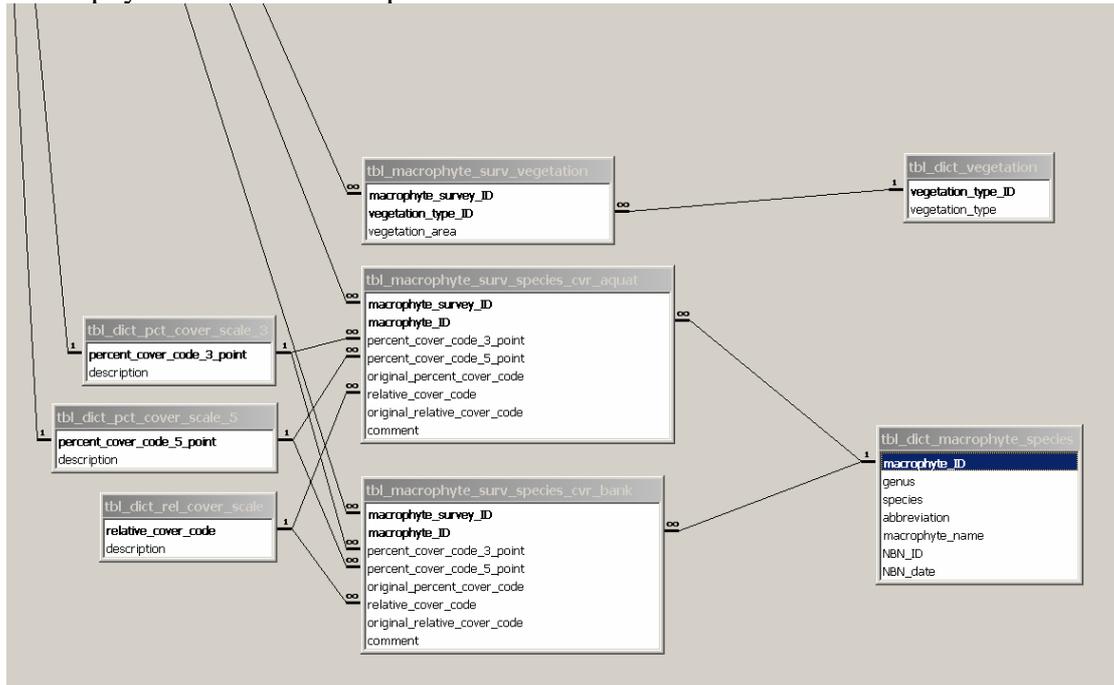
Macrophyte survey relationships



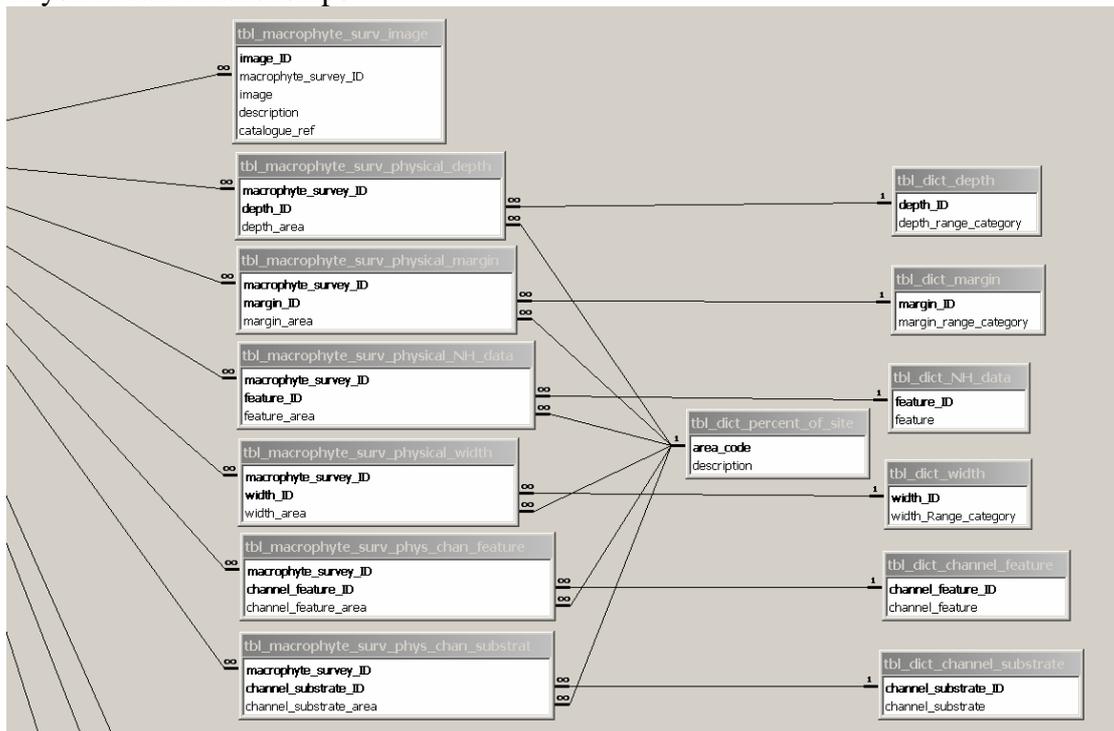
Fish survey and Invertebrate survey details relationships



Macrophyte cover relationships



Physical data relationships



5.2 Data types for primary and foreign key fields

| Parent table | PK type | Child tables | FK type |
|---------------------------------|----------|--|----------|
| tbl_dict_catchment_area | long int | tbl_river | long int |
| tbl_dict_CB_type | text 4 | tbl_macrophyte_survey | text 4 |
| tbl_dict_channel_feature | byte | tbl_macrophyte_surv_phys_chanl_feature | byte |
| tbl_dict_channel_substrate | byte | tbl_macrophyte_surv_phys_chan_substrat | byte |
| tbl_dict_country | text 2 | tbl_site | text 2 |
| tbl_dict_depth | byte | tbl_macrophyte_surv_physical_depth | byte |
| tbl_dict_fish_population_status | text 4 | tbl_fish_survey_species | text 4 |
| tbl_dict_fish_species | long int | tbl_fish_survey_species | long int |
| tbl_dict_geology_drift | long int | tbl_site | long int |
| tbl_dict_geology_solid | long int | tbl_site | long int |
| tbl_dict_invertebrate_species | long int | tbl_invertebrate_survey_species_main | long int |
| | | tbl_invertebrate_survey_species_supp | long int |
| tbl_dict_macrophyte_species | long int | tbl_macrophyte_surv_species_cover | long int |
| tbl_dict_margin | byte | tbl_macrophyte_surv_physical_margin | byte |
| tbl_dict_national_soil_survey | text 2 | tbl_soil_name | text 2 |
| tbl_dict_NH_data | long int | tbl_macrophyte_surv_physical_NH_data | long int |
| tbl_dict_pct_cover_scale_3 | byte | tbl_macrophyte_survey | byte |
| | | tbl_macrophyte_surv_species_cover | byte |
| tbl_dict_pct_cover_scale_5 | byte | tbl_macrophyte_survey | byte |
| | | tbl_macrophyte_surv_species_cover | byte |
| tbl_dict_percent_of_site | byte | tbl_macrophyte_surv_physical_depth | byte |
| | | tbl_macrophyte_surv_physical_margin | byte |
| | | tbl_macrophyte_surv_physical_NH_data | byte |
| | | tbl_macrophyte_surv_physical_width | byte |
| | | tbl_macrophyte_surv_phys_chan_feature | byte |
| | | tbl_macrophyte_surv_phys_chan_substrat | byte |

| | | | |
|--|--------------------|---|----------|
| tbl_dict_rel_cover_scale | byte | tbl_macrophyte_surv_species_cover | byte |
| tbl_dict_river_type | text 5 | tbl_macrophyte_survey | text 5 |
| tbl_dict_season | text 2 | tbl_invertebrate_survey | text 2 |
| tbl_dict_survey_type | text 5 | tbl_macrophyte_survey | text 5 |
| tbl_dict_vegetation | byte | tbl_macrophyte_surv_vegetation | byte |
| tbl_dict_width | byte | tbl_macrophyte_surv_physical_width | byte |
| tbl_fish_site_data_temp | long int | Stand-alone table | |
| tbl_fish_survey | <i>long int</i> | tbl_fish_survey_species | long int |
| tbl_fish_survey_species | long int, long int | No child tables | |
| tbl_invertebrate_site_information_temp | long int | Stand-alone table | |
| tbl_invertebrate_survey | <i>long int</i> | tbl_invertebrate_survey_species_main | long int |
| | | tbl_invertebrate_survey_species_supp | long int |
| tbl_invertebrate_survey_species_main | long int, long int | No child tables | |
| tbl_invertebrate_survey_species_supp | long int, long int | No child tables | |
| tbl_macrophyte_survey | <i>long int</i> | tbl_macrophyte_surv_image | long int |
| | | tbl_macrophyte_surv_physical_depth | long int |
| | | tbl_macrophyte_surv_physical_margin | long int |
| | | tbl_macrophyte_surv_physical_NH_data | long int |
| | | tbl_macrophyte_surv_physical_width | long int |
| | | tbl_macrophyte_surv_phys_chan_feature | long int |
| | | tbl_macrophyte_surv_phys_chan_substrat | long int |
| | | tbl_macrophyte_surv_species_cvr_aquatic | long int |
| | | tbl_macrophyte_surv_species_cvr_bank | long int |
| | | tbl_macrophyte_surv_vegetation | long int |
| tbl_macrophyte_surv_image | <i>long int</i> | No child tables | |
| tbl_macrophyte_surv_physical_depth | long int, byte | No child tables | |
| tbl_macrophyte_surv_physical_margin | long int, byte | No child tables | |
| tbl_macrophyte_surv_physical_NH_data | long int, long int | No child tables | |
| tbl_macrophyte_surv_physical_width | long int, byte | No child tables | |
| tbl_macrophyte_surv_phys_chan_feature | long int, byte | No child tables | |

| | | | |
|---|--------------------|-------------------------|----------|
| tbl_macrophyte_surv_phys_chan_substrat | long int, byte | No child tables | |
| tbl_macrophyte_surv_species_cvr_aquatic | long int, long int | No child tables | |
| tbl_macrophyte_surv_species_cvr_bank | long int, long int | No child tables | |
| tbl_macrophyte_surv_vegetation | long int, byte | No child tables | |
| tbl_NGR_conversion | text 2 | Stand alone table | |
| tbl_river | text 255 | tbl_site | text 255 |
| tbl_site | <i>long int</i> | tbl_fish_survey | long int |
| | | tbl_invertebrate_survey | long int |
| | | tbl_macrophyte_survey | long int |
| tbl_soil_group | byte | tbl_soil_name | byte |
| | | tbl_site | byte |
| tbl_soil_name | byte | No child tables | |

Italics = Parent table primary key is auto number NB no dictionaries use auto number, IDs need to be entered manually.

5.3 Indexing

To ensure high performance during querying and forms display, all fields from a table containing more than 100 rows that were used in a WHERE statement or ORDER BY statement in any query or form were indexed. Similarly all fields that act as foreign keys were also indexed. All tables except tbl_weightings have a primary key which by definition are a unique index and have not been listed. When a pattern of use has been established for the system it may well be worth reappraising the indexing to fine tune the performance.

The following fields had indexes set on them. To see the full information on the indexing, open the tables in design mode and select the menu | view | indexes.

tbl_dict_invertebrate_species.invertebrate_name

tbl_dict_macrophyte_species.macrophyte_name

tbl_fish_survey.site_id

tbl_fish_survey_species.fish_survey_id

tbl_fish_survey_species.fish_id

tbl_fish_survey_species.fish_population_status_code

tbl_invertebrate_survey.site_id

tbl_invertebrate_survey.season_code

tbl_invertebrate_survey_species_main.invertebrate_species_id

tbl_invertebrate_survey_species_main.invertebrate_survey_id

tbl_invertebrate_survey_species_supp.invertebrate_species_id

tbl_invertebrate_survey_species_supp.invertebrate_survey_id

tbl_macrophyte_survey.alternative_river_type_code

tbl_macrophyte_survey.cb_type_code

tbl_macrophyte_survey.original_survey_type_code

tbl_macrophyte_survey.river_type_code

tbl_macrophyte_survey.site_id

tbl_macrophyte_survey.survey_date

tbl_macrophyte_survey.surveyor

tbl_macrophyte_surv_physical_depth.depth_area

tbl_macrophyte_surv_physical_depth.depth_id

tbl_macrophyte_surv_physical_depth.macrophyte_survey_id

tbl_macrophyte_surv_physical_margin.margin_area

tbl_macrophyte_surv_physical_margin.margin_id

tbl_macrophyte_surv_physical_margin.macrophyte_survey_id

tbl_macrophyte_surv_physical_NH_data.feature_area

tbl_macrophyte_surv_physical_NH_data.feature_id

tbl_macrophyte_surv_physical_NH_data.macrophyte_survey_id

tbl_macrophyte_surv_physical_width.width_area
tbl_macrophyte_surv_physical_width.width_id
tbl_macrophyte_surv_physical_width.macrophyte_survey_id

tbl_macrophyte_surv_physical_chan_feature.channel_feature_area
tbl_macrophyte_surv_physical_chan_feature.channel_feature_id
tbl_macrophyte_surv_physical_chan_feature.macrophyte_survey_id

tbl_macrophyte_surv_physical_chan_substrat.channel_substrate_area
tbl_macrophyte_surv_physical_chan_substrat.channel_substrate_id
tbl_macrophyte_surv_physical_chan_substrat.macrophyte_survey_id

tbl_macrophyte_surv_species_cvr_aquat.macrophyte_id
tbl_macrophyte_surv_species_cvr_aquat.macrophyte_survey_id
tbl_macrophyte_surv_species_cvr_aquat.macrophyte_percent_cover_code_3_point
tbl_macrophyte_surv_species_cvr_aquat.macrophyte_percent_cover_code_5_point
tbl_macrophyte_surv_species_cvr_aquat.macrophyte_percent_cover_code_3_point
+ relative_cover_code
+ macrophyte_id

tbl_macrophyte_surv_species_cvr_bank.macrophyte_id
tbl_macrophyte_surv_species_cvr_bank.macrophyte_survey_id
tbl_macrophyte_surv_species_cvr_bank.macrophyte_percent_cover_code_3_point
tbl_macrophyte_surv_species_cvr_bank.macrophyte_percent_cover_code_5_point
tbl_macrophyte_surv_species_cvr_bank.macrophyte_percent_cover_code_3_point
+ relative_cover_code
+ macrophyte_id

tbl_macrophyte_surv_vegetation.macrophyte_survey_id
tbl_macrophyte_surv_vegetation.vegetation_type_id

tbl_river.catchment_id

tbl_site.country_code
tbl_site.distance_to_source
tbl_site.drift_geology_code
tbl_site.river_name
tbl_site.site_altitude
tbl_site.site_name
tbl_site.slope
tbl_site.soil_group_code
tbl_site.solid_geology_code
tbl_site.river_name
+ site_name

6 Queries

There are 19 queries in the code .mdb. These are all sources for record sets in sub forms or reports that require data from multiple tables. To get the detailed information for each query, display the query in design view in Access.

7 Forms

Nearly all the forms in the code .mdb have some visual basic code embedded in them. This may be a simple routine to close the form when an exit button is clicked, or may be a complex routine such as validating a NGR when it is updated. Except for the most trivial functions where it is obvious what the code does, all the code is well commented. For details on exactly what each function does, see the source code and its comments.

To view the visual basic code select View | Design view from the menu. The form then changes to the design view. From this design view, select View | Code from the menu. This will bring up the visual basic window. If the project explorer is not already displayed select View | Project explorer from the visual basic window menu. This will display the project explorer window, which will enable you to select the form, report or module to view the code for.

When you add or update data in a field on a form, in order for the field to get flagged as updated you need leave the field by moving the cursor into another field. This is important for fields such as NGR that get validated, as the validation is not done until the field is flagged as updated.

To write a record to the database you have to either move to a different record or explicitly save the record from the menu or toolbar. If you update site data, then update data in several different surveys for that site, the site updates do not get written to the database until you move to another site record.

Most of the forms have an associated help form. This is simply a pop up form containing a series of text labels giving information and help for the form.

7.1 Startup page form

This form contains several functions and subroutines, most are simple subroutines that run when buttons are clicked i.e. Exit and Reconnect.

The subroutine cmdLoadData_click runs when the load data button is clicked. This code loads data from the data capture spreadsheet.

The subroutine form_load runs when the form is loaded, this checks the user status and sets up the enabling/disabling of buttons and displays the status information.

7.2 View form

This is the main form for viewing the data. It is a hierarchical form consisting of a form and 2 levels of linked sub forms. This enables all the information for a site to be viewed on a single form by having tabbed pages displaying the required information for the different surveys for the site.

There are several simple subroutines to re-sort the order of the sites based on the users choice, and also to go to a specific site record within the record set. These also ensure that the sort order is always by river name when the form is loaded.

There are also subroutines to validate the NGRs when they are updated, and to calculate an easting and northing from the NGRs. These are in the basValidateNGR code module

7.2.1 Macrophyte survey sub form

This is a linked sub form to display the macrophyte survey and associated information. It is on the macrophyte surveys tabbed page and has 3 tabbed sub pages for the associated macrophyte cover data, physical data and images. This information is displayed in linked sub forms on the appropriate tabbed sub page. It has several simple subroutines, two of which display a details form for the river type when the button next to the river type field is clicked.

The classify river subroutine is run when the classify river button is clicked and is simply a routine to call the classify river function in the VB modules. The most complex subroutine is the routine to export the list of macrophytes to an Excel worksheet when the export to Excel button is clicked. This is a well commented subroutine, so to see exactly how it works, refer to the comments in the code. Finally it has a subroutine to open a report for the displayed survey when the 'Print report' button is clicked.

7.2.1.1 Macrophyte cover sub forms

The aquatic and bank cover sub forms are very similar. They display the macrophyte and cover information for the list of macrophytes found on the survey. There are subroutines that run when the column title labels are double clicked. These re-sort the macrophyte list based on whatever the title clicked was and if run a second time they reverse sort the list.

7.2.1.2 Physical data sub forms

There are 6 sub forms for the physical data information. Each displays the data value and the cover code of the physical features found on the survey.

7.2.1.3 Images sub form

The ability to add or link images to the database is a new feature in the current version.

This is the sub form that displays the data from the images table.

The image field is an OLE object in the table and the control on the form is a bound object frame. MS Access help has detailed information about storing, displaying and editing images.

When storing images there are several points to note.

1. There are two distinct ways to store the image.

You can simply store a link to the original image file (a linked image). This takes up very little space in the database, but means that all the images have to be in the same pathname in all installations of the software which may not be practical. Alternatively you can store a copy of the image within the database (an embedded object). This

takes up roughly the same amount of space in the database as the file size of the image, but does mean that the images are part of the database and will get distributed with it. This is probably the more practical way of storing images.

2. If you store the image as a compressed file such as a .jpg you will not get a thumbnail preview of the image on the form, a filename is displayed in the image control instead and the user has to open the file by double clicking on the control. The only way to get a thumbnail preview is to store the image as an uncompressed bitmap .bmp file. This takes a huge amount of space, typically 5 MB per image and storing a bitmap as a linked image makes no difference to this. If you just want thumbnail images you should resize the image file to a smaller size (8cm x 6cm takes up about 100kb). If you want bigger images you have to balance the image size and display quality against the file size.
3. It is not possible to set a linked image to read only. The user can always open the image editor and update the image. If using linked images rather than embedded, write protection must be done at file or directory level.
4. Images take up huge amounts of space. If you want them to be visible as an image rather than an icon on the image sub form, Access has to store them as a bitmap. Even if they are linked images rather than embedded, they still take up huge amounts of space.

7.2.2 Invertebrate survey sub form

This is a simple linked sub form to display the invertebrate survey information. It is on the invertebrate surveys tabbed page and has two embedded sub forms to display the list of invertebrates found for the main and supplementary surveys. It has a subroutine to open a report for the displayed survey when the 'Print report' button is clicked.

7.2.3 Fish survey sub form

This is a simple linked sub form to display the fish survey information. It is on the fish surveys tabbed page and has an embedded sub form to display the list of fish found and their population status. It has a subroutine to open a report for the displayed survey when the 'Print report' button is clicked.

7.3 Query form

This is a form to allow the user to interactively build up query criteria and then run the query. There are 5 tabbed pages which display sub forms to do various different types of query. Each tabbed page has a 'Run query' button which when clicked, runs a subroutine to build up the SQL and run the query.

The SQL query gets built up in 3 parts.

1. The fields list which is hard coded for each query type and remains constant.
2. The criteria which are built up based on the values entered in the criteria fields.
3. The order by statement which is hard coded for each query type and remains constant.

The river queries tab has 2 different sub forms on the page that can be viewed and hidden using the 3 toggle buttons at the top.

The sites by macrophyte query is a slight variation in that it is building up several similar queries that are combined using AND or OR criteria. It's 'run query click' subroutine does this by calling the subroutine criteria to build up the query for each macrophyte and appending them to the SQL code built up so far. The subroutines are well commented, so for details on exactly how the SQL is built up see the source code.

When the query form is first displayed it runs queries that specifically fetch no records into all the sub forms. This takes about 45 seconds when the form is first displayed, but once this has been done, the queries are then very quick.

7.4 River form

This is a simple form to view river data that is displayed by clicking on the button on the site form.

8 Reports

There are 3 reports in the code .mdb which can be run by clicking on the 'Print report' button on the appropriate survey tab on the view data form. This runs a subroutine which has a call to the DoCmd.OpenReport routine to open the report in preview mode with the current survey id as the filter to produce the report. The reports are structured as a report with a series of linked sub reports to show detailed information.

The record sources for the reports and sub reports are queries which are all named to be very obvious which report they populate. To get the detailed information for each report, display the report in design view in Access.

9 Visual Basic modules

There are 8 visual basic code modules in the code .mdb. These contain visual basic functions and subroutines that are called when certain events occur in the forms i.e. pressing a button or after updating a field.

They are named to be fairly self-explanatory. Each function has a header describing what it does and the code is well commented. For a detailed description of each, refer to the source code header and comments.

9.1 BasCheckLinkedTables

This module contains 5 functions required to check that the data file that the application is linked to exists, and to reconnect to another data file if required.

9.2 BasCheckUser

This module contains a function to check whether the user belongs to the Admins group.

9.3 basClassifyAll

This module contains a subroutine to go through all the surveys in the database and run the classify river routine for each one in turn.

9.4 basClassifyRiver

This module contains a function to go through all the macrophytes found during the survey, and define a river type and alternative river type for the survey.

9.5 basConvertNGR

This module contains a function to calculate a 0,0 based easting and northing using the values in the NGR_conversion table from a NGR.NB the constants used to convert from the Irish national grid were copied from the old code, they were not checked.

9.6 basGetFilePathName

This module contains a function to display an input dialog box for the user to enter a filename into.

9.7 basValidateNGR

This module contains a function that validates the format of a NGR.

9.8 basValidateQueryCriteria

This module contains a function to check that the criteria for a numeric field in a query is in the correct format i.e. nn, =nn, <nn, >=nn etc.

10 River classification

Since the old system was written there has been more work done on the classification system used to classify the rivers based on the macrophyte assemblage found during a survey. This has been implemented as a spreadsheet, and part of the project was to include this new classification methodology into the application. The classification uses a decision tree structure to divide the river type into 2 possible groupings at each decision level. It does this at several levels, thus dividing the rivers into one of 10 'Types'.

At each level the relative cover code of each macrophyte found in the survey is checked, against an entry in the Weightings table. This table gives a score value if certain macrophytes have a cover value of 1 (present) or 3 (present as the dominant species). These scores may be + or - values, and the total score for all the species is calculated for each level. Based on this total score, the decision is made as to which grouping to put the river in and which decision node in the tree to use at the next level. This decision would normally be based on whether the total score is ≥ 0 or < 0 , but to allow fine tuning of the decision process the threshold table contains a threshold value which the total score is compared against rather than the fixed value of 0.

If a total score is much greater or less than the threshold value, you can be confident about the decision, but as the score gets closer to the threshold, a greater degree of uncertainty is present. To allow for this a second classification is done to give an 'Alternative river type'. On this second pass, if the total score is not greater or less than a tolerance value difference from the threshold, the first time (the highest level most significant decision) this occurs the alternative decision is made and the opposite node at the next level is used to give an alternative river type.

By having these Threshold and tolerance values in the tolerance table, the methodology can be fine tuned without having to edit the program code.

This new classification method has only been implemented down to type level, so to classify from there down to sub type level, the code has been copied in from the old system and the typing continues by using the old code.

The classification is done by the function ClassifyRiver() in the visual basic module basClassifyRiver. The code is well commented and documented so for details on exactly how the function works see the source code.

NB: There is a zero length string (“”) entry in the river type dictionary table. This has to be there as the ‘classify all’ routine enters a zero length string into the river type field if it has not got data to do a classification for a survey i.e. when there are no macrophytes. This is because ADO does not allow you to enter a NULL.

11 Data Export

11.1 GIS output

Previous versions of the database outputted only a single point NGR without further information. The GIS output has now been considerably enhanced so that the output file contains additional data such as site name, river type and geology.

The GIS output file is a simple comma delimited text file (.txt). This format should be acceptable to most GIS packages including MapInfo and the ESRI family of products and will be convertible to point data using the Create Points... function in the GIS package.

The following fields are included: -

| | | |
|--------------|-------------------------|----------------|
| NGR, | Surveyor, | Sample length, |
| Country, | Survey type, | Altitude, |
| River, | % vegetated, | Slope, |
| Site name, | River type, | Solid geology. |
| Survey date, | Alternative river type, | |

If the field is empty, an empty string is written.

E.g.

```
"SH434787","Wa","CEFNI"," 2 CEFNI [Llangwyllog]","16/07/1980",  
"HOLMES, N","JNCC3","","Vc","VIIc","0.5","46","5","Caradoc"
```

Quoted strings have been used because some of the fields may contain commas.

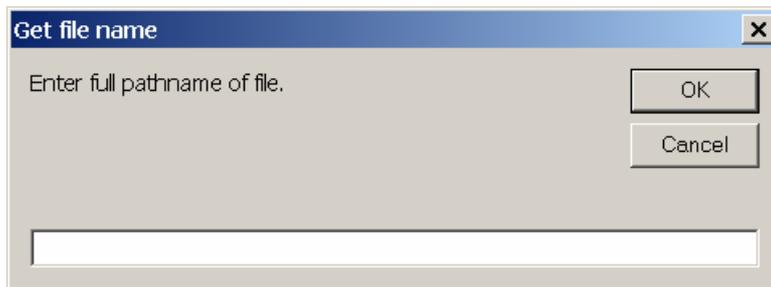
The fields that can be outputted into a GIS file are not user selectable. If more detailed GIS output is required, or a custom layer containing different fields, a bespoke query can be created or export the file as an Excel spreadsheet (see below).

11.2 Excel spreadsheet output

The new version of the database now includes a means of exporting data as an Excel spreadsheet in order to simplify data analysis. This has been implemented as a method of the Export to excel button (subroutine cmdExport_Click()) that is run when the button is clicked. It is well commented and for full details of how it works, refer to the source code header and comments.

NB: As the subroutine creates an Excel application instance to write the data to, you need Excel 2000 to be installed on your computer.

The subroutine brings up a standard windows input box to allow the user to enter the filename of the .xls spreadsheet to save the data to.



It then saves the data to the spreadsheet file with the first row as a header row e.g.:

| Aquatic macrophyte ID | Aquatic macrophyte name | Aquatic rel cover | Aquatic %cover | Aquatic comment | Bank macrophyte ID | Bank macrophyte name | Bank rel cover | Bank %cover | Bank comment |
|-----------------------|-------------------------|-------------------|----------------|-----------------|--------------------|----------------------|----------------|-------------|--------------|
| 147 | Equisetum arvense | 1 | 1 | | 789 | Elodea canadensis | 2 | 2 | |
| 594 | Equisetum fluviatile | 2 | 1 | | 146 | Epilobium spp. | 1 | 1 | |
| 148 | Equisetum palustre | 1 | 1 | | 147 | Equisetum arvense | 2 | 1 | |

12 Data capture spreadsheet

12.1 General notes

This section explains the technical working of the data capture spreadsheet. For information on how to use the spreadsheet to collect data ready for loading into the database see the 'Data capture spreadsheet' manual.

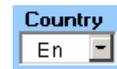
The Site, Survey, Physical data, and Macrophyte data worksheets have values in 'hidden' cells. These cells have been hidden by setting the text colour the same as the background colour. If you select an area, or the whole worksheet, the background colour changes slightly and the hidden values show. A full explanation of these cells is given under each of the worksheet descriptions below.

To enable the data load routine to work with older versions of ODBC there is a line of hidden numeric 1 values in line 2 of the Site, Survey and Macrophyte data worksheets above the hidden cells. This is because some versions only look at the first 8 rows to determine what data type it will use for the column when treating the spreadsheet as an ADO data source. If there is no entry in the first 8 rows you cannot be sure it will

treat it as the required data type, so putting a hidden value in ensures the correct data type.

There are 3 different types of data entry field.

1. Free format data entry fields that you can type anything into. These may be limited to specific types of data i.e. numeric entries, dates etc.
2. List box data entry fields that are restricted to a list of values. These have a drop down arrow at the right hand edge that only appears when you enter the field. You should select a value from the drop down list, although you can type the data directly into the field. The entry is validated against the list, so if you enter a value that is not in the list, you will get an error message and you will need to delete the invalid entry and enter a valid value.



3. Drop down combo box controls that overlay the data entry field. Combo boxes have a permanent drop down arrow on their right hand edge. It is possible to tab into the underlying data entry field and type data into it, or to type data into the combo box rather than selecting the value from the drop down list.

DO NOT DO THIS, as it is not possible to validate the data entered in drop down combo boxes. This is the only way of entering invalid values that will cause the data load to fail.

If you enter a wrong value into a drop down combo box, either select the correct value to overwrite it, or select the blank from the top of the list to delete it. **Do not try to correct it by typing directly into the box.**

Combo boxes are actually control objects that overlay the cell, so you are not actually seeing the data cell itself. The value in the combo box control is mirrored in the underlying cell. The reason for using combo boxes rather than list boxes is that you cannot populate a list box list from a list of values on a different worksheet. To use list boxes we would have had to have the dictionary values spread around the different worksheets rather than all on one, which would have made checking and keeping them, synchronised with the database dictionaries more awkward.

The spreadsheet is only suitable for capturing survey values using the JNCC 3 point survey method. Any surveys done as a JNCC 5 point or other method will have to be recorded on paper and entered manually.

If in the future it is decided that there is a requirement for capturing 5 point survey data, a separate spreadsheet will need to be developed, and the data load routine in the database will need to be modified.

The worksheets are protected to prevent unintentional changes. If you want to unprotect them, go to Tools/Protection/unprotect sheet in the excel menus.

The protection is not password protected, it is assumed that the people that use the spreadsheets can be trusted not to disable the protection and change the spreadsheet.

12.2 The worksheets

12.2.1 Read me sheet

This is a very simple description and user guide for the spreadsheet.

12.2.2 Site sheet

The site sheet has hidden lists of values at the top of columns M and N. These are where the list boxes for 'Site surveyed previously?' and 'Grid ref. and altitude point' get their list of valid values from.

Columns O to S also have hidden numeric fields.

Row 2 has a line of numeric 1 values to force the columns to a numeric data type as explained above.

The other cells contain values that mirror the values in visible data cells in the spreadsheet. The reason for doing this is twofold.

1. When opening the spreadsheet as an ADO data object for data loading, a column has to be a specific data type, you cannot have mixed data types in a column. This gets set automatically based on the data type found in the column, so by having text titles over the numeric cells the column gets treated as text. By mirroring the contents in another column it guarantees it is read as the correct data type.
2. By using Excel's IF() function, the mirrored cell can be set to a specific value if the visible cell is blank, or to the visible cell value if a value has been entered. In the data load routine, if this specific value is found, it is loaded into the database field as NULL. In the Site worksheet the values are set to -1 if there is no entry in the visible cell, because 0 could be a valid entry that you want loaded as 0.

There are some fields that are for info only rather than relating to database fields. I.e. contractor name, contractor reference etc. There is also a field to identify where the grid ref and altitude measurements refer to on the site, this could usefully become a database field in future versions.

The site surveyed previously will probably be set to ? in most cases as the contractor won't know if the site has been surveyed before. It is really there to remind the data loader to check.

Hopefully the new naming standard should enable the data loader to spot duplicates at data load time, but being a free format text field it won't be 100% reliable as minor typos could fool it.

It would be best if CCW/EH/SNH pre allocated site names prior to the surveyors entering the data. Without the database, the contractor won't necessarily know whether there are other surveys for the river, so will be unable to allocate numbers, u or l etc. to the site name.

NB there is no software check for duplicate site names, they are not a keyed field so it is possible to have duplicates.

12.2.3 Survey sheet

Column N has a mirror of the survey date cell which will get read as a date data type. Columns N to R have hidden numeric fields.

Row 2 has a line of numeric 1 values to force the columns to a numeric data type as explained above.

The other cells contain values that mirror the values in visible data cells in the spreadsheet. The reason for doing this is twofold.

1. When opening the spreadsheet as an ADO data object for data loading, a column has to be a specific data type, you cannot have mixed data types in a column. This gets set automatically based on the data type found in the

2. By using Excel's IF() function, the mirrored cell can be set to a specific value if the visible cell is blank, or to the visible cell value if a value has been entered. In the data load routine, if this specific value is found, it is loaded into the database field as NULL. In the Survey worksheet the values are set to -1 if there is no entry in the visible cell, because 0 could be a valid entry that you want loaded as 0.

12.2.4 Physical data sheet

This has hidden fields containing the code for the physical attribute in the column to the left of the attribute.

All the physical cover codes are entered from drop down combo boxes.

12.2.5 Macrophyte data sheet

Columns X to AH have hidden numeric fields.

Row 2 has a line of numeric 1 values to force the columns to a numeric data type as explained above.

The other cells contain values that mirror the values in visible data cells in the spreadsheet. The reason for doing this is twofold.

1. When opening the spreadsheet as an ADO data object for data loading, a column has to be a specific data type, you cannot have mixed data types in a column. This gets set automatically based on the data type found in the column, so by having text titles over the numeric cells the column gets treated as text. By mirroring the contents in another column it guarantees it is read as the correct data type.
2. By using Excel's IF() function, the mirrored cell can be set to a specific value if the visible cell is blank, or to the visible cell value if a value has been entered. In the data load routine, if this specific value is found, it is loaded into the database field as NULL. In the Macrophyte data worksheet the values are set to 0 if there is no entry in the visible cell, because 0 is not a valid entry.

All the macrophyte cover codes are entered from list boxes that take their list of valid values from the list at the top of the worksheet.

12.2.6 Dictionaries sheet

This has a list of all the dictionaried items that are used on the spreadsheet, including a full list of the rivers. It is where the drop down combo boxes get their lists of values from.

The lists for Physical data (Depth, Width, Margin, Channel feature, Channel substrate, NH data) and the list of Macrophytes are for information only. They are not used for lists of values for combo boxes as they reflect the fields on the sheet rather than the possible entries for a field.