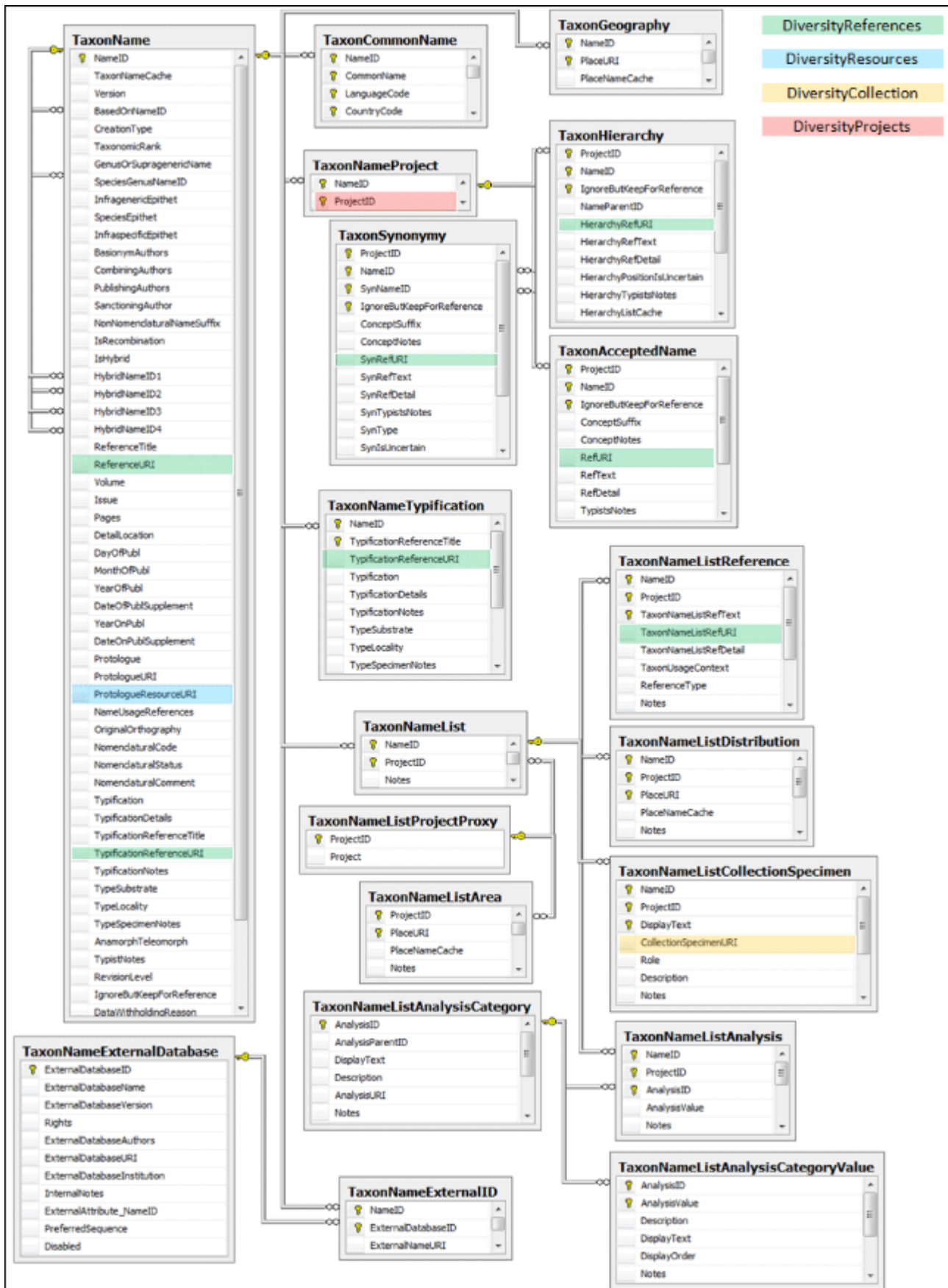# Diversity Workbench

The Diversity Workbench is composed of components for building and managing biodiversity information, each of which focuses on a particular domain. Each component can provide services to the other components. DiversityTaxonNames can link data to the modules DiversityProjects, DiversityResources, DiversityGazetteer, DiversityAgents and DiversityReferences as illustrated in the image below. DiversityTaxonNames itself is a source for DiversityCollection.

The modules communicate with each other to provide their services for the other modules.

Modules

The Diversity Workbench is a set of components for building and managing biodiversity information, each of which focuses on a particular domain. To see the open connections to other modules, choose **Connection** - **Module connections** .... from the menu. Each component can provide services to the other components. The optional links to other modules within the Diversity Workbench are shown in the image below.

TaxonName
- NameID
- TaxonNameCache
- Version
- BasedOnNameID
- CreationType
- TaxonomicRank
- GenusOrSupragenericName
- SpeciesGenusNameID
- InfragenericEpithet
- SpeciesEpithet
- InfraspecificEpithet
- BasionymAuthors
- CombiningAuthors
- PublishingAuthors
- SanctioningAuthor
- NonNomenclaturalNameSuffix
- IsRecombination
- IsHybrid
- HybridNameID1
- HybridNameID2
- HybridNameID3
- HybridNameID4
- ReferenceTitle
- ReferenceURI
- Volume
- Issue
- Pages
- DetailLocation
- DayOfPubl
- MonthOfPubl
- YearOfPubl
- DateOfPublSupplement
- YearOnPubl
- DateOnPublSupplement
- Protologue
- ProtologueURI
- ProtologueResourceURI
- NameUsageReferences
- OriginalOrthography
- NomenclaturalCode
- NomenclaturalStatus
- NomenclaturalComment
- Typification
- TypificationDetails
- TypificationReferenceTitle
- TypificationReferenceURI
- TypificationNotes
- TypeSubstrate
- TypeLocality
- TypeSpecimenNotes
- AnamorphTeleomorph
- TypistNotes
- RevisionLevel
- IgnoreButKeepForReference
- DataWithholdingReason

TaxonCommonName
- NameID
- CommonName
- LanguageCode
- CountryCode

TaxonGeography
- NameID
- PlaceURI
- PlaceNameCache

TaxonNameProject
- NameID
- ProjectID

TaxonHierarchy
- ProjectID
- NameID
- IgnoreButKeepForReference
- NameParentID
- HierarchyRefURI
- HierarchyRefText
- HierarchyRefDetail
- HierarchyPositionIsUncertain
- HierarchyTypistsNotes
- HierarchyListCache

TaxonSynonymy
- ProjectID
- NameID
- SynNameID
- IgnoreButKeepForReference
- ConceptSuffix
- ConceptNotes
- SynRefURI
- SynRefText
- SynRefDetail
- SynTypistsNotes
- SynType
- SynIsUncertain

TaxonAcceptedName
- ProjectID
- NameID
- IgnoreButKeepForReference
- ConceptSuffix
- ConceptNotes
- RefURI
- RefText
- RefDetail
- TypistsNotes

TaxonNameTypification
- NameID
- TypificationReferenceTitle
- TypificationReferenceURI
- Typification
- TypificationDetails
- TypificationNotes
- TypeSubstrate
- TypeLocality
- TypeSpecimenNotes

TaxonNameListReference
- NameID
- ProjectID
- TaxonNameListRefText
- TaxonNameListRefURI
- TaxonNameListRefDetail
- TaxonUsageContext
- ReferenceType
- Notes

TaxonNameList
- NameID
- ProjectID
- Notes

TaxonNameListDistribution
- NameID
- ProjectID
- PlaceURI
- PlaceNameCache
- Notes

TaxonNameListProjectProxy
- ProjectID
- Project

TaxonNameListArea
- ProjectID
- PlaceURI
- PlaceNameCache
- Notes

TaxonNameListCollectionSpecimen
- NameID
- ProjectID
- DisplayText
- CollectionSpecimenURI
- Role
- Description
- Notes

TaxonNameListAnalysisCategory
- AnalysisID
- AnalysisParentID
- DisplayText
- Description
- AnalysisURI
- Notes

TaxonNameListAnalysis
- NameID
- ProjectID
- AnalysisID
- AnalysisValue
- Notes

TaxonNameExternalDatabase
- ExternalDatabaseID
- ExternalDatabaseName
- ExternalDatabaseVersion
- Rights
- ExternalDatabaseAuthors
- ExternalDatabaseURI
- ExternalDatabaseInstitution
- InternalNotes
- ExternalAttribute_NameID
- PreferredSequence
- Disabled

TaxonNameExternalID
- NameID
- ExternalDatabaseID
- ExternalNameURI

TaxonNameListAnalysisCategoryValue
- AnalysisID
- AnalysisValue
- Description
- DisplayText
- DisplayOrder
- Notes

Legend:
- DiversityReferences
- DiversityResources
- DiversityCollection
- DiversityProjects

Module related entry

The Diversity Workbench is a set of components for building and managing biodiversity information, each of which focuses on a particular domain. Each module provides services for the other modules. To use the service of a module, you need access to the database of the module and optionally the module application placed in your application directory. Entries related to an external module have a standard interface in the main form. There are 2 states of this interface:

1 - the value is only set in the local database with no connection to the remote module

Reference: [Alexopoulos & Henney 1971. Myxomycetes from Texas. II Ad]

In this state, you can either type the value or select it from the values that are already available in the database. To get a list of the available values type the beginning of the value (you may use wildcards) and click on the [▼] button. If you want to set a relation to the remote module, click on the [button]. A form will open where you may select an entry from the foreign database.
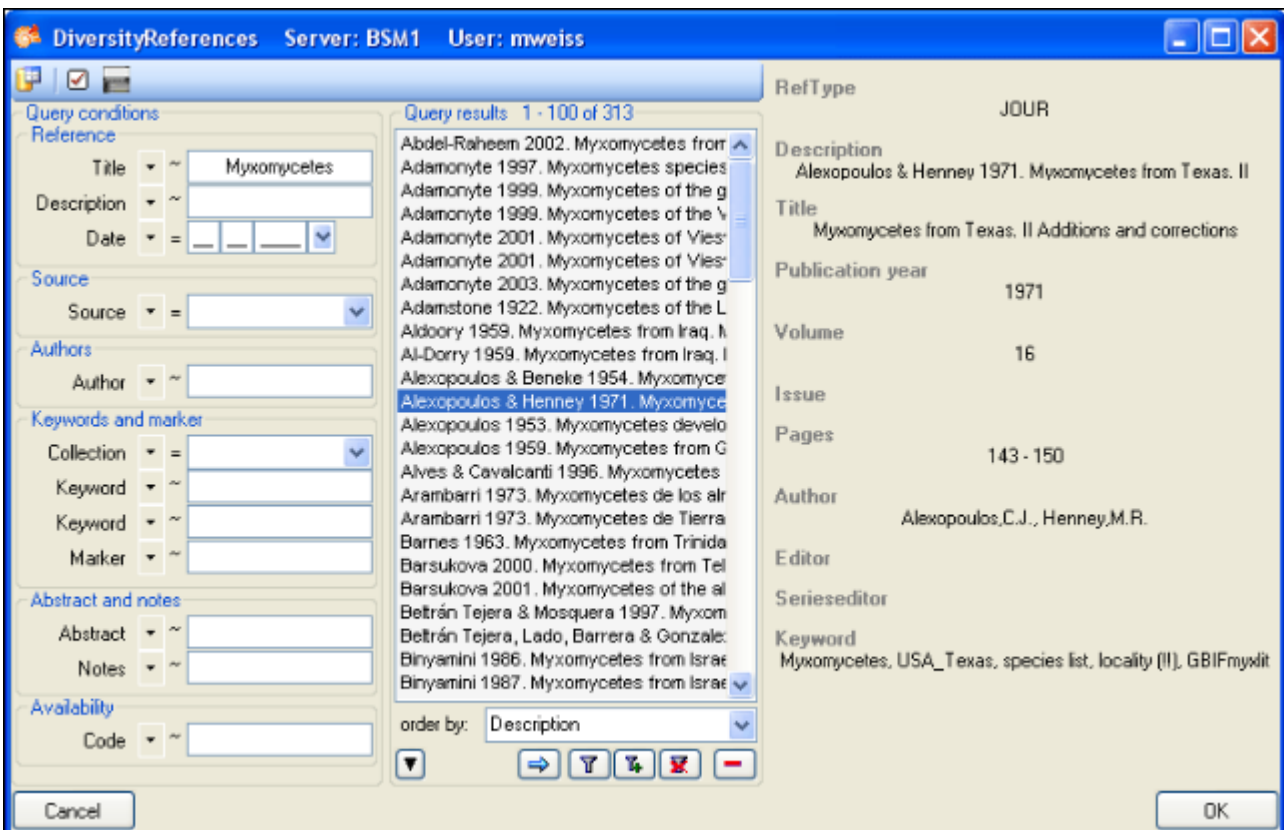
2 - the value is related to the remote module

Reference: [Alexopoulos & Henney 1971. Myxomycetes from Texas. I]

If the value has a relation to the remote module, the interface will appear as shown above. To release the connection to the remote module click the [X] button. If you need further information about the value, click the [button]. This will open a form, showing an overview of the related value.

RefType	
	JOUR
Description	
	Alexopoulos & Henney 1971. Myxomycetes from Texas. II Additions and corrections. Southwest Nat 16: 143 - 150.
Title	
	Myxomycetes from Texas. II Additions and corrections
Publication year	
	1971
Volume	
	16
Issue	
Pages	
	143 - 150
Author	
	Alexopoulos,C.J., Henney,M.R.
Editor	
Serieseditor	
Keyword	
	Myxomycetes, USA_Texas, species list, locality (!!), GBIFmyxlit

If the client application of the module is available you can inspect the details of the entry.

To start the client application of the remote module, just click on the button.

Projects

Projects with the DiversityWorkbench are located in the database DiversityProjects. The access to the data is organized via projects. The projects a user has access to can be selected in the query options.

To edit the project list use the project area. Use the ☐ and ✖ buttons to add or delete projects from the list.

If there are projects, to which you have no access, an additional list will appear, showing these projects (see below).
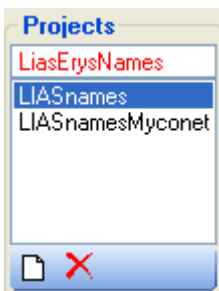
The information related to opinions of the editors are allways restricted to a certain project (see Database).

Reference

Details about References are stored in the module DiversityReferences. If you choose an entry from the references stored in DiversityTaxonNames you can choose one of the entries in the picklist.

Gen. Pl.

If the module DiversityReferences is available, you can access this module by clicking on the button.

For access to the projects from other modules like DiversityTaxonNames, you need the application **DiversityReferences.exe** in your application directory. To use this application you need access to the Database DiversityReferences.

Specimen

If you are connected to a DiversityCollection database (see menu Connection -> Module connections ... . Of several databases, the first will be used), the program will search in this database for type specimen, that are linked to the current name. These will be listed as shown below.
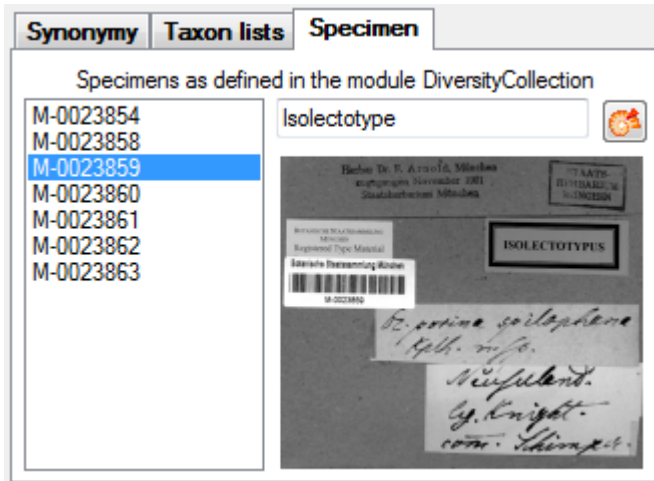
To see a larger view of the image, just double click on it. To see further informations about the specimen, click on the button.

DiversityTaxonNames

DiversityTaxonNames is the module for administration of taxonomic names within the Diversity Workbench. For access to the services provided by other components the application needs several files in the application directory:

DiversityTaxonNames.exe , DiversityWorkbench.dll , DiversityTaxonNames.chm

To start the application double click on DiversityTaxonNames.exe.

If you start the client for the first time, you will not be connected to any database, indicated by the icon in the left upper corner as shown below.
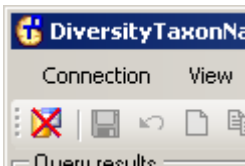
See Login for details about the connection to the database.

Installation

To run DiversityTaxonNames, you need the database and the client. The image below gives an overview of the software you need.
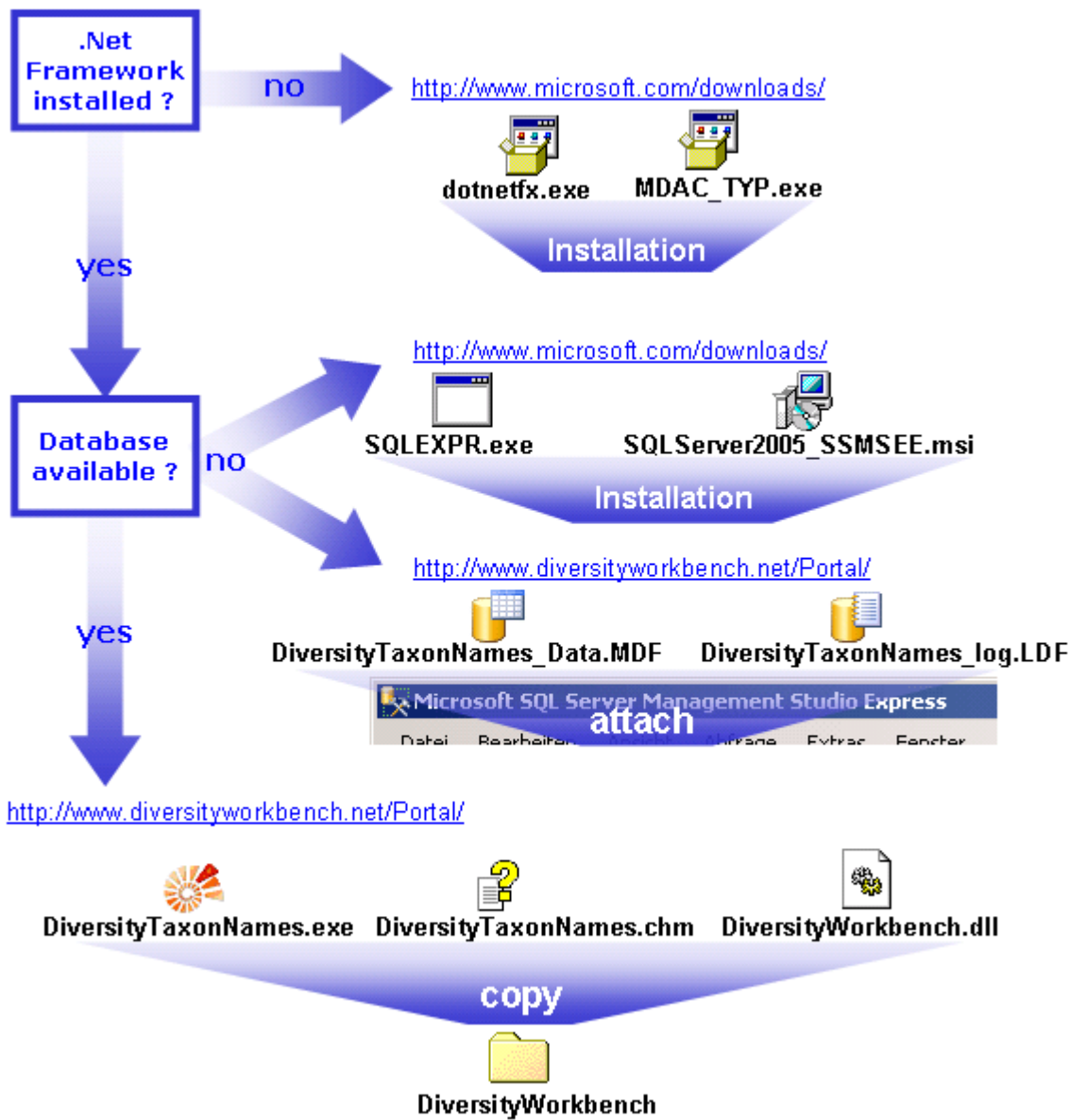
Database

To run the database you need MS SQL-Server 2005 or later.

Client

The client is based on the .Net framework from Microsoft. If not already present, you have to install:

Microsoft .NET Framework Version 3.5

(see http://www.microsoft.com/downloads/ for the latest versions)

Copy all files extracted from DiversityTaxonNames.zip into one directory.

Database

For the installation of a local database see the section <u>Installation of the database</u>.

User administration

The permissions of users in the database are set via roles and the access to the projects.
To set the permissions choose **Administration** - **User** ... from the menu. A form will open
where you can change the permissions of the user. The first list shows the user accounts that
are available in the database but have no access to any of the projects. To synchronize this
list with the Diversity Workbench module DiversityUsers click on the
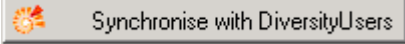
[Synchronise with DiversityUsers] button. To create a new SQL-Server User, click on the 👤
button. To create a new Windows user with access to the database resp. allow an existing
user the access to the database use the Microsoft SQL Server Management Studio (see the

installation section for further details). To permit access to a project click on the [>] button.

To remove a user from this list use the [<] button.

The area **Permission of user** shows the projects a user has access to and the roles of the
user within the database. To move users between the lists Projects that are available for a
user and Projects that are not avialable for a user use the [>] and [<] buttons. To create a
new project click on the 🔓 button. If you use the Diversity Workbench
module DiversityProjects, you can create a new project there and user the synchronize
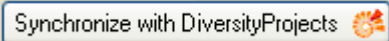
button [Synchronize with DiversityProjects]. To change the roles of a user use the [▲] and [▼]
buttons. Underneath the project list for a user you find the list of the Roles of the user. This
list can be changed by using the and buttons. Underneath the roles list you find the list with
the roles available in the database. On the left side of this list the permissions of the selected
role are listed and on the right side the members of this role.
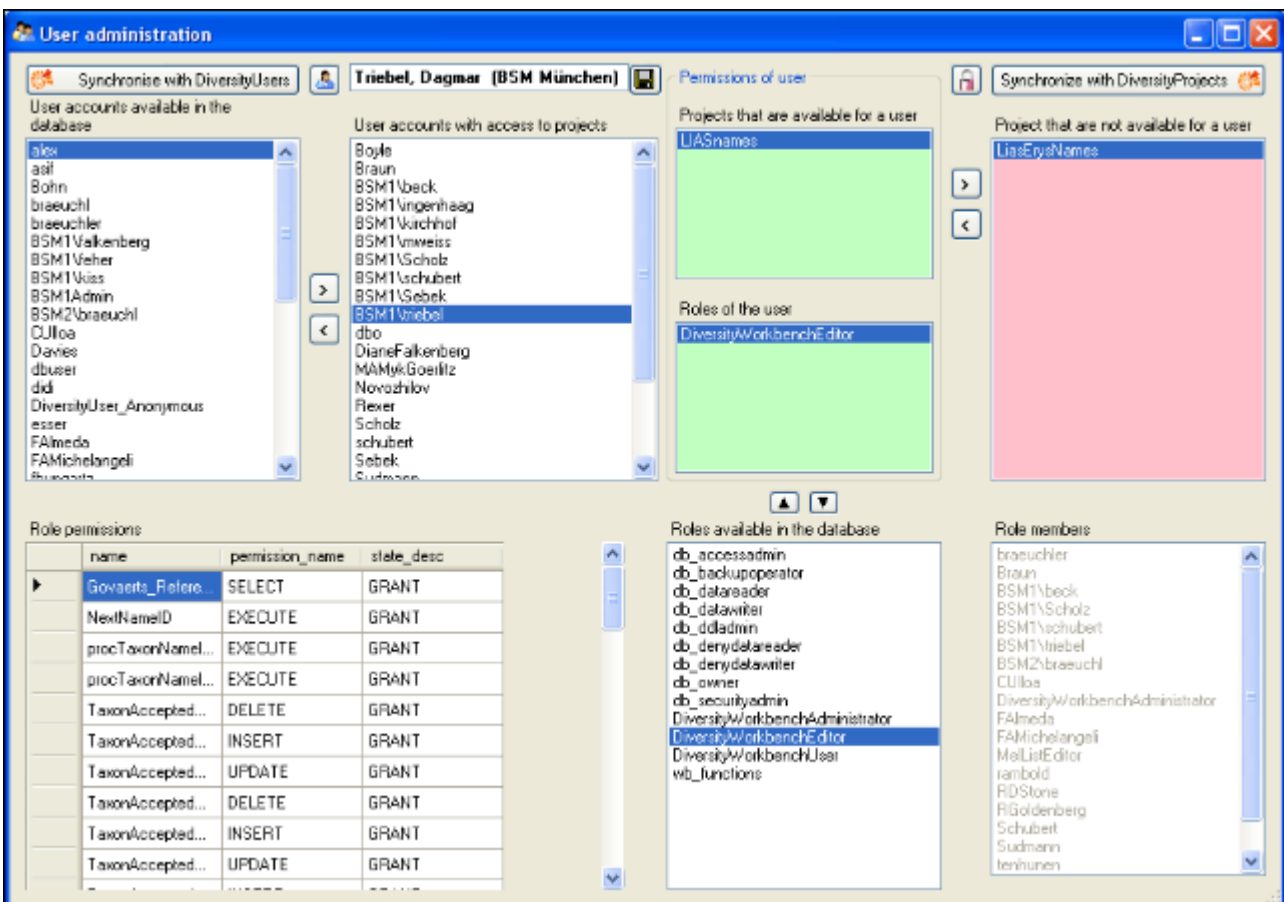
The permissions of the roles are shown in the lower left part of the form.

The role **DiversityCollectionUser** can see the data within the permitted projects, but can not change anything.

The role **DiversityCollectionTypist** can edit a part of the user defined data.

The role **DiversityCollectionEditor** can change the user defined parts of the data.

The role **DiversityCollectionCurator** has the same rights as the group DiversityCollectionEditor and in addition can handle transactions, i.e. shipments of specimen between collections.

The role **DiversityCollectionAdministrator** can delete data, edit the contents of internal tables, change user permissions etc.

Database - access

If you start the client for the first time, you will not be connected to any database, indicated by the icon in the left upper corner as shown below.
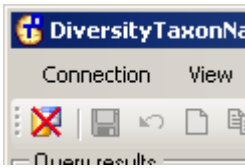
If you are not connected to a database this will be indicated by the icon of the

connection button. If you are connected to a database this is indicated by the icon . To access any database, you must specify the server where the database is located. For the configuration of this connection choose **Connection**, **Database...** from the menu or click on

the button.
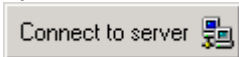
Database name, IP-address and Port

A dialog will open, where you can specify the name or IP-adress and port number of the server and the authentication mode. You can either choose Windows authentication (see left image below) or SQL-Server authentication (see middle image below).
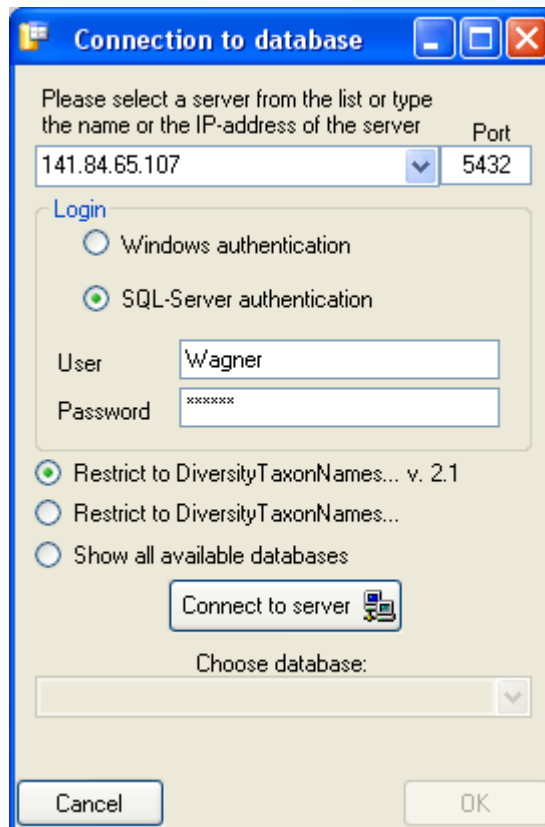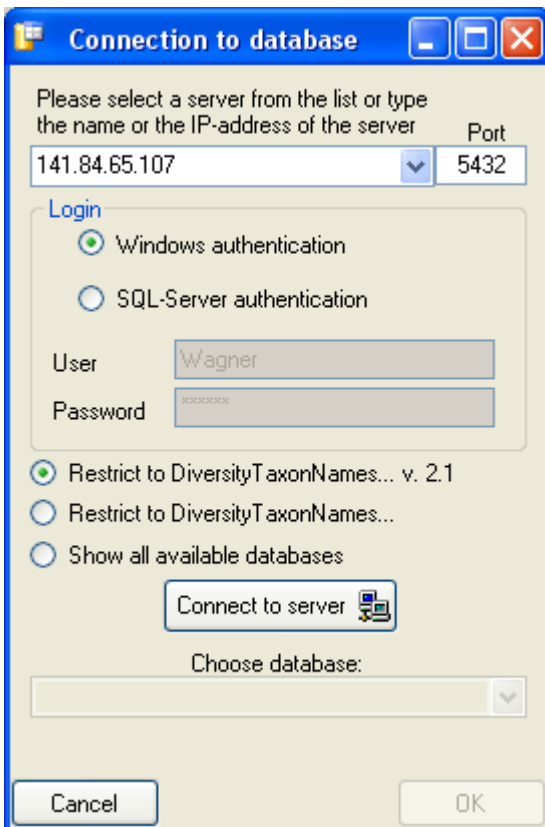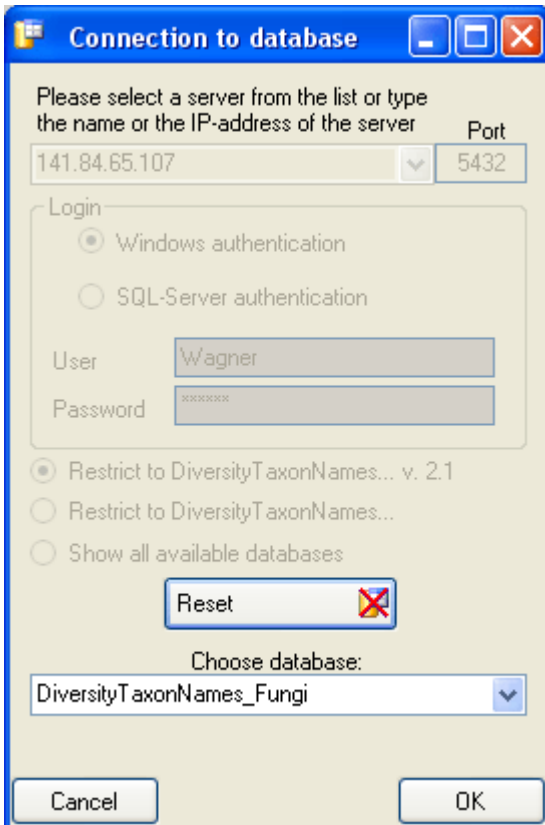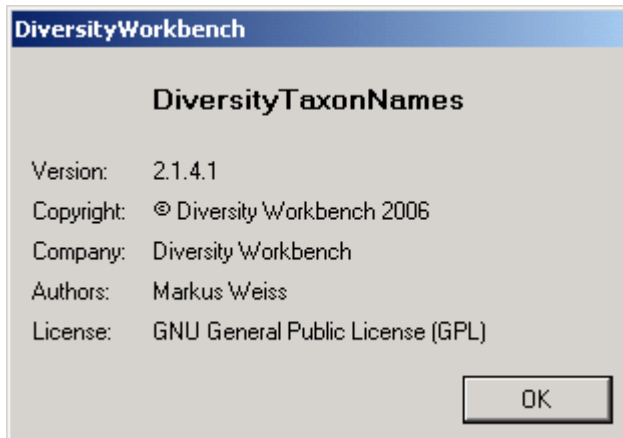
The standard port number for SQL-Server is 1433 and will be set as a default. If the database server is configured using a port different from that port, you must give the port number in the field Port. Click on the [Connect to server] button to connect to the server. If the connection informations are valid, you can choose a database from the server from the combobox at the base of the form (see right image above). To restart the connecting process click on the [Reset] button.

Version

For information about the version of the client application choose **Help**, **Info…**

```
DiversityWorkbench

        DiversityTaxonNames

Version:    2.1.4.1
Copyright:  © Diversity Workbench 2006
Company:    Diversity Workbench
Authors:    Markus Weiss
License:    GNU General Public License (GPL)

                              [  OK  ]
```

The current version in the example above is 2.1.

the numbers following are buildnumber and revision.

Feedback

For any wishes or problems with the database that you want to send to the administrator you can do this under the menu **Help - Feedback**. Move to the window you want to be included in your feedback and click **ALT - PRINT** to create a screenshot. Then open the feedback form under the menue **Help - Feedback ...** . A form as shown below will open. To enter the screenshot, click on the [Insert image] button. The description is entered in the field **Description** .
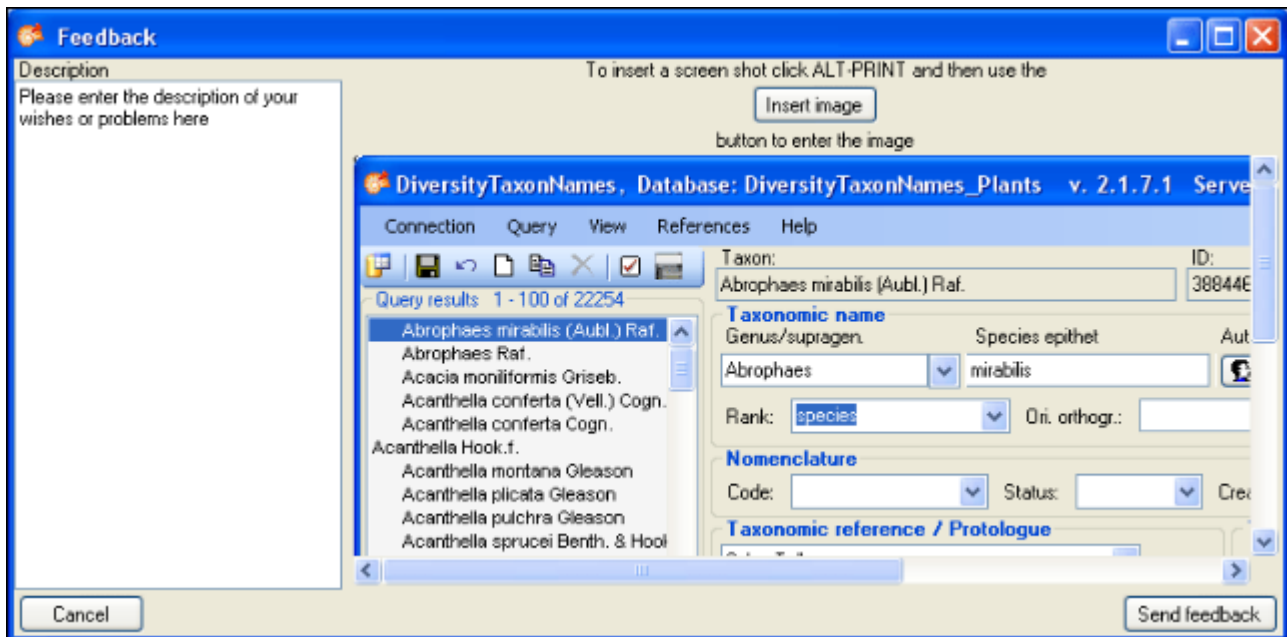

To send the feedback click on the [Send feedback] button.

Export

To export the taxonomy, including the hierarchy and the synonymy, select the corresponding taxon, choose **Data -> Export -> Export hierarchy and synonymy...** from the menu.

To see the whole hierarchy, select the desired node in the hierarchy tree and click on the ⛁ button besides the hierarchy.
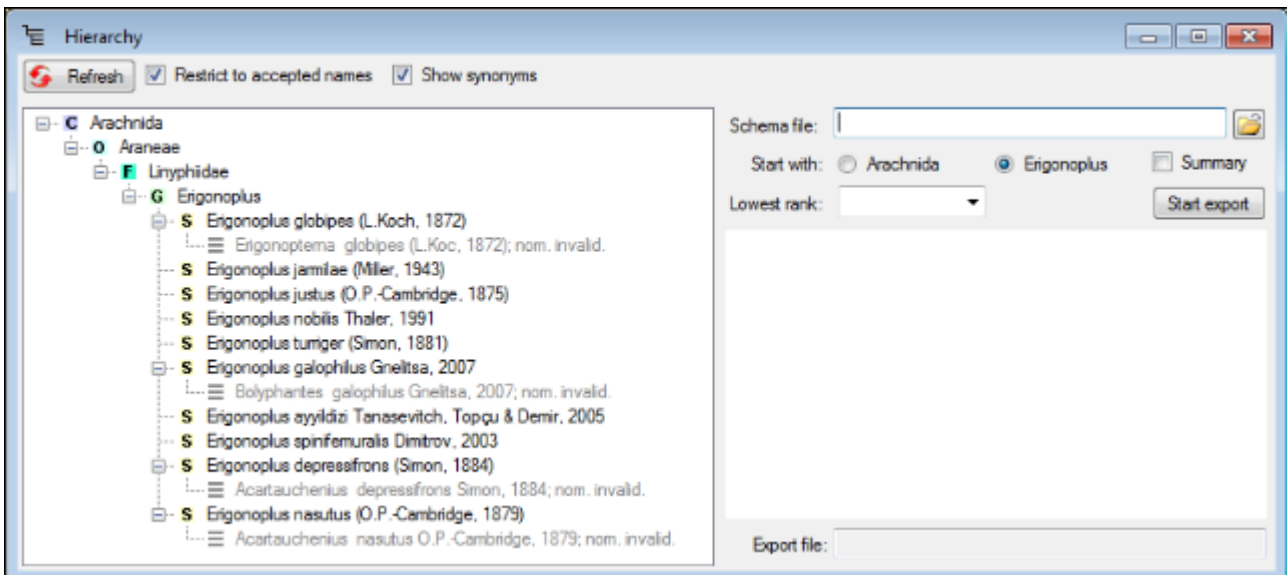
A window as shown below will open
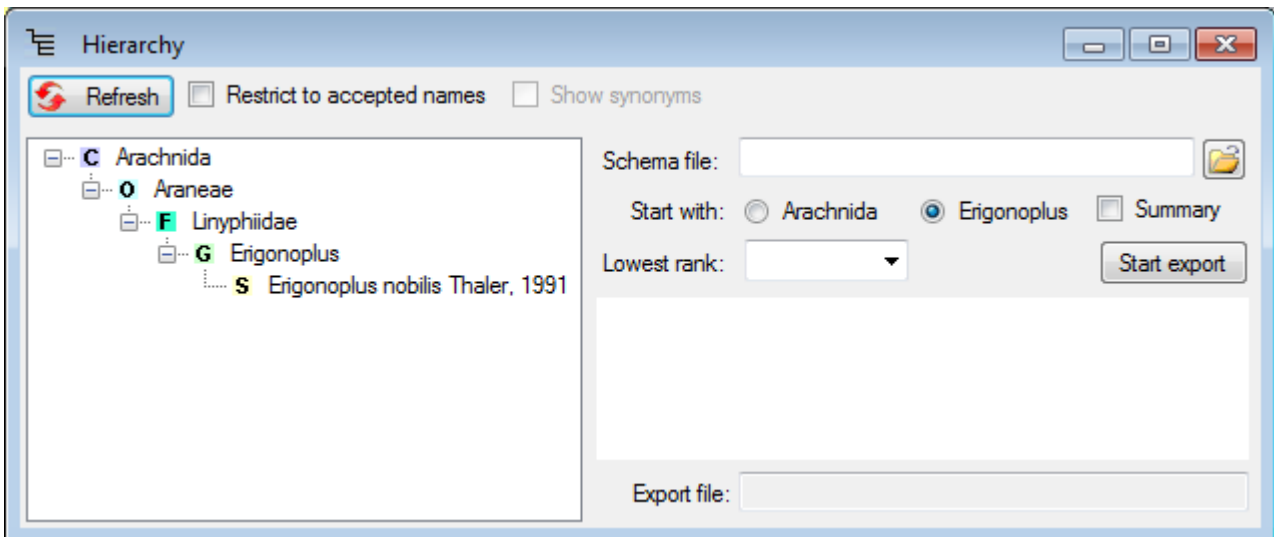
The data you in the export will depend upon the taxon you selected in the main form. So if e.g. you want to export the names within a genus, select the genus in the main form.
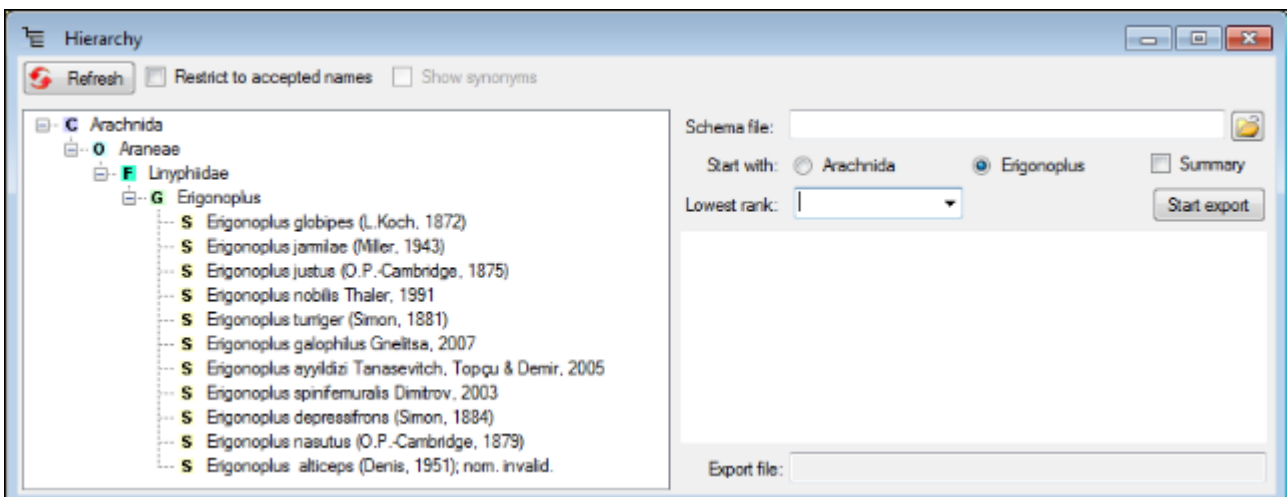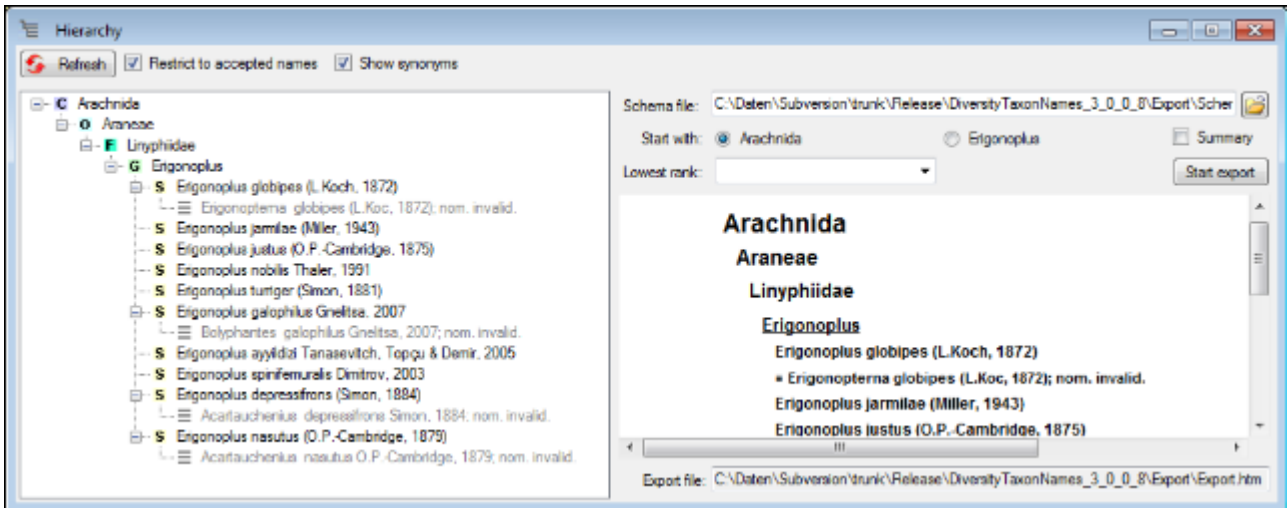

If you want to include the synonyms, check **Restrict to accepted names** and **Show synonyms** and click on the ⟳ Refresh button (see below).

 If you start the export without any schema file, you will get the data strutured as xml (see below).
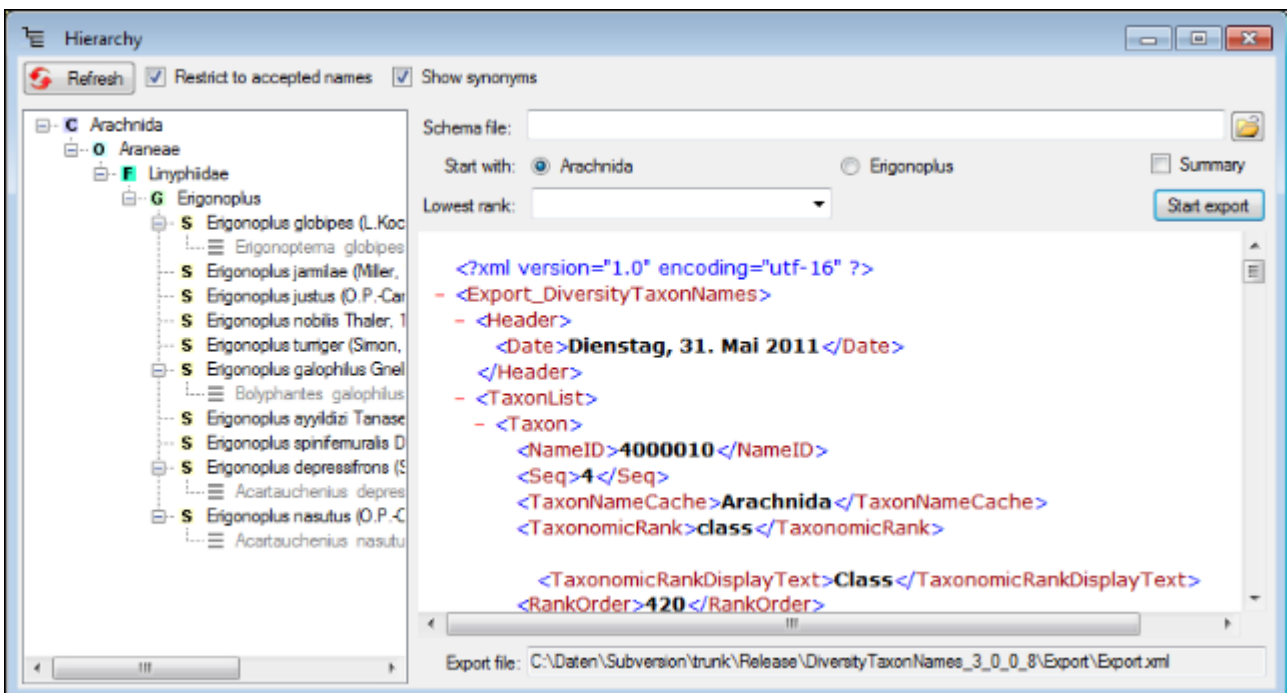

To format the data, specify the schema file containing your preferred format as shown below.

You can export the data including all higher taxa (see above) or starting with the taxon for which you selected the export (see below). mao format the data, specify the schema file containing your preferred format as shown below.
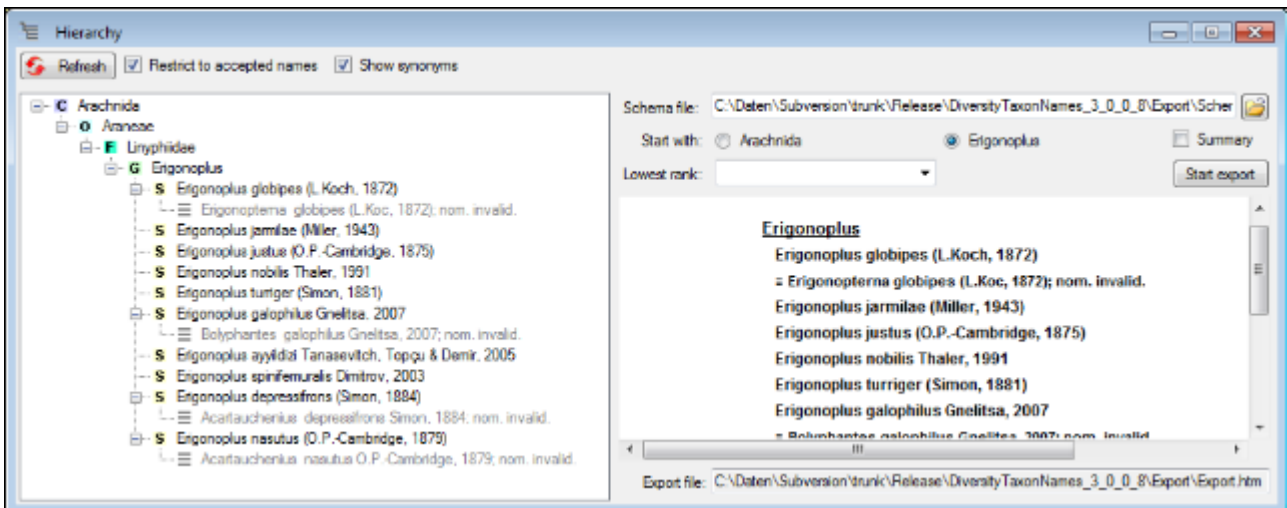
To export only the higher taxa, you can restrict the lowest rank to e.g. the genus as shown below.
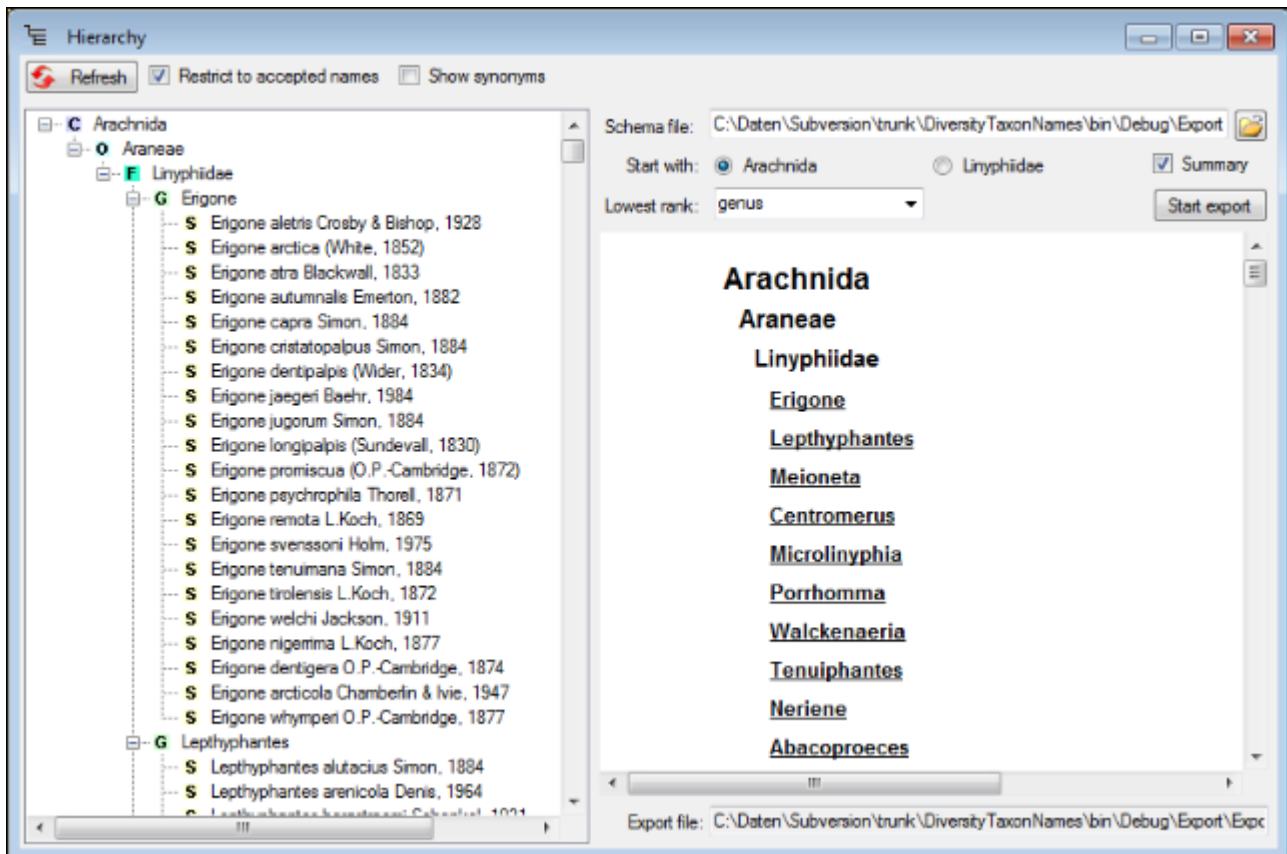
If you choose the option **Summary**, the numbers of the differnet hierarchical levels will be listed at the bottom of the report (see image below)
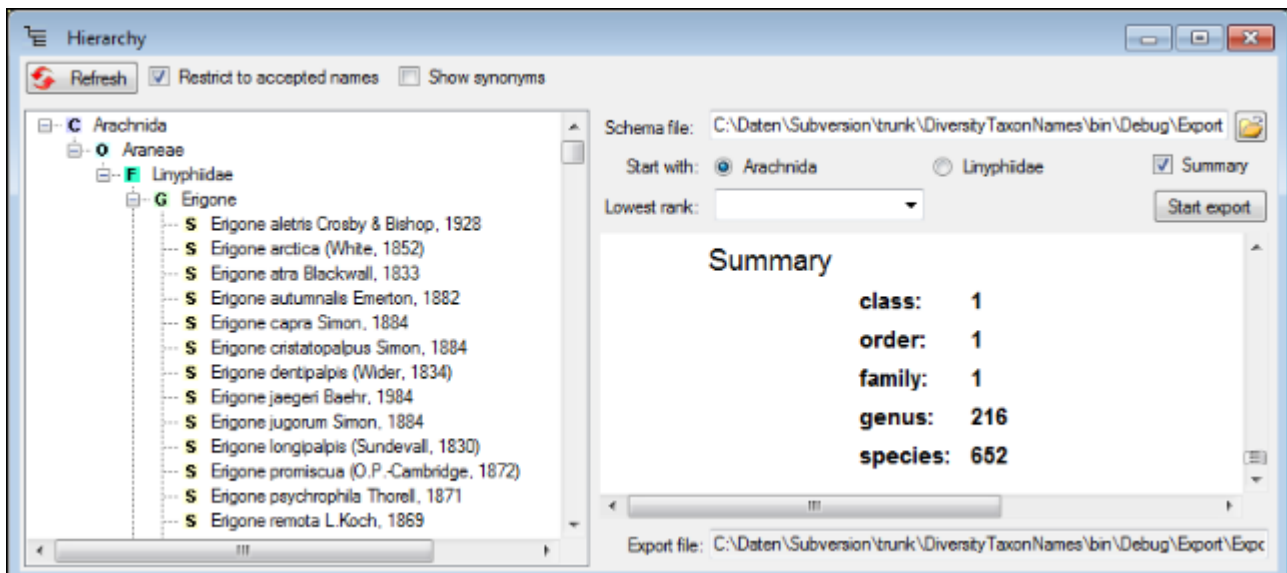
Export list

To export lists, including information about the distribution or the synonymy, choose **Data -> Export -> Export taxon list ...** from the menu. A window as shown below will open, where you can choose the taxonomic list and the project where the synonymy of the taxa is documented.

Choose a Schema file and the options for your export. Than click on the Create the taxon list to start the export. The resulting html file (see below) will be named according to the choosen schema file.
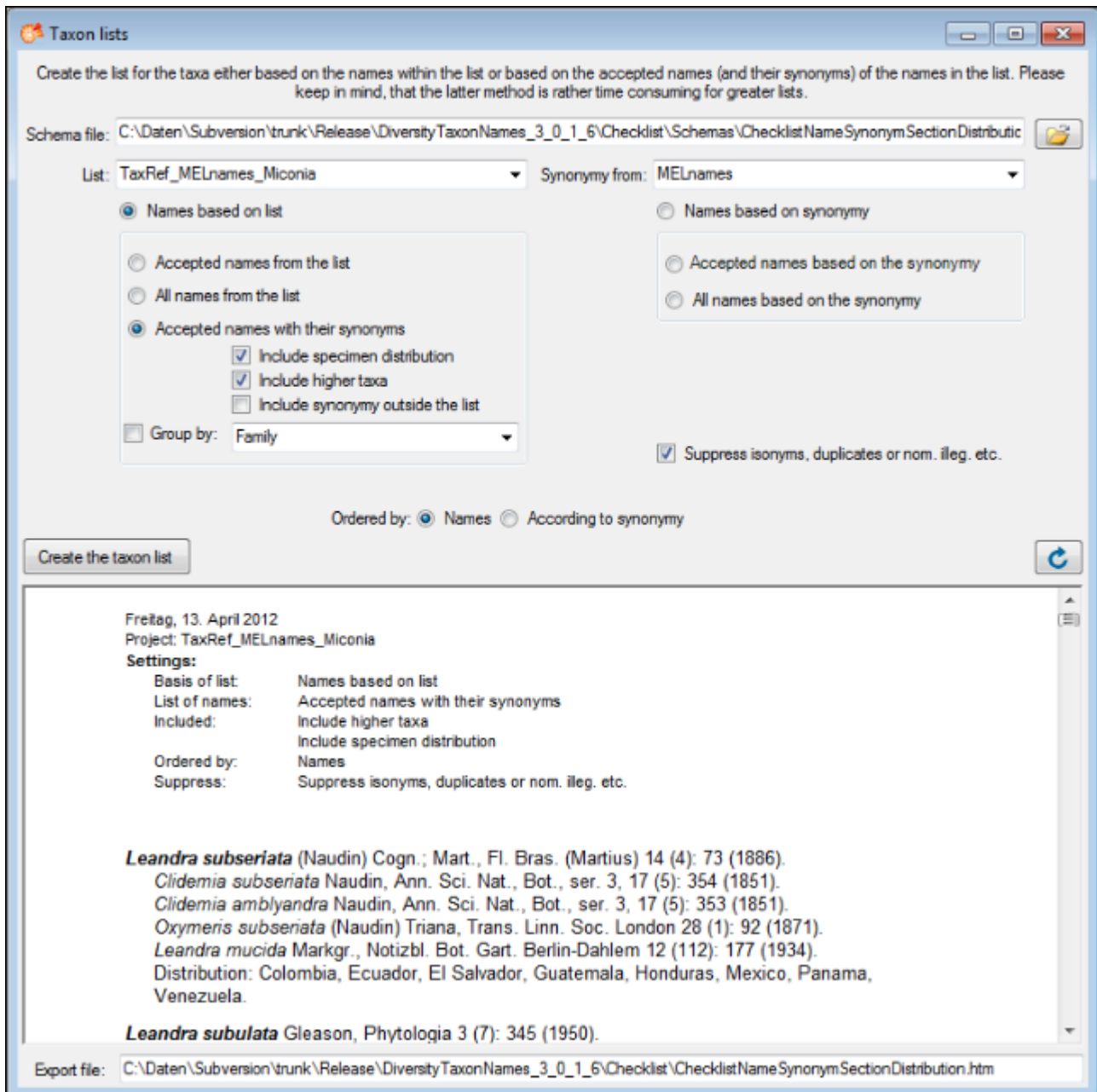
If you miss to choose a schema file, the original XML file will be shown (see below). To convert this into a html file, choose a schema file and click the ↻ button.

Taxon lists

Create the list for the taxa either based on the names within the list or based on the accepted names (and their synonyms) of the names in the list. Please keep in mind, that the latter method is rather time consuming for greater lists.

Schema file:

List: TaxRef_MELnames_Miconia ▼   Synonymy from: MELnames ▼

◉ Names based on list   ○ Names based on synonymy

- ○ Accepted names from the list   ○ Accepted names based on the synonymy
- ○ All names from the list   ○ All names based on the synonymy
- ◉ Accepted names with their synonyms
  - ☑ Include specimen distribution
  - ☑ Include higher taxa
  - ☐ Include synonymy outside the list
- ☐ Group by: Family ▼   ☑ Suppress isonyms, duplicates or nom. illeg. etc.

Ordered by: ◉ Names ○ According to synonymy

[Create the taxon list]   ↻

```
<?xml version="1.0" encoding="utf-16" ?>
- <Export_DiversityTaxonNames>
  - <Header>
      <Date>Freitag, 13. April 2012</Date>
      <Project>TaxRef_MELnames_Miconia</Project>
      <ProjectTitle>TaxRef_MELnames_Miconia</ProjectTitle>
      <BaseURI>http://id.snsb.info/TaxonNames_Plants/</BaseURI>
      <Analysis />
    - <Settings>
        <BasisOfList>Names based on list</BasisOfList>
        <ListOfNames>Accepted names with their synonyms</ListOfNames>
      - <Includes>
          <Included>Include higher taxa</Included>
          <Included>Include specimen distribution</Included>
        </Includes>
        <OrderedBy>Names</OrderedBy>
        <SuppressIsonyms>Suppress isonyms, duplicates or nom. illeg.
```

Export file: C:\Daten\Subversion\trunk\Release\DiversityTaxonNames_3_0_1_6\Checklist\Checklist.xml

Import

To import taxonomic names, choose **Data → ⧉Import taxonomic names ...** from the menu. A window as shown below will open. Choose the encoding of your datasource, that means if your datasource was encoded with ASCII or a unicode format. To ensure a correct import of special signs please provide resp. convert your data in one of the available formats. You must choose a **Project**, in which your data should be imported. You may choose a **Taxon list** and a **Datasource** and if the names should be **imported as accepted**.

To start the analysis of the data, just click on the 📂 button to open the source file. If you want to re-analyse a file with new settings, click on the ↻ button. To send a feedback, use the ✉ button. If the names should be linked to higher taxa, **check the Link taxa to genera if present** option. The names can be compared with existing names within the current project or the whole database. If you compare the names with the whole database, you can insert a link for the current project for identical names that are missing in the project.
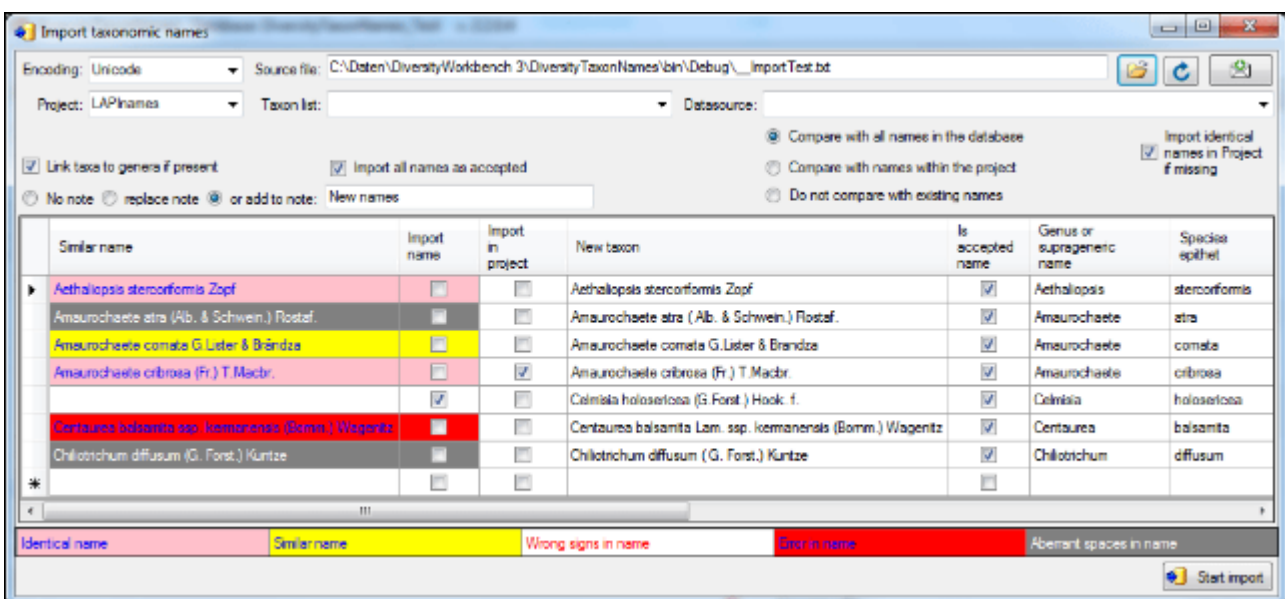
Your names will be listed as shown in the image above. If an error (e.g. special signs converted with the wrong encoding)

or a similar name

or an identical name

is found these will be marked as shown above and the names of your list will not be imported unless you check the according field (second column - Import name). If you changed one of the options, click on the **Requery** button to see the result. To start the import, click on the according button ⧉**Start import**.

Menu

Overview of the menu in DiversityTaxonNames

Connection

	Database ...	Choose one of the databases available on the server. Only those databases will be listed to which the user has access permission
	Module connections ...	Edit the connections to the other modules within the Diversity Workbench.
	Transfer previous settings	Transfer the settings for IP-Address and port of the server, name of the database, login etc. of a previous version of the client to the current version.
	Quit	Quit the application and stop all processes started by the application

Query

Show query	Show resp. hide the query area
Show ignored data	If data that were set on ignore should be visible
Predefined queries	This dialog will only appear if predefined queries are available. The submenu then lists the names for the predefined queries

Lists

	List view ...	Show the data in a spreadsheet
	Edit taxa in list ...	Edit the taxa in a spreadsheet

Administration

	Analysis categories ...	Administration of the analysis categories used within the taxon lists
	Data sources ...	Administration of the data sources of the taxonomic names
	Image description template ...	Administration of the description template for the images within a project
	List User ...	Administration of the logins and user
	Logins ...	Administration of the logins and user
	Predefined queries ...	Administration of the predifined queries
	Syonchronize projects ...	Administration of the projects
	Rename database	Rename the current database
	Set published address	Setting the address published for links by other modules
	Taxon lists ...	Administration of the taxon lists
	Tools	
	Maintenance ...	Maintenace functions within the DiversityTaxonNames
	Documentation ...	Documentation functions for the database

References

Edit references ...	Edit the local list of the references
Edit journals ...	Edit the local list of the journals

Data

Export hierarchy and synonymy ...	Export the taxonomic names with their hierachy and synonoymy
Export taxon lists ...	Export the taxon lists as XML / HTML files
Import taxonomic names ...	Import taxonomic names from text files
Backup database ...	Backup of the whole database
Export as CSV (bcp)	Export data of the whole database as csv files

Help

|	
Manual	Opens the online manual
Feedback ...	Opens a window for sending feedback
Feedback history ...	Opens a window for browsing former feedback
Statistics	Show the statistics within a project
Info	Show the version and corresponding information

Manual

The online manual DiversityTaxonNames.chm must be placed in your application folder, together with the application DiversityTaxonNames.exe and the library DiversityWorkbench.dll.To get information to any topic in the application DiversityTaxonNames and open this manual, just click on the field you need information about and press F1.

Maintenance

Hierarchy

In DiversityTaxonNames the hierarchy of a taxon is represented by the relation to a parent taxon, e.g. Rosa canina -> Rosa. To enable the search within the whole hierarchy is cached as a text string e.g. (... Rosales - Rosaceae - Rosa - ...). After changes in the hierarchy this string should be updated , the cached values must be updated. Choose **Administration -> ⚒Maintenance ...** from the menu. A window as shown below will open. Choose the project that should be updated an optional a separator for the taxa. Then click the start button to perform the update.
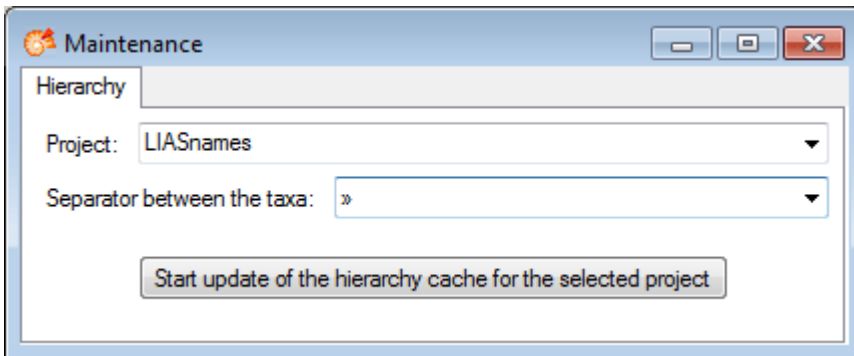
Projects and lists

In DiversityTaxonNames names of projects and list are cached according to the original data in DiversityProjects. If the original names are changed, you can use the maintenance to synchronise the local names with the original data (see image below). Click on the **Load projects** resp. **Load lists** button to find all entries with names different from the originals. To update the names in the local cache, click on the **Start update** button.
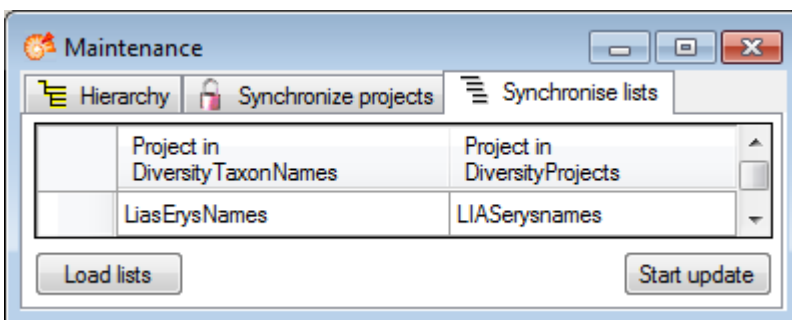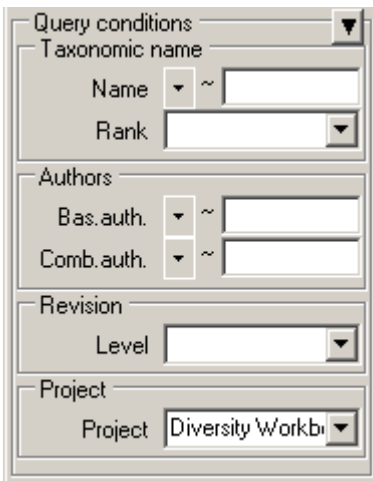
Query

The application provides two main options to search for a taxonomic name. The results of any search will be listed in the listbox on the left side of the main form (selected names). To choose one of the found names just click on it.
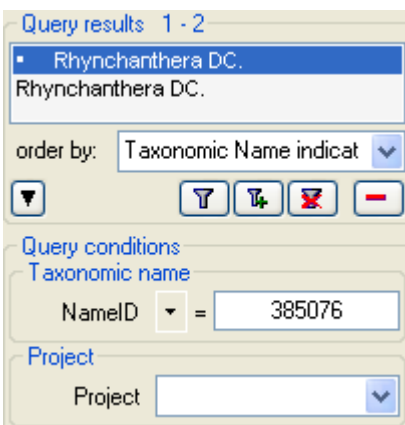
Direct query

For a direct search of a name enter the search conditions in the fields provided in the "Query conditions". You may alter the visible field as described under options . You can change this arrangement using the ![button] ![button] button to place the query options on the left side of the item list.

The available operators for text values are = for exactly equal entries or ~ for similar entries. Here you can choose "%" (any sequence of characters) and "_" (one character) as wildcards. For more details of SQL-wildcards refer to the manual or SQL-Server.

If you omit to select a project and the synonymy for a name is defined in several projects, you will get an entry in the resultset for every project as shown in the example below.

If you want to hide the Query conditions area, click on the ![button] button. The Query conditions area will be hidden. To show it again click on the ![button] button. To change the displayed fields for searching specimen click on the ![button] button to change the query options .

After you have specified your search phrase click on ![button]. This starts the search for any name matching your phrase. If you want to add the searched name to those allready listed then

click on ![icon]. If the list of items is longer than your maximal number of returned items you can browse the next items with the ![icon] button. To move back to the previous block of items click on the ![icon] button. If you want to remove entries from the selected list, choose them and click on the ![icon] button. This will not delete the data from the database, but remove them from your query result.

Within the query options you have several possibilities to specify your search restriction. Use the drop down button to change between the operator. The available operators are shown in the table below.

Operator	Meaning	Example
Text		
~	search for an entry like ...	Pinus s[iy]lvestris % (you can use wildcards)
=	search for an entry exactly equal to ...	Pinus silvestris L.
≠	search for an entry not like ...	Pinus s[iy]lvestris % (you can use wildcards)
∅	search for an entry where a value is missing ...	
•	search for an entry where a value is present ...	
-	search for an entry between ... and ...	2000 - 2005
|	search within a list of entries, separated by "|" or return	2000 | 2003 | 2005
Numeric		
=	search for an entry exactly equal to ...	2006
<	search for an entry lower than ...	2006
>	search for an entry bigger than ...	2006
-	search for an entry between ... and ...	2000 - 2005
∅	search for an entry where a value is missing ...	
•	search for an entry where a value is present ...	
|	search within a list of entries, separated by "|", space, tab or return	2000 | 2003 | 2005
Date		
=	search for an entry exactly equal to ...	20.3.2006
<	search for an entry lower than ...	20.3.2006
>	search for an entry bigger than ...	20.3.2006
∅	search for an entry where the date is missing ...	
•	search for an entry where the date is present and complete ...	
Hierachy		
=	search for an entry exactly equal to ...	M-Fungi
≠	search for an entry that is not equal to ...	M-Fungi

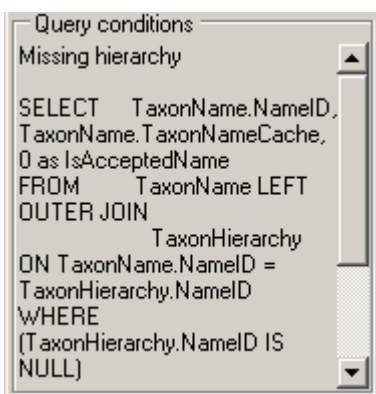| ∅ | search for missing entry ... | |
•	search for present entry ...	
Δ	search including childs in a hierarchy ...	M-Fungi
XML		
/	Search for entries containing a given XML node	settings
¬	Search for entries not containing a given XML node	settings
∅	search for missing entry ...	
•	search for present entry ...	

For yes/no fields you will get an checkbox with 3 options: ☑= yes, ☐= no, 🟩= undefined

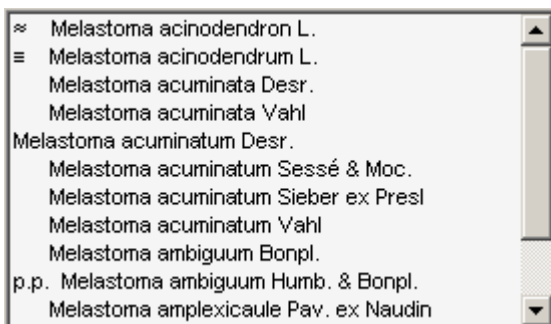If you want to indicate accepted names in the result list, choose **View**, **indicate accepted names**

Predefined queries

If prefined queries are available for a database, you can choose these querries under **Query, Predefined queries**. The command for the retrieval will be shown in the Query conditions area. Click on the 🔽 button to start the predefined query.

```
Query conditions
Missing hierarchy

SELECT    TaxonName.NameID,
TaxonName.TaxonNameCache,
0 as IsAcceptedName
FROM      TaxonName LEFT
OUTER JOIN
          TaxonHierarchy
ON TaxonName.NameID =
TaxonHierarchy.NameID
WHERE
(TaxonHierarchy.NameID IS
NULL)
```

Query results

To indicate the synonymy of the names in the query results check the menu **Query**, **Indicate synonymy**. The synonymy of the names will then be indicated as shown in the image below.

```
≈   Melastoma acinodendron L.
≡   Melastoma acinodendrum L.
    Melastoma acuminata Desr.
    Melastoma acuminata Vahl
Melastoma acuminatum Desr.
    Melastoma acuminatum Sessé & Moc.
    Melastoma acuminatum Sieber ex Presl
    Melastoma acuminatum Vahl
    Melastoma ambiguum Bonpl.
p.p. Melastoma ambiguum Humb. & Bonpl.
    Melastoma amplexicaule Pav. ex Naudin
```

An accepted name will not be indented. Synonyms and names with no synonymy defined will be indented. The syononyms will be marked with a sign to indicate the synonymy as described in the list below.

| Sign in list	meaning
not indented	Accepted name
indented, no sign	No synonymy defined
≡	homotypic synonym
=	heterotypic synonym
−	unknown synonymy
≈	duplicate, isonym or orthographic variant
p.p.	pro parte resp. more than one synonymy defined
•	final revision

Options

The maximal number of items shown in a query result can be set in the window for the query options. The default value is set to 100. If you have a fast connection to your database or need to see more or less results, you may change this value to any number you like.
To change the displayed search fields click on the button. This opens a form where you can select and deselect the fields shown for searching specimens. You might also change the maximum number of items that will be shown in the result list.

The limit for the creation of drop down lists is by default set to 1000. If you connect to a database, the programm will create drop down list based on the entries in the database to ease the entry in the query. For slow connections you may set this to lower value to speed up the start of the program. If you set the value to 0 no drop down lists will be created.
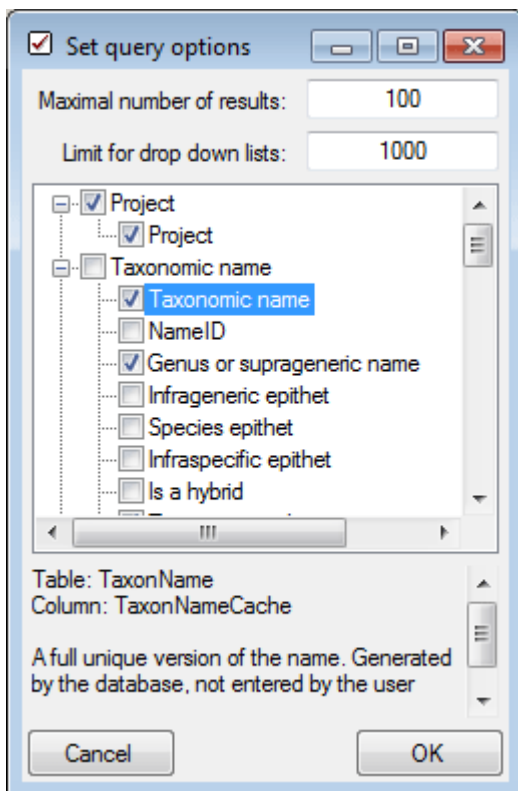
In the top of the form you can specify the maximal number of resultset that will be retrieved in one query. In the lower part you can select the search fields.

Wildcards in SQL

There are 4 different possibilities for wildcards in SQL:

%   any string consisting of no, one or many characters, e.g. Pinus **%** will find anything like Pinus, Pinus sylvestris, Pinus strobus etc.

_   a single character, e.g. Pinus s_lvestris will find Pinus s**y**lvestris and Pinus s**i**lvestris etc.

[]   any character out of a given range like **[abcde]** or **[a-e]**, e.g. Pinus s**[iy]**lvestris will find Pinus sylvestris and Pinus silvestris.

[^]   any character not in a given range like **[^abcde]** or **[^a-e]**, e.g. Pinus s**[^i]**lvestris will find Pinus sylvestris but not Pinus silvestris

Main form

The header of the main form allways shows you, with which database you are working, the version of the client, the database server and your login. The area for the data entry is devided into 2 main parts. In the upper part you enter nomenclatural information and facts, in the lower part taxonomic opinions of the editors (see topic database for further informations).
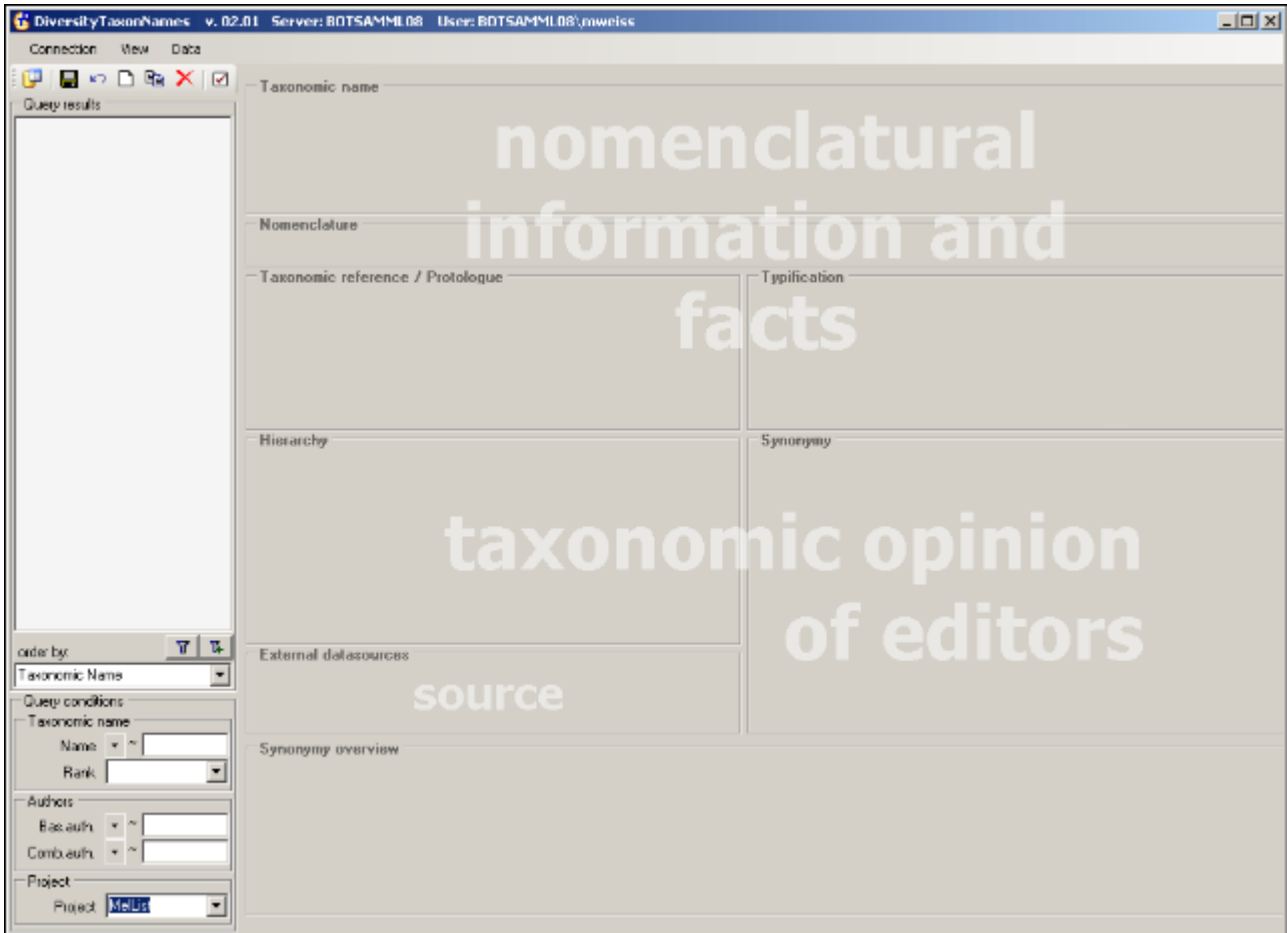

To see the descriptions for the fields of the database, simply move your mouse over it. A tooltip will appear with an explanation as shown in the example below.
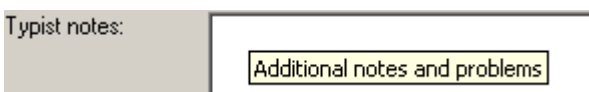
Where you have to choose names from lists, type the beginning of the name in the field and then click on the drop down button ▼. This will start a query in the database and list the result in the combobox.

If a field is to small for reading its content double-click on it. A form will open, where you can edit the text.

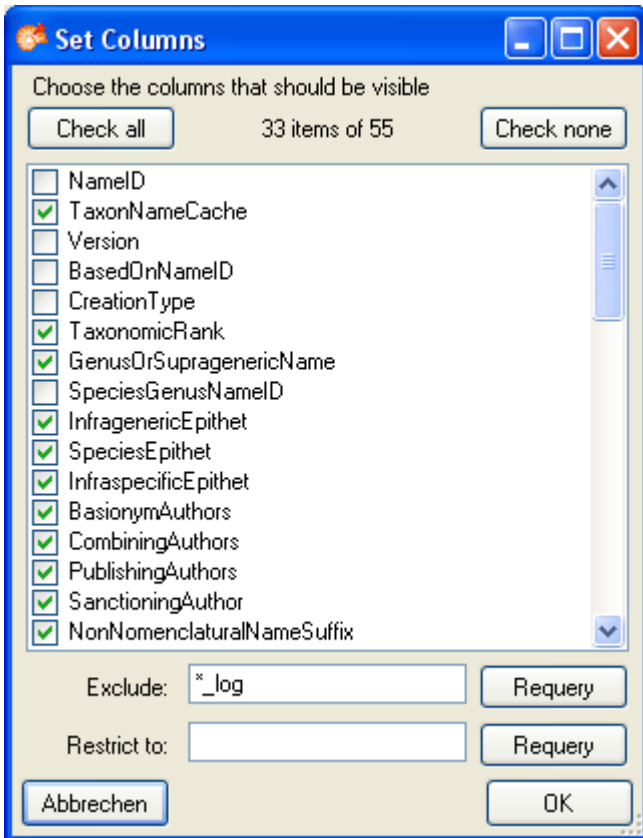**Edit DateOfPublSupplement**

May 1823

Cancel OK
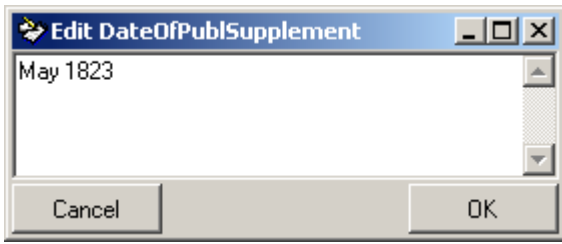
List view

To see the data in a spreadsheet, choose **View** → **List view ...** from the menu. A window will open where all names from the query will be listed.

Customize visibility of fields

To change the selection of the visible fields, click on the **Column visibility** button. A window as shown below will open, where you can change the visibility of the columns.

Customize sequence

To change the sequence of the columns, just use your mouse to drag the columns to the position of your choice.

Sorting of the data

To sort the data in the grid just click in the header of the column which you want to use as sorting column. The sorting sequence will be kept even if you change values in this column. That means that if you change a value in the sorting column the changed dataset will be placed at the new position according to its new value. The sorting of a column will be indicated with an arrow for the direction of the sorting (up or down).

Change to data in main form

To change to a name in the main form, click on the , use the ⚏button.

Formatting the grid

To adapt the width of the columns or the height of the rows either drag the border with the mouse, double click the border to get the optimal size for one column or respectively or click on the ⚎button for an optimal height of the rows or the ⚎button for an optimal width of the columns.

Edit the taxa in a list

To edit the taxa in a spreadsheet, choose **View →** ✏️**Edit taxa in list ...** from the menu. A window will open where all names from the query will be listed.

Customize visibility of fields

To change the selection of the visible fields, click on the 📊**Column visibility** button. A window as shown below will open, where you can change the visibility of the columns.

Customize sequence

To change the sequence of the columns, just use your mouse to drag the columns to the position of your choice.

Sorting of the data

To sort the data in the grid just click in the header of the column which you want to use as sorting column. The sorting sequence will be kept even if you change values in this column. That means that if you change a value in the sorting column the changed dataset will be placed at the new position according to its new value. The sorting of a column will be indicated with an arrow for the direction of the sorting (up or down).

Change to data in main form

To change to a name in the main form, click on the , use the ![button icon]button.
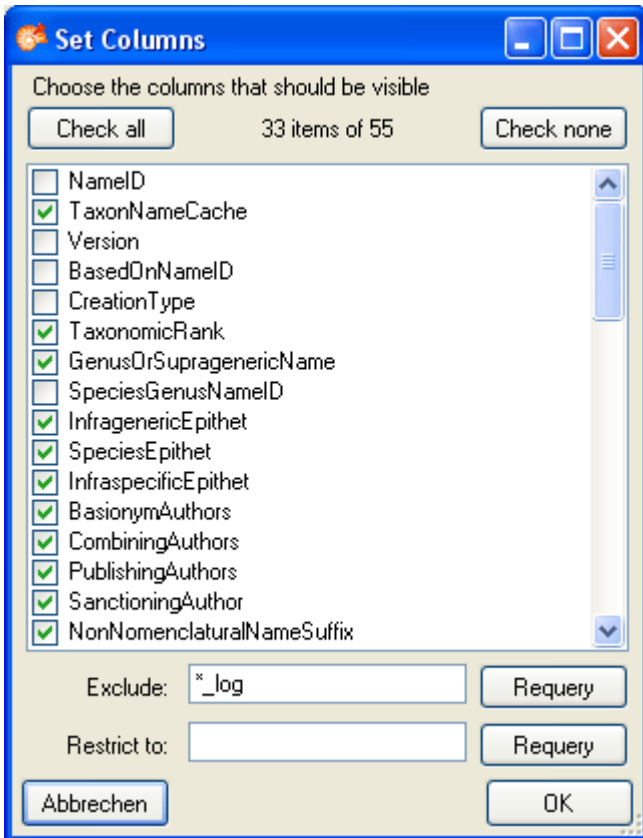
Formatting the grid

To adapt the width of the columns or the height of the rows either drag the border with the mouse, double click the border to get the optimal size for one column or respectively or click on the ![button icon]button for an optimal height of the rows or the ![button icon]button for an optimal width of the columns.

Data

To **save** the changes in a dataset click on the ![save] button.

To **undo** the changes in a dataset click the ![undo] button. This will recover the original data unless the changes had been saved or changes were done in the synonymy.

To create a **new** entry in the database, click on the ![new] button above the search result listbox. A window will open where you can add additional projects to the list in which the new name shoul be included (see below). By default only your current project will be selected.
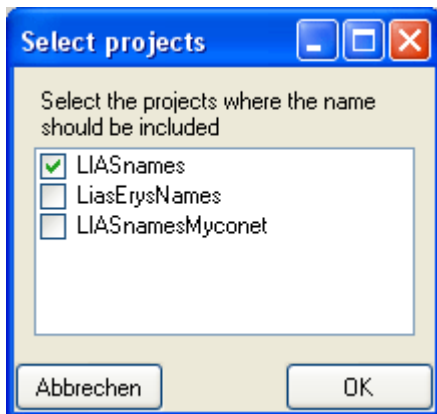
Leave only those projects the new name is related to. Click OK to create a entry with the name "New Taxon".

To **copy** a taxon, choose it from the list and click on the button ![copy]. As with the creation of a new dataset you can insert the copy of a name into other projects (see above).

To **delete** a dataset click on the button ![delete].

Taxonomic Name
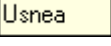
The header shows the name, the ID and the version of the name - these data can not be edited. If a dataset should be ignored, check the checkbox "Ignore". In the last field you can enter a revision level for the dataset. If you want to inspect the [history](#) of a dataset click on the button.

Data concering the taxonomic name including the [authors](#) are entered in the corresponding area (see image below).

Depending on the taxonomic rank, the form will show only the fields available for this rank.

If the genus is linked to an entry in the database, the field for the genus will change as shown here and the program will not allow you to edit the name. To edit the genus of the name, click the button to release the link to the genus.
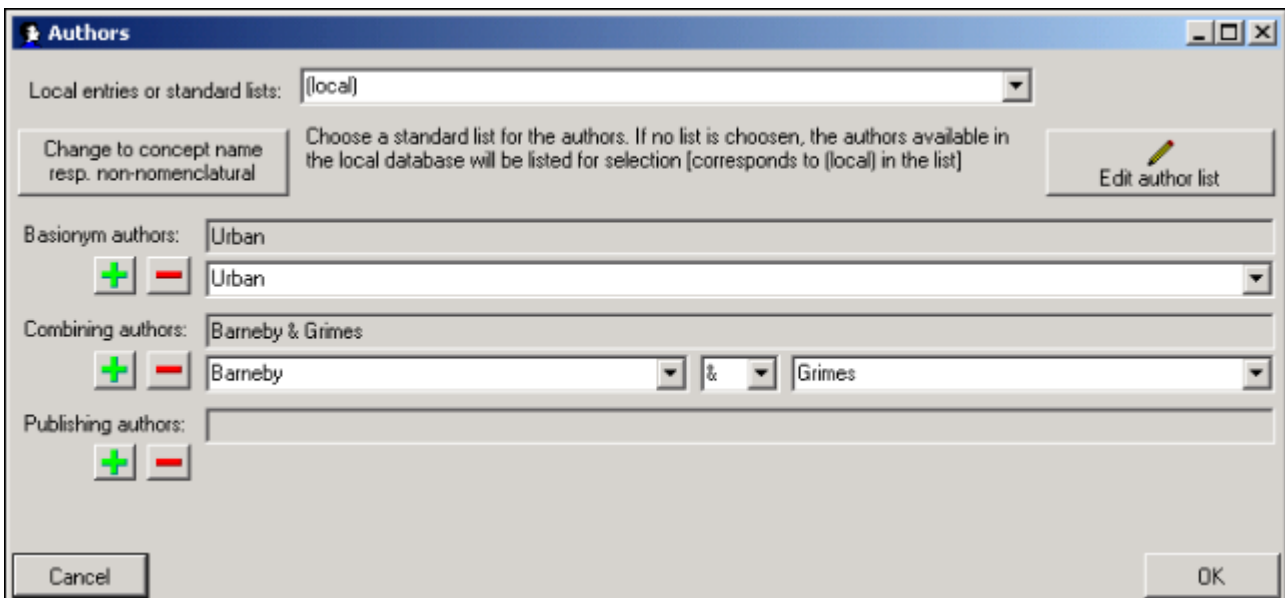
If the name is a [hybrid](#) , check the checkbox "Is hybrid" to get access to the fields for hybridization.

Authors

For names according to an official code (ICBN) the authors of the name should be given.
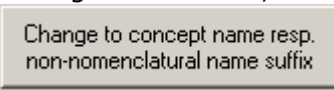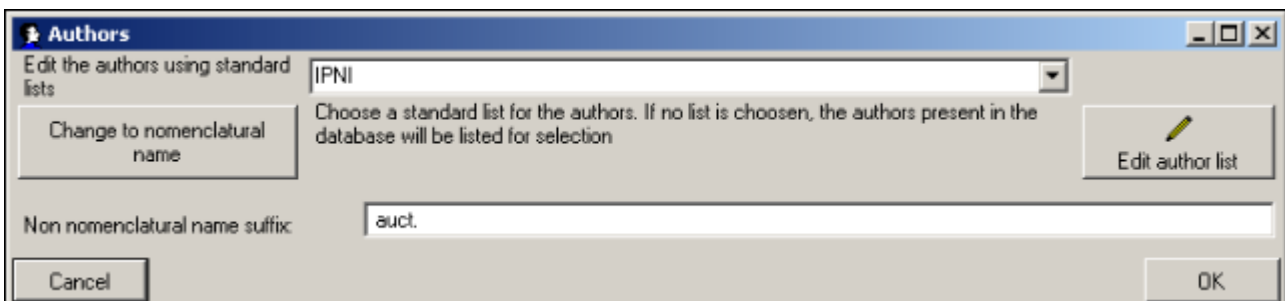
To edit the authors using standard abbreviation list click on the [icon] Button. This will open the form for editing the authors.
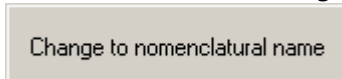
here you add (click on the [+] button) and remove (click on the [−] button) authors according to the standard list. You can choose the standards for the abbreviation from the provided lists.

For names not following an official code authors should not be given. Instead, include a descriptive phrase. To enter a floristic name click on the **Change to concept name resp. non-nomenclatural name suffix** button. The form will change as shown below.

Here you can enter a non nomenclatural name suffix for e.g. floristic names. To change back to a taxonomic name according to nomenclature click on the **Change to nomenclatural name** button. In the main form the authors part will then change to depict the field for the concept suffix for the name as shown below

To edit the selected list of the authors click on the button.

Nomenclature

DiversityTaxonNames was developed to follow, for the most part, the International Code of Botanical Names (ICBN). Other codes like the code for names in zoology have not yet been taken into account. Information corresponding the nomenclature are entered in the provided fields as shown below.

Taxonomic reference and protologue

Details about the protologue are entered in the area Taxonomic reference / Protologue

To enter a website for the protologue either enter the link in the field or click on the button to start a query. This will open simple browser where you can search the URL for the protologue as shown below.
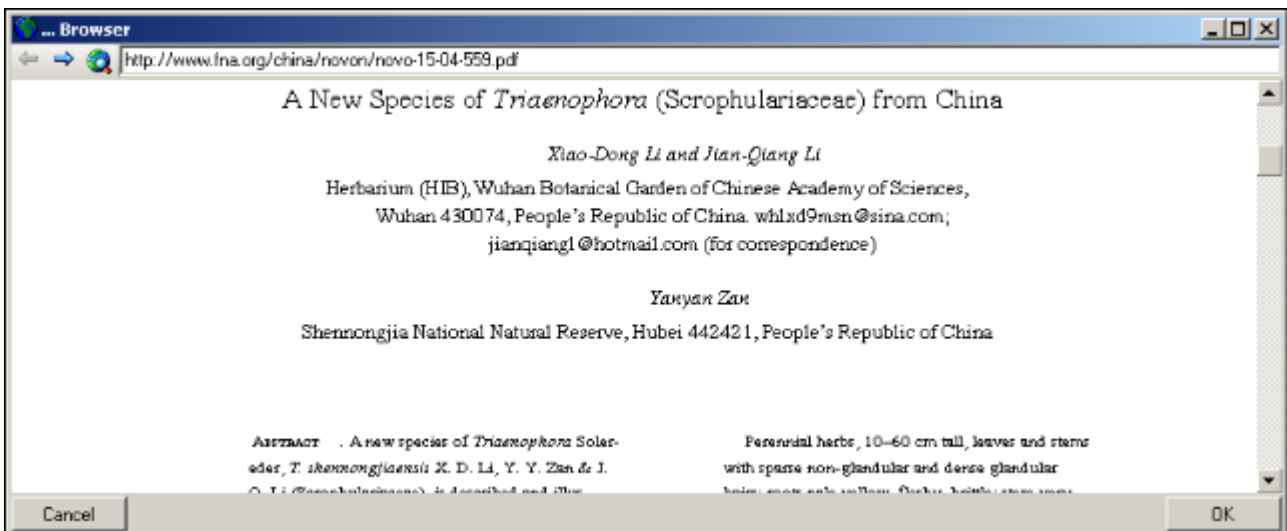
Typification

Details about the type can be entered in the area "Typification".

The navigate through the data use the ▮◀, ◀, ▶and ▶▮button. To add a new typification use the 🞢button, to delete the current dataset, use the ✕button.

Hybrids

If a taxon is a hybrid, check the checkbox Is hybrid ☑ to open the part for the entry for the parents of the hybrid. You can determine up to 4 parents for a hybrid as shown in the figure below

| Hybridization: | Sagina procumbens L. ▾ | Sagina saginoides (L.) Karsten ▾ | ▾ | ▾ |

Synonymy

Basionym

The name on which the choosen name is based on should be entered in the "Based on:" field.

A name can be either an accepted name or a synonym to an accepted name. If nothing was specified so far, two buttons are visible:

choose one of these to make the choosen name to either an accepted name or a synonym.

Synonym

If the name is a synonym, the area for the synonymy will appear as shown in the images

A name in most cases is synonym to one name but may sometimes be synonym to several names (e.g. pro parte). If you want to keep information of the synonymy you can set it to ignore.

Delete a synonym

To delete a synonym, choose it in the list and click on the delete button .

Accepted Name

If the name is accepted, the form will show data related to accepted names

You can indicate accepted names in the search result list (in the menu choose **View**, **indicate accepted names**)

Conflict (Name declared a synonym and a basionym)

If by any chance (e.g. after import of synonymy relations from a foreign source) a name happens to be a synonym and an accepted name at the same time this is regarded as an error and will be shown as in the image below.
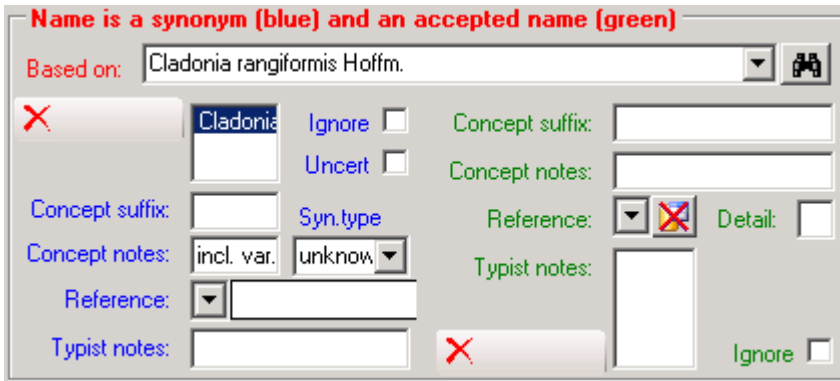
The data for the synonym are shown on the right side with blue labels, while the data for the accepted name are shown on the left with green labels. To correct this error you can either remove or ignore the assignement as a synonym or an accepted name.

Overview

In the area **Synonymy overview** you see the accepted name for the name you chose and a list of all the synonyms to this name. The sign ≡ in front of a name following the accepted name means that this and the following names are homotypic synonyms to the accepted name. Groups of hetereotypic names start with a "=" in front of the first entry. If present, the first entry in each group is the basionym of this group. Names with unclear relation to the accepted name are preceded by a "-". To change to one of the names in the list, choose it in the list and click on the search button 🔍. The overview can be shown either as a list, corresponding to lists in publications or as a tree view. Use the ⊟ and ⊟ button to change between these views. In the tree view the current name is marked with a yellow background as shown below. The scrutiny of the search can be adapted by changing the number in the tool bar ⌷3⌷. As standard it is set on 3, but can be changed to values between 0 (just the current name) to 9. The higher the scrutiny the more thoroughly the search for synonyms will be performed and the longer it will take.

Synonymy overview

Ceratiomyxa fruticulosa (O.F.Müll.) T.Macbr. , N.Amer.Slime-Moulds : 18 (1899) [Martin & Alexopoulos, 1969: 33]
≡ Byssus fruticulosa O.F.Müll., Fl.Dan. 4 (12): 8 (1777) [Martin & Alexopoulos, 1969: 33]
  Famintzinia fruticulosa (O.F.Müll.) Lado, Cuad.Trab.Fl.Micol.Iber. 16: 43 (2001)
= Tremella hydnoidea Jacq., Misc.Austriac. 1: 145 (1778) [Martin & Alexopoulos, 1969: 33]
  Ceratium hydnoideum (Jacq.) Alb.& Schwein., Consp.Fung.Lusat. : 358 (1805) [Martin & Alexopoulos, 1969: 33]
  Ceratiomyxa hydnoidea (Jacq.) Kuntze, Revis.Gen.Pl. 3 (3): 507 (1898) [Martin & Alexopoulos, 1969: 34]
  Ceratiomyxa mucida var. hydnoidea (Jacq.) Torrend, Brotéria, Sér.Bot. 6 (2): 64 (1907)
= Clavaria puccinia Batsch, Elench.Fung. : 139 (1783) [Martin & Alexopoulos, 1969: 33]

Explanation of the example above:

accepted name		Ceratiomyxa fruticulosa (O.F.Müll) T.Macbr.
basionym	≡	Byssus fruticulosa O.F.Müll.
homotypic synonym		Famintzinia fruticulosa (O.F.Müll.) Lado
heterotypic synonym and basionym	=	Tremella hydnoidea Jacq.
heterotypic synonym		Ceratium hydnoideum (Jacq.) Alb.& Schwein.

Editor

For a direct editing of synonymy list click on the button [pencil icon]. The list with the selected names and the synonymy overview will get a yellow background (see image).
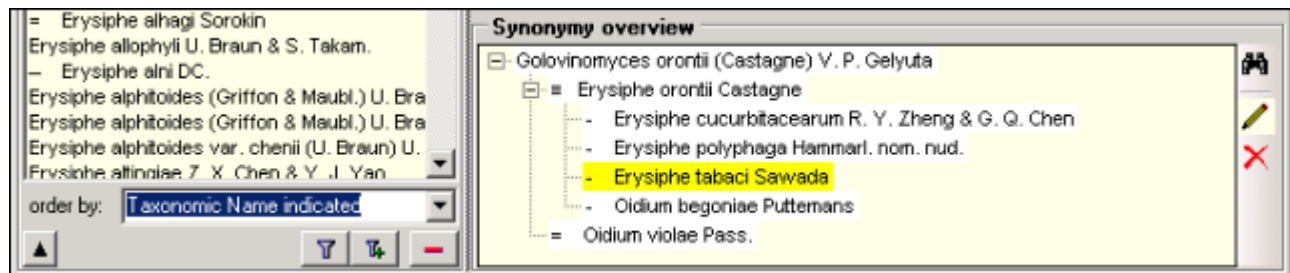
Now you can add synonyms to the synonymy list by simply dragging them from the selected names list on a name in the synonymy tree view. If you drag the name on the accepted name, it will become a homotypic synonym of the accepted name. If you drag it on a heterotypic synonym, it will become a heterotypic synonym of the accepted name and get the same basionym as the name you dragged it upon. If this name has no basionym, the dragged name will become a homotypic name of the name you dragged it upon. If you want to create a new heterotypic synonym or a synonym with unknown relation to the accepted name, drag the name in the empty space of the tree view.

If you want to correct the synonymy you can drag the names within the synonymy overview either on another name to create a homotypic synonym or into the free space underneath the names to create a heterotypic synonym to the accepted name.

To remove a name from the lists of synonyms, choose it in the tree view and click on the [X icon] button.

Isonyms, duplicates and orthographic variants

Some names are a variant spelled exactly as another name but with deviating citation of the author(s). These might be a name with wrong citation of the publishing or recombining author(s) or an isonym sensu ICBN. To change the type of the synonymy to an isonym choose the taxon and change the syn. type e.g. to "isonym". In the synonymy overview list the isonyms, duplicates etc. are indicated by square brackets "[duplicate: ...]".

Ignored datasets

If it should happen, that a dataset that is ignored is included in a synonymy in the form that

another name is referring to this ignored name, the system will not be able to generate the synonymy properly. To detect ignored datasets in the synonymy choose "Query -> Show ignored data" from the menu. Ignored dataset will than be shown with red letters in the synonymy as shown below.
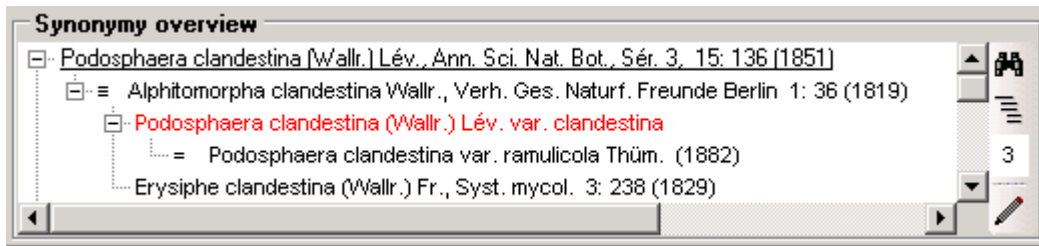
The information about the synonymy is related to opinions of the editors and are allways restricted to a certain project. The data of the synonymy are stored in table TaxonSynonymy. The data of the accepted names are stored in table TaxonAcceptedName.

Hierarchy

The taxonomy for the choosen name is depicted in the area Hierarchy. If the taxonomy has not been specified a button (set hierarchy) will be visible
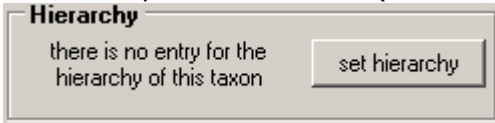
click on it to insert a dataset for the hierarchy of this taxon.

To choose a higher taxon you first have to specify the taxonomic rank of the choosen name

and type at least 2 characters of the higher taxon in the 'Superior taxon:' field.

Click on the drop down button to get the list of taxa above the rank of the name for which you want to set the hierarchy. The hierarchy will then be shown in the tree view.
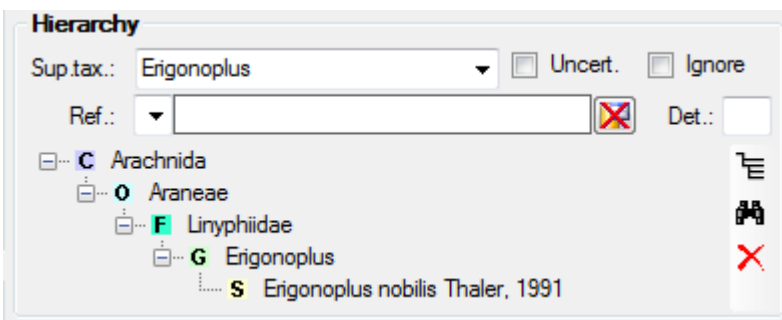
The first line shows the taxon or taxonomic group superior to the choosen taxon. To directly access the database entry of a higher taxon, choose this taxon in the hierarchy tree and click on . The button will delete this entry. To get an overview of the whole hierarchy click on the button . You get an overview for the whole taxonomy including the current taxon as shown below.
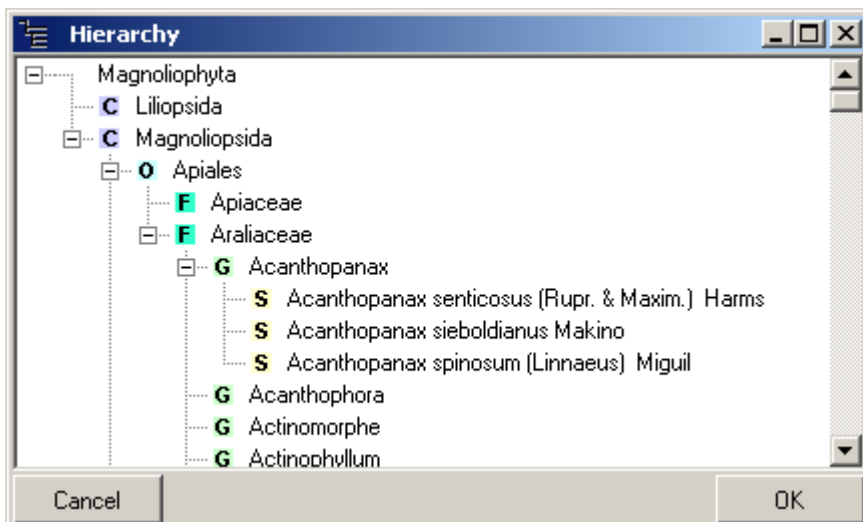
In the taxon tree the main taxonomic groups are indicated with icons (S = species, G = Genus, F = Family, O = Order, C = Class, K = Kingdom). To change the position of a taxon within the hierarchy, use the drag & drop function of the tree.

To enable a search within the hierarchy, please use the maintenace function to update the hierachy cache. The hierarchy is stored in one string, so to search for e.g. all members of e.g. a certain familiy e.g. Rosaceae, please use the ~ operator and a leading wildcard in your search string e.g. "% Rosaceae" or "* Rosaceae".

The information about the hierarchy is related to opinions of the editors and are allways restricted to a certain project. The data of the hierarchy are stored in table TaxonHierarchy.

Common names

A name can have several common names. These are entered in the section **Common names**. To add or delete a common name use the ▯ and ✗ buttons.

The data of the common names are stored in table [TaxonCommonName](#).

Geography

Details about the type can be entered in the area "Geography".
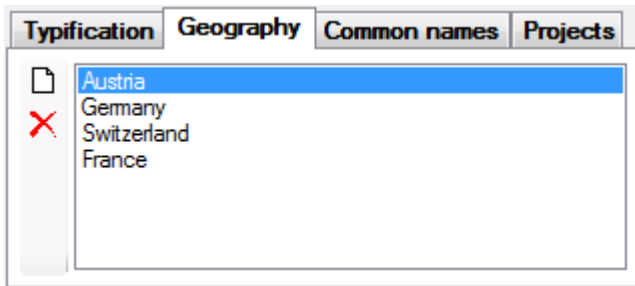
The geographical distribution of the type specimen according to the protologue can be typed in the field Locality of the area Typification. In the area Geography names according to a standard can be entered as a list as shown above. To remove an entry from the list click on the ✗button. To add an entry click on the ⬜button. A form will open where you can choose geographical places from a list.
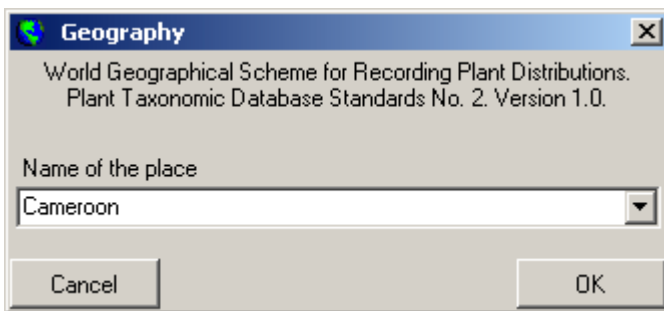
The data of the geography are stored in table TaxonGeography.

Resources

Resources related to a taxon can be added in the area "Resources". Use the ☐(add), ✕ (remove) and 💾(save) buttons to handle the resources.

For every project, a description template can be defined for the resources. Choose **Adminstration** - 📑**Image description template ...** from the menu to define the template. To add a description for a resource, use the button 📑. A window will open where you can edit the description for a resource.

The data of the geography are stored in table TaxonNameResource.

Lists for taxonomic names

With DiversityTaxonNames you can document list for taxonomic names like e.g. checklists. The taxon lists may refers to a project defined in DiversityProjects (see **Administration - User ...** in the menu). To edit the taxon lists choose **Administration → Taxon lists ...** from the menu. A window as shown below will open, where you can enter, edit or delete taxon lists. The administrative functions are only available for administrators.
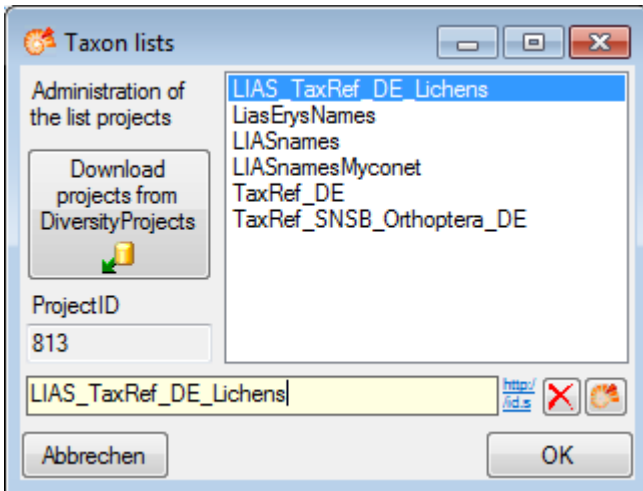
For the detailed documentation of the **Taxon lists** please use DiversityProjects. To load projects into the list, click on the **Download projects from DiversityProjects** button (see above). A window as shown below will open. If several Databases for DiversityProjects are available on your server you have to choose one of these. The projects available in the DiversityProjects database will be listed as shown below where those already present are checked and green, while missing projects are red. Check those projects you need as lists and click the **Start download** button.
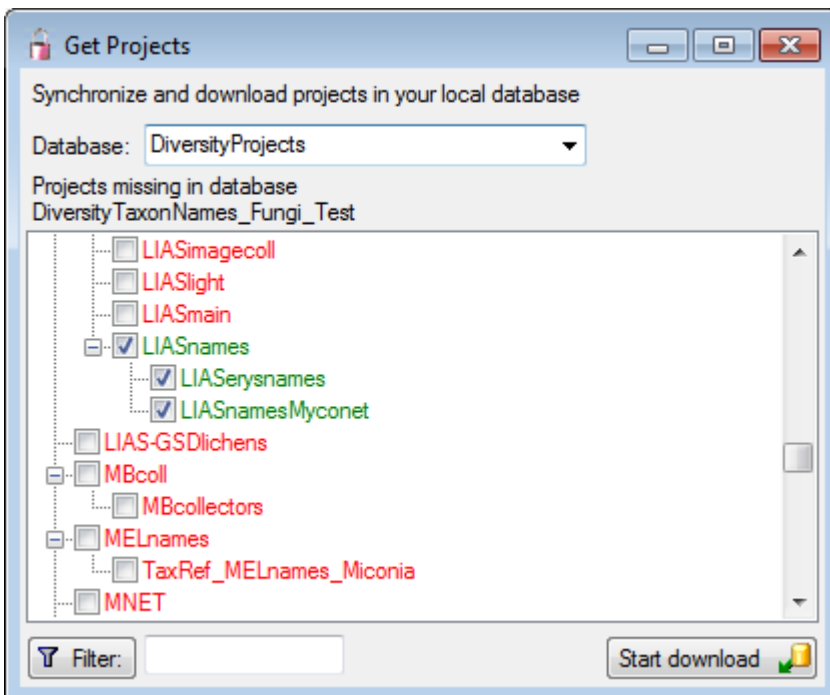
For access to the related data choose the tab **Taxon lists** (see below).
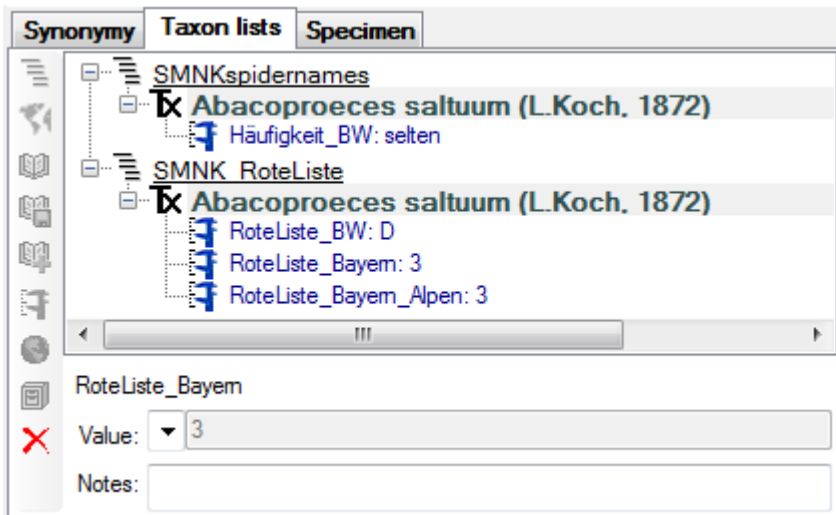
To enter a new list, click on the ⬓button. This will open a window where you can choose from the available taxon lists. Choose the list in which you want to include your name and click OK.

Area

The geographical area can be definded for a single name (via 🌐**Distribution**) or for the whole taxon list. The entries refer to the World geographical scheme for recoding plant distributions.

To enter the geographic area for the whole list, click on the 🗺button. A window will open (see below), where you can select a geographic region as defined in the world geographical scheme for plant distributions.
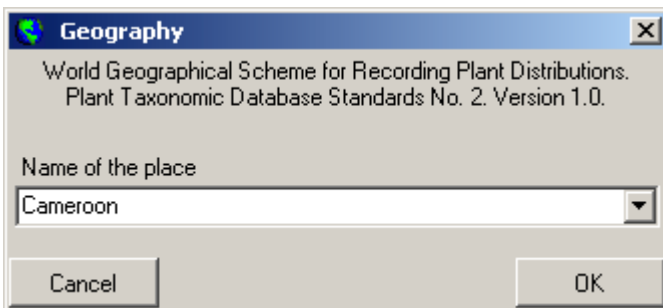
If you need a certain region not provided in the list, just type the name of the region and click OK. The new region will be listed directly underneath the list (see below).

References

For every name in a taxon list, you can enter the references where the informations for the list are derived from. To add a reference, select the list for which you want to add a reference and click on the 📖button. The entry for the reference will appear underneath the taxon as shown below. To edit this entry, select it in the hierarchy and use the contols underneath the tree. You may link the reference to an external source like DiversityReferences or one of the provided webservices.
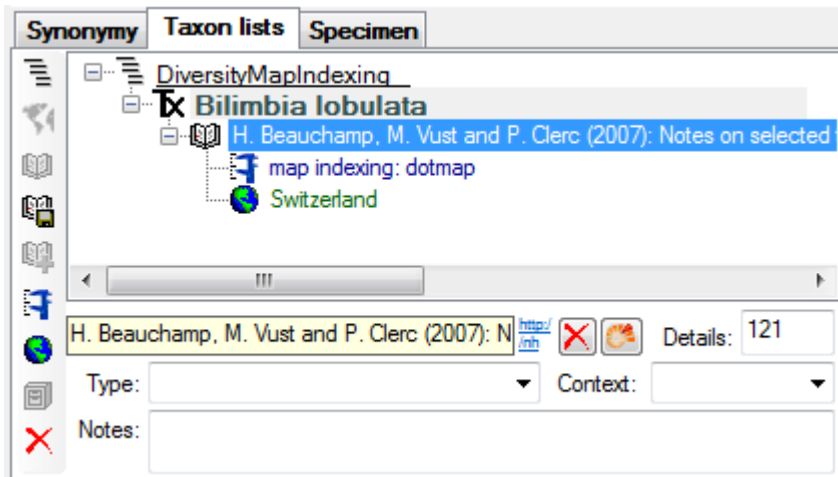
If you want to transfer the informations of a reference together with depending ⊞analysis and 🌐distribution entries to another name, select the reference in the tree und click on the 📖button. As a result you can use the 📖button to insert these informations at a different place.

Analysis

To edit the analysis used within the database, choose **Administration - Analysis categories ... from the menu**. A window will open, where you can define new types of analysis or edit existing ones. The analysis tyes may be organized within hierarchies as shown below. For categorized analysis you can define value lists, from which the user can choose.
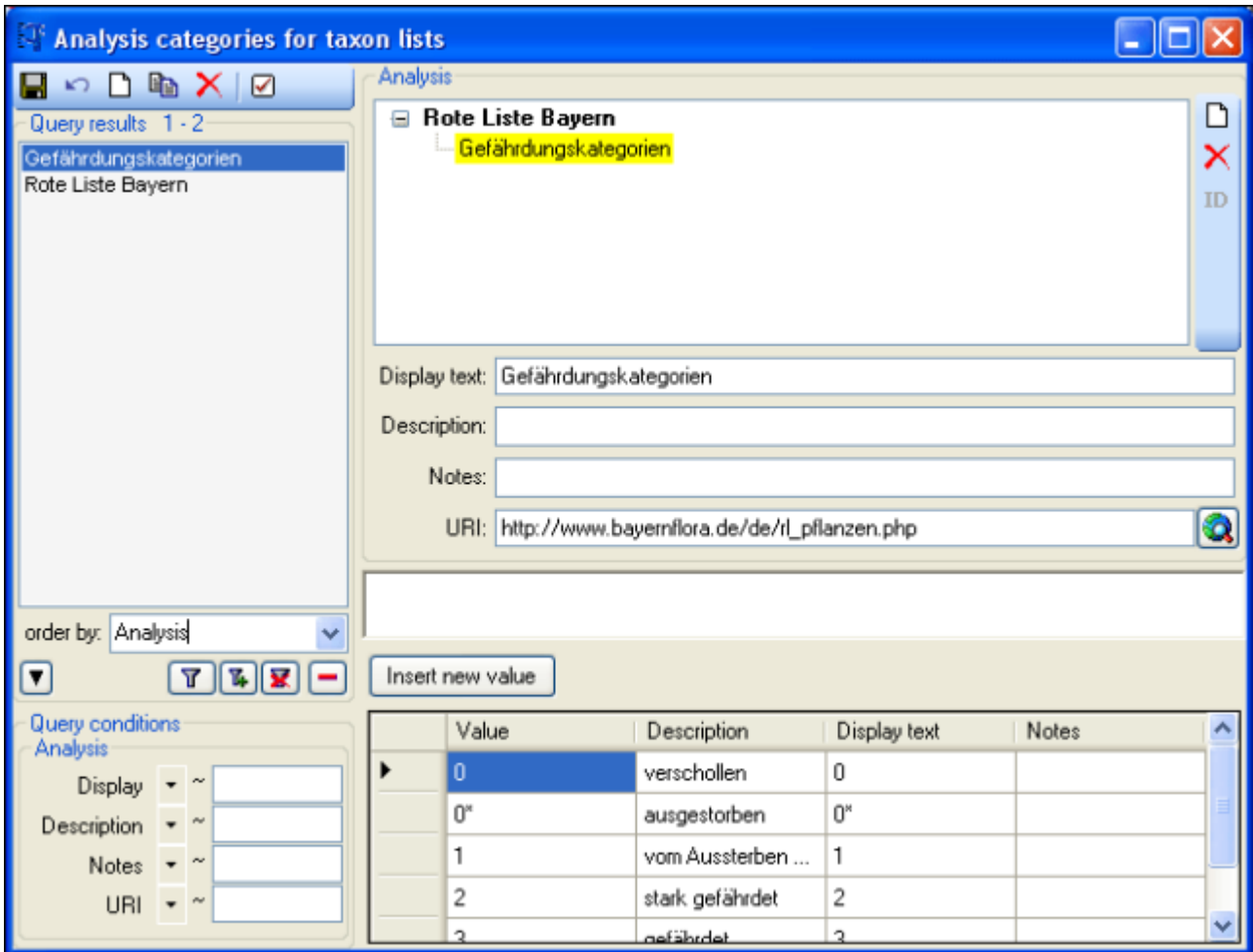
For a name within a list, you can enter an analysis can either directly for the name or a reference linked to this name. Select the name or the reference in the tree and click on the button to enter a new analysis. Depending on the definition of the analysis To edit the analysis used within the database, select it in the tree and use the area below the the tree. Depending on the definition of the analysis (see above) the values are either typed or selected from a list.
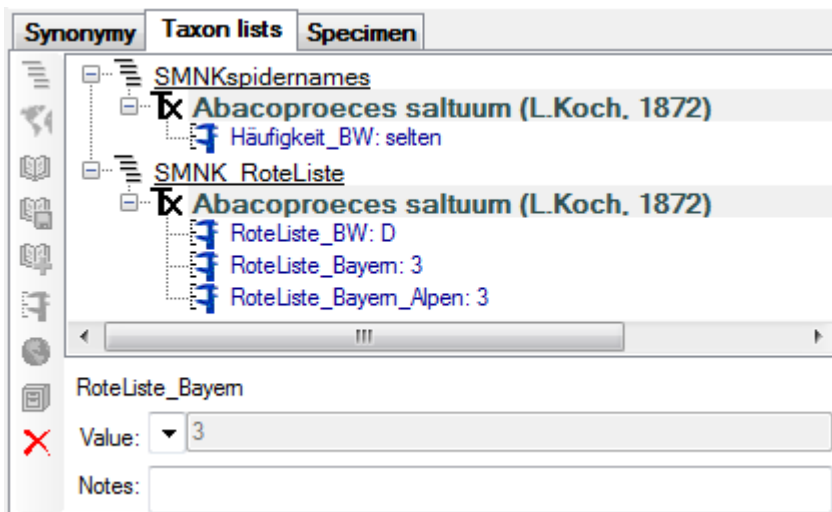
Distribution

For every entry in a taxon list, you can document the ⊕distribution of a taxon (see below). This information may be linked to the taxon or a reference (see below).
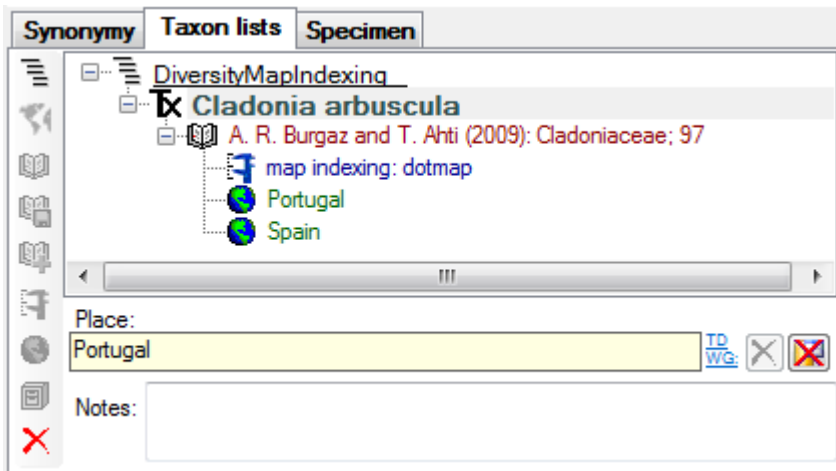
Choose either the list taxon or the reference in the tree and click on the button to insert a new ⊕distribution. The entries refer to the World geographical scheme for recoding plant distributions. A window will open (see below), where you can select a geographic region as defined in the world geographical scheme for plant distributions.
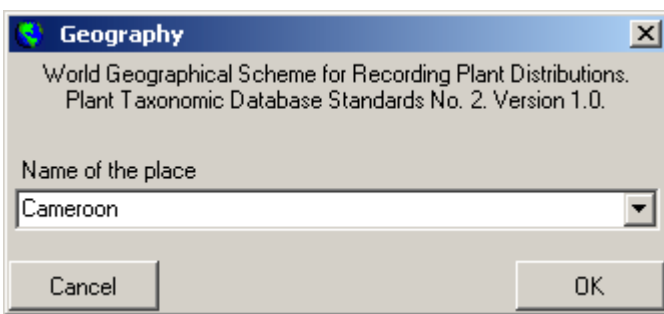
If you need a certain region not provided in the list, just type the name of the region and click OK.

Specimen

For every entry in a taxon list, you can document the collection specimen linked to this taxonomic name (see below). The entries may be linked to the module DiversityCollection. To enter a distribution for the specimen, use the ⊕button. A window will open (see above), where you can select a geographic region as defined in the World geographical scheme for recoding plant distributions.
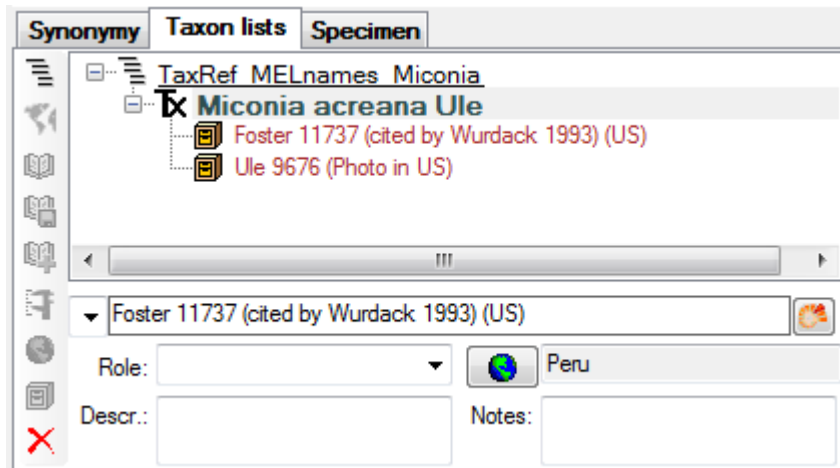
Export/font>

To export the taxon lists resp. checklist, choose **Data -> Export taxon list ...** from the menu. A window as shown below will open. By default the data will be exported as XML (see below). Choose a schema file of your choice to convert the XML-output in a HTML document. You have several options for the export in regard of the names as shown in the window below. The export may be based on the names in the list and can be restricted to the accepted names among these (see below - left). In the second option (see below - right), the accepted names for all names in the list will be selected and the taxon list will contain only these names and optionally their synonyms. Regarding the synonyms, if you want to suppress the export of isonyms and duplicates, use the according checkbox. For long lists, the second method will be rather time consuming, as the synonymy must be checked for every single name. The synonymy for the exort may be derived from any available project. The path of the exported file (either *.xml or *.html) is shown in the field Export file at the bottom of the form. The names in the list may be sorted according to the names or the synonymy where synonyms will be inserted after their assigned accepted names. The format the XML-export with a schema file of your choice, choose the schema file and click on the ⟳button.
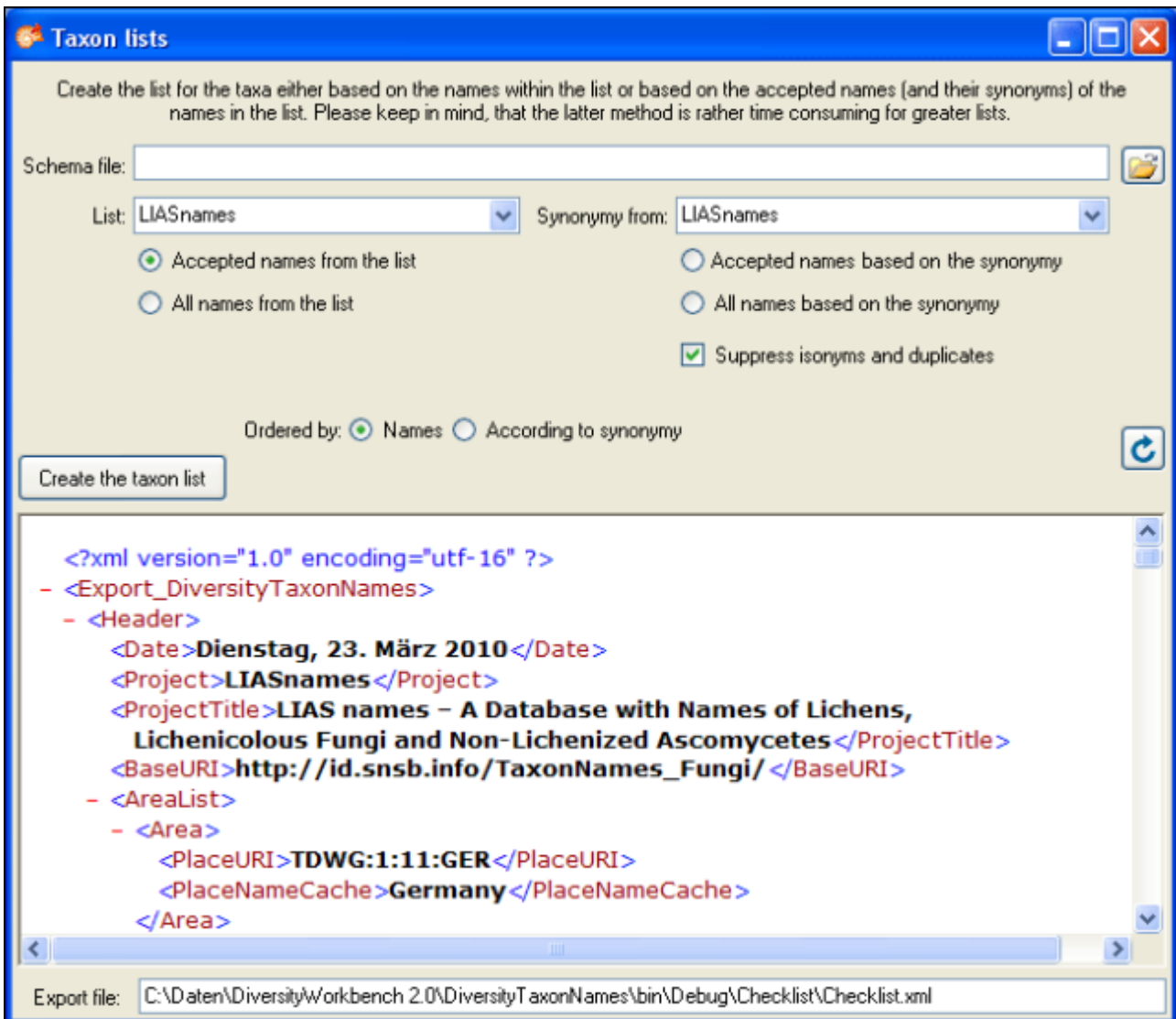
The tables involved in the storage for the taxon list are documented in the taxon list tables part.

Restriction of the access of users to taxonomic lists

If the access of a user should be restricted to names within a taxonomic list, choose **Administration - List user ...** in the menu. A windows as shown below will open, where you can add and remove lists thats should be accessible by a user. There is no restriction for the user if no lists are selected.
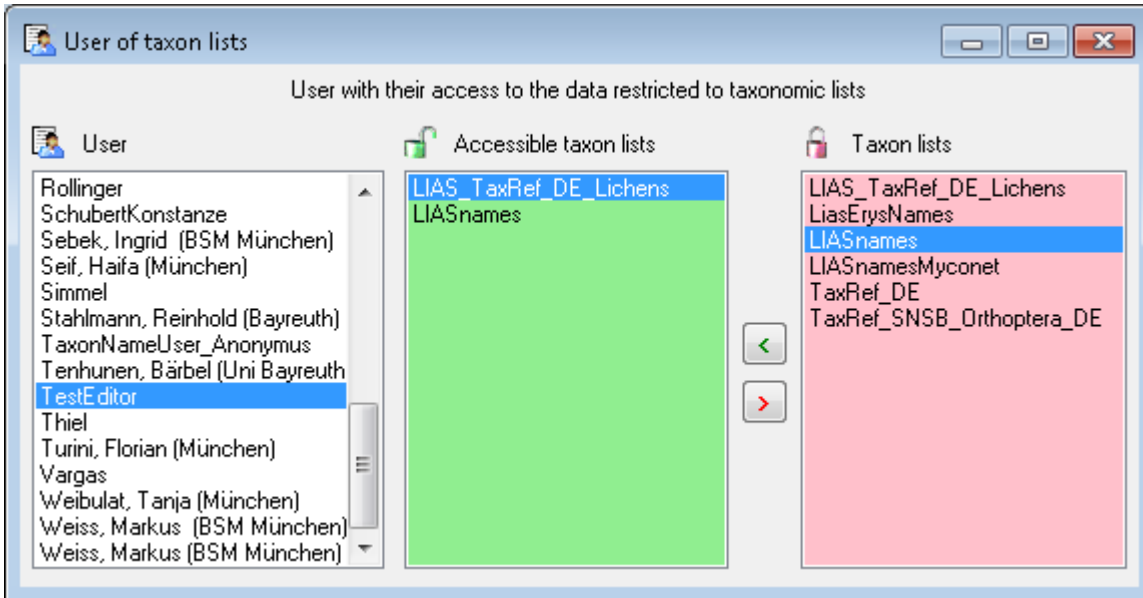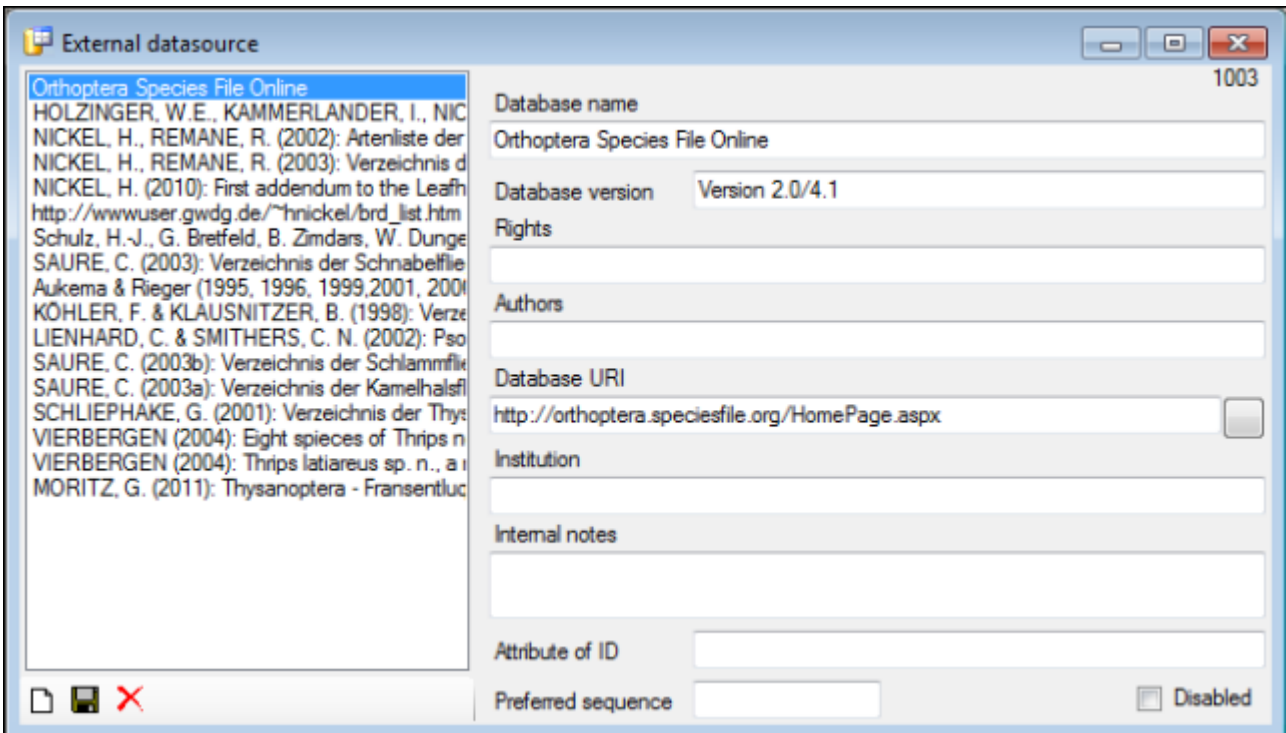
The tables involved in the storage for the taxon list are documented in the taxon list tables part.

External databases

Most names in DiversityTaxonNames were imported from external sources. These sources are documented in the area **External data sources**.

To enter or delete an external datasource, use the ⬜and ❌button button respectively. With the 🖹button you will get a detailed information about the external database. To edit the datasources themselve, choose **Administration - Data sources ...** from the menu. A window as shown below will open where you can edit the informations about the sources.

The data of the external sources are stored in tables TaxonNameExternalID and TaxonNameExternalDatabase

Database

The database for DiversityTaxonNames is based on Microsoft SQL-Server 2005. The names of the taxa were separated for practical reasons in several databases, each comprising a certain range of taxa. There is one master version, DiversityTaxonNames, containing the names of all databases and several databases containing the complete datasets including synonymy, hierarchy etc. for the following groups:

Database	Taxa	Source
DiversityTaxonNames_Fungi	higher fungi including lichens	LIAS
DiversityTaxonNames_Myxomycetes	slime moulds	NomenMyx
DiversityTaxonNames_Plants	higher plants	Species 2000, Tropicos, IPNI, Precis, ...
DiversityTaxonNames_Vertebrates	vertebrates	SAPM, ...

To access any database, you must specify the server where the database is located. For the configuration of this connection click on the ![button] button or choose **Connection**, **Database...** from the menu. If you got a new version of the client you can transfer the settings for the previous version: Choose **Connetion -> Transfer previous settings** from the menu. (see Database access for details)

Organisation of the data

The data are organised in several groups, data connected with the name, data based on the opinion of an editor, the source of the data and the projects.

TaxonName and **TaxonGeography** are containing the nomenclatural information. Most items are expected to be undisputed nomenclatural facts. Some data items include editorial opinion (nomenclatural status, but also the accepted spelling which may involve orthographical or grammatical corrections as required by ICBN). However, it is expected that within a project agreement can be achieved on the information contained in TaxonName, i.e. collaborators can "correct" information without consultation.

The tables **TaxonHierarchy**, **TaxonAcceptedName** and **TaxonSynonymy** - represent a higher level of taxonomic opinion, where different editors and different digitized data sources need to be kept separate. This is achieved by making a "Project" attribute part of the primary key. Editors that wish to collaborate and form agreement may work in a single project, editors that desire to define separate opinions work in separate projects. Furthermore, if the opinion is based on published information, the reference and page number may be recorded (in ...RefText, ...RefID, ...RefDetail). The duality of representing editorial opinion and remaining true to a digitized sources causes some problems. For example, a species accepted in a referenced source may not longer be accepted by the project editors. To avoid a further complication of the model, the "IgnoreButKeepForReference" attribute (default "false") may be set to "true", indicating that the statement is true from the perspective of the referenced source, but should be ignored when representing editorial opinion.

- **TaxonHierarchy** contains information about the hierarchical position of the taxa. For each project, a name may be referred to a single parent taxon. The attribute "NameParentID" may, however, be Null, in which case "HierarchyPositionIsUncertain" should be true.
- **TaxonAcceptedName** contains the accepted names ("valid or preferred taxon names"). For each project, a name in "TaxonName" may either be accepted (a corresponding record in "TaxonAcceptedName" exists) or not. Examples for not accepted names are synonyms of accepted names and nomenclaturally rejected names.
- **TaxonSynonymy** contains names, which are synonym to other names. Here a single name may be a synonym of multiple accepted names ("pro parte synonym"). The "pro parte" or corresponding sensu information is captured in the "ConceptSuffix" attribute, which is part of the primary key.

TaxonNameExternalID and **TaxonNameExternalDatabase** document the sources of the names.

TaxonNameProject document the projects of the names.

Overview of the main tables of the database
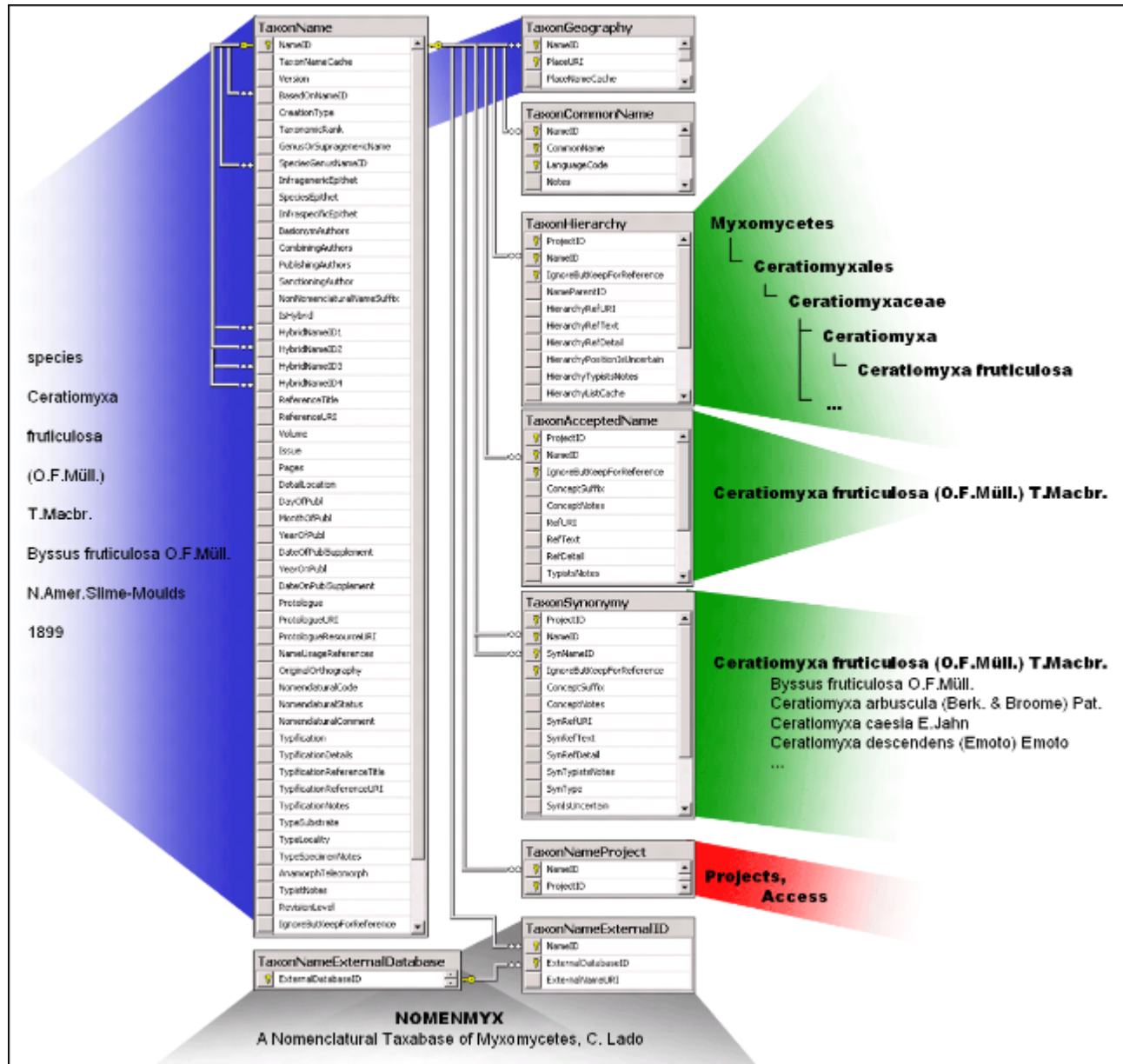
For details about the tables see Data tables.

Installation of the database

DiversityTaxonNames uses Microsoft SQL-Server 2008 as database engine. If you do not have a database server with DiversityTaxonNames already available, you have to install the database engine first. Download the free version of Microsoft SQL Server Express 2008 (e.g. SQLEXPRADV_x86_DEU.exe 🖳) from http://www.microsoft.com/downloads/.. Start the program and follow the instructions for the installation.

Download the database files DiversityTaxonNames_Data.MDF 🔲 and DiversityTaxonNames_log.LDF 🔲 from http://www.diversityworkbench.net/Portal/ provided as a zip archive (DiversitTaxonNames.zip) and copy them into your database directory.

Server configuration

To configure your server for remote access, launch the **SQL Server Configuration Manager** (see image below).
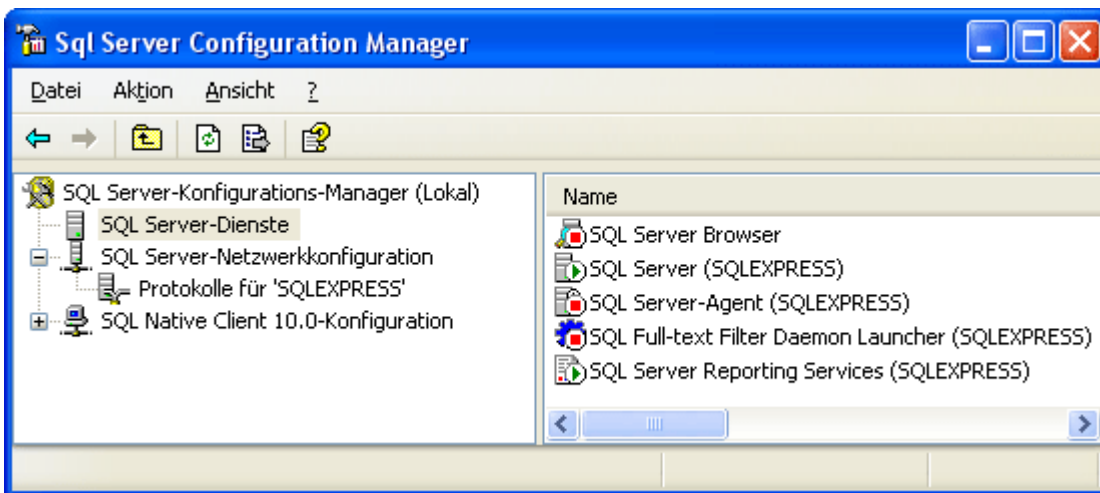
Then click on the "**Protocols** for SQLEXPRESS" node. Right click on "**TCP/IP**" in the list of Protocols and choose "enable" for TCP/IP.
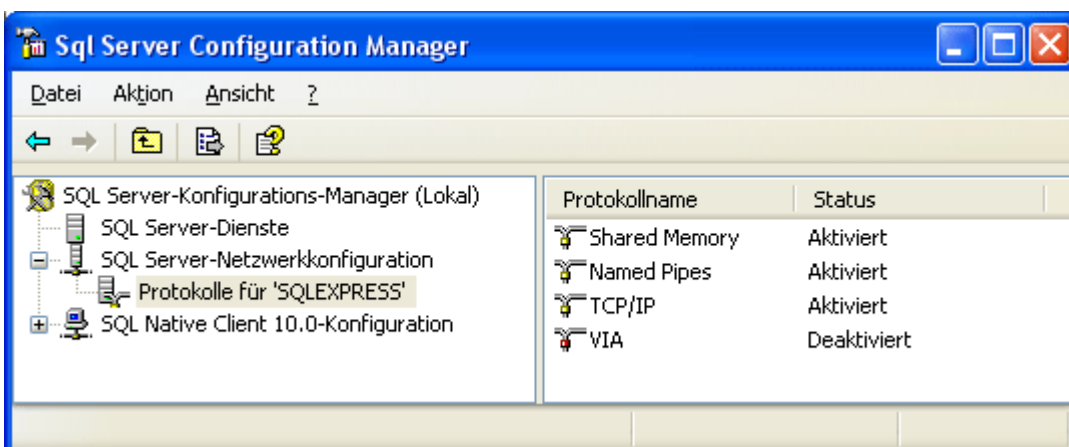
Right click on the TCP/IP node and select, "**Properties**" to open a window as shown below.
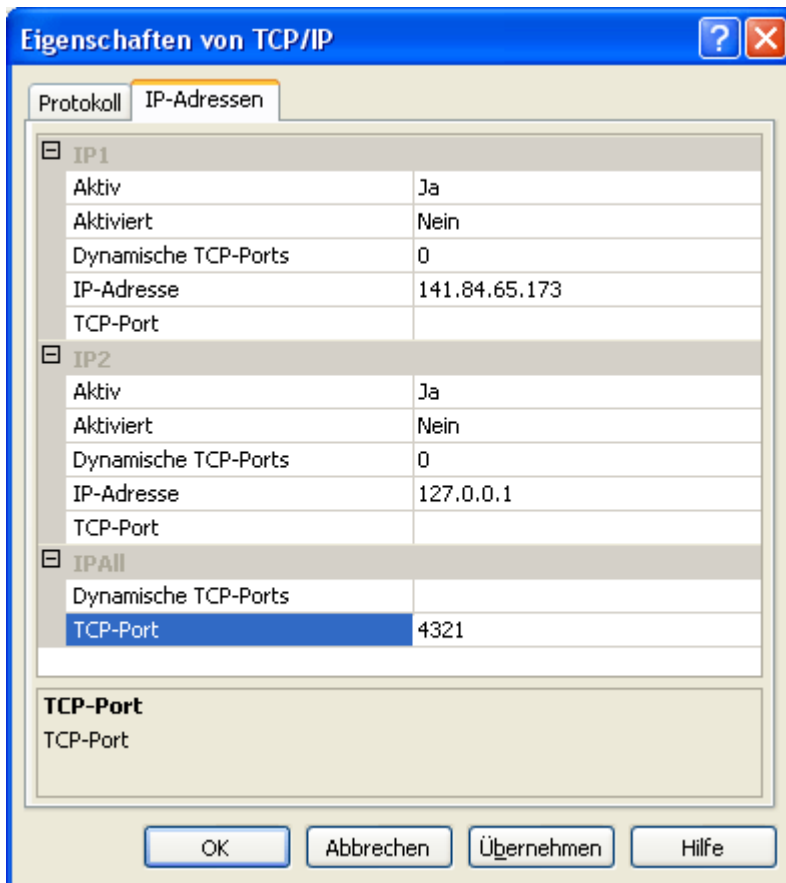
In the part **IPALL** clear out the value for "TCP Dynamic Ports". Give a **TCP-Port** number to use when making remote connections, e.g. "4321" as shown above. You have to restart the SQL Server Express service before you can connect to your database.

If you use a database on a server, make sure that the firewall of the server allows access via the port you set for the connections (see below).
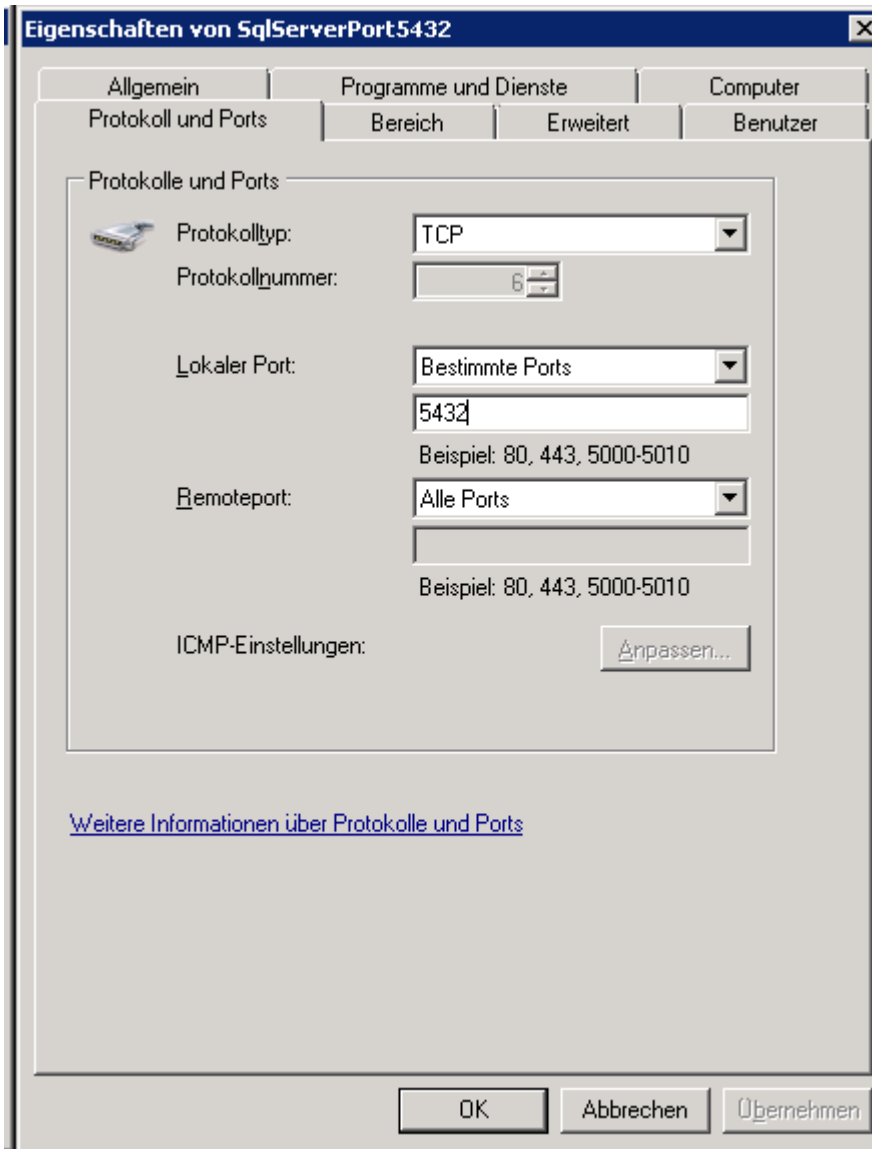
Start the Microsoft SQL Server Managment Studio and attach the database as shown below. Choose the node "databases" and right-click on it to open the context menu (see below). Then choose "attach" from the context menu. A window will open where you can choose the file DiversityTaxonNames_Data.MDF from your database directory and attach it to the database engine.
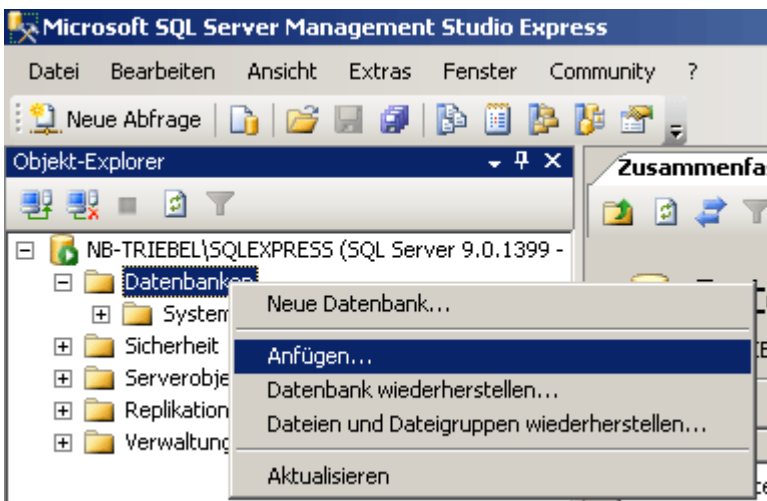
After the installation make shure to get the latest updates from
http://windowsupdate.microsoft.com/.

Database configuration

To configure your Database, use the Client as described in [Database configuration.](#)

Configuratation of the database

To configure your Database, choose **Administration -> Rename database** to change the name of the database according to your requirements. During this renaming all processes in the database will be terminated (you will get a warning if processes from other host are active).

Afterwards you should adapt the address that is published by the database for access by other modules. Choose **Administration -> Set published address** from the menu. This will change the published address to the name of the server where your database is located and an identifier for you database, e.g. http://xy.diversityworkbench.de/TaxonNames/ .

Database tables

Structure of the database including all tables

TaxonName
- NameID
- TaxonNameCache
- Version
- BasedOnNameID
- CreationType
- TaxonomicRank
- GenusOrSupragenericName
- SpeciesGenusNameID
- InfragenericEpithet
- SpeciesEpithet
- InfraspecificEpithet
- BasionymAuthors
- CombiningAuthors
- PublishingAuthors
- SanctioningAuthor
- NonNomenclaturalNameSuffix
- IsRecombination
- IsHybrid
- HybridNameID1
- HybridNameID2
- HybridNameID3
- HybridNameID4
- ReferenceTitle
- ReferenceURI
- Volume
- Issue
- Pages
- DetailLocation
- DayOfPubl
- MonthOfPubl
- YearOfPubl
- DateOfPublSupplement
- YearOnPubl
- DateOnPublSupplement
- Protologue
- ProtologueURI
- ProtologueResourceURI
- NameUsageReferences
- OriginalOrthography
- NomenclaturalCode
- NomenclaturalStatus
- NomenclaturalComment
- Typification
- TypificationDetails
- TypificationReferenceTitle
- TypificationReferenceURI
- TypificationNotes
- TypeSubstrate
- TypeLocality
- TypeSpecimenNotes
- AnamorphTeleomorph
- TypistNotes
- RevisionLevel
- IgnoreButKeepForReference
- DataWithholdingReason

TaxonCommonName
- NameID
- CommonName
- LanguageCode
- CountryCode

TaxonGeography
- NameID
- PlaceURI
- PlaceNameCache

TaxonNameProject
- NameID
- ProjectID

TaxonHierarchy
- ProjectID
- NameID
- IgnoreButKeepForReference
- NameParentID
- HierarchyRefURI
- HierarchyRefText
- HierarchyRefDetail
- HierarchyPositionIsUncertain
- HierarchyTypistsNotes
- HierarchyListCache

TaxonSynonymy
- ProjectID
- NameID
- SynNameID
- IgnoreButKeepForReference
- ConceptSuffix
- ConceptNotes
- SynRefURI
- SynRefText
- SynRefDetail
- SyniTypistsNotes
- SynType
- SynIsUncertain

TaxonAcceptedName
- ProjectID
- NameID
- IgnoreButKeepForReference
- ConceptSuffix
- ConceptNotes
- RefURI
- RefText
- RefDetail
- TypistsNotes

TaxonNameTypification
- NameID
- TypificationReferenceTitle
- TypificationReferenceURI
- Typification
- TypificationDetails
- TypificationNotes
- TypeSubstrate
- TypeLocality
- TypeSpecimenNotes

TaxonNameListReference
- NameID
- ProjectID
- TaxonNameListRefText
- TaxonNameListRefURI
- TaxonNameListRefDetail
- TaxonUsageContext
- ReferenceType
- Notes

TaxonNameList
- NameID
- ProjectID
- Notes

TaxonNameListDistribution
- NameID
- ProjectID
- PlaceURI
- PlaceNameCache
- Notes

TaxonNameListProjectProxy
- ProjectID
- Project

TaxonNameListArea
- ProjectID
- PlaceURI
- PlaceNameCache
- Notes

TaxonNameListCollectionSpecimen
- NameID
- ProjectID
- DisplayText
- CollectionSpecimenURI
- Role
- Description
- Notes

TaxonNameListAnalysisCategory
- AnalysisID
- AnalysisParentID
- DisplayText
- Description
- AnalysisURI
- Notes

TaxonNameListAnalysis
- NameID
- ProjectID
- AnalysisID
- AnalysisValue
- Notes

TaxonNameExternalDatabase
- ExternalDatabaseID
- ExternalDatabaseName
- ExternalDatabaseVersion
- Rights
- ExternalDatabaseAuthors
- ExternalDatabaseURI
- ExternalDatabaseInstitution
- InternalNotes
- ExternalAttribute_NameID
- PreferredSequence
- Disabled

TaxonNameListAnalysisCategoryValue
- AnalysisID
- AnalysisValue
- Description
- DisplayText
- DisplayOrder
- Notes

TaxonNameExternalID
- NameID
- ExternalDatabaseID
- ExternalNameURI

The description of the tables and the fields

Index

Table <u>TaxonAcceptedName</u>

TaxonAcceptedName contains the accepted names ("valid or preferred taxon names"). For each project, a name in "TaxonName" may either be accepted (a corresponding record in "TaxonAcceptedName" exists) or not. Examples for not accepted names are synonyms of accepted names and nomenclaturally rejected names.

Column	Data type	Description
ProjectID	int	Each project can have a different opinion regarding synonmy. Refers to the common project definition in the DiversityProjects module. *Default value: 0*
NameID	int	ID of the accepted name. Refers to the NameID of TaxonName (= foreign key).
IgnoreButKeepForReference	tinyint	If true, the record is ignored for all purposes of evaluation (because contradicted). It is kept only to maintain the cited reference. If no reference is given, it may be deleted instead. *Default value: 0*
ConceptSuffix	nvarchar (200)	Empty for default and primary concept; else 's. lat.', 's. str.', 'sec.', 'sensu Muell.', 'emend.' , '(Auct.)' etc. *Default value: ''*
ConceptNotes	nvarchar (500)	Notes on the concept of the name, e.g. incl. forms with black apothecia
RefURI	varchar (255)	ReferenceURI: Source publication where synonymization is published (not publication of name!)
RefText	nvarchar (255)	Free text, esp. where a RefURI is missing. Source publication where synonymization is published (not publication of name!)
RefDetail	nvarchar (255)	Esp. page number on which the synonymization is published
TypistsNotes	nvarchar (255)	An internal note of the responsible person concerning this synonymization. This information is NOT included in any report.
LogInsertedBy	nvarchar (50)	Name of user who first entered (typed or imported) the data. This is the operator (or typist) name, which may be different from the person responsible. *Default value: user_name()*
LogInsertedWhen	smalldatetime	Date and time when record was first entered (typed or imported) into this system. *Default value: getdate()*
LogUpdatedBy	nvarchar (50)	Name of user who last

Column	Data type	Description
		updated the data. This is the operator (or typist) name, which may be different from the person responsible. *Default value: user_name()*
LogUpdatedWhen	smalldatetime	Date and time when record was last updated. *Default value: getdate()*

Table __TaxonCommonName__

TaxonCommonName contains the common names that are assigned to a scientific name.

Column	Data type	Description
__NameID__	int	ID taxonomic name. Refers to the column NameID of the table TaxonName (= foreign key).
__CommonName__	nvarchar (220)	A common name of the taxonomic name
__LanguageCode__	varchar (2)	The 2-letter code of the language of the common name according to ISO
__CountryCode__	varchar (2)	The 2-letter code of the country where the common name is used according to ISO ISO 3166-1
__ReferenceTitle__	nvarchar (220)	The title of the reference where the common name was published
ReferenceURI	varchar (255)	The URI of the reference e.g. as provided by the module DiversityReferences
ReferenceDetails	nvarchar (500)	Esp. page number on which the common name is published
SubjectContext	nvarchar (500)	The context in which the common name is used, e.g. pharmacy, food
Notes	nvarchar (MAX)	Notes about the common name
LogInsertedBy	nvarchar (50)	Who inserted this dataset
LogInsertedWhen	smalldatetime	The time when this dataset was inserted
LogUpdatedBy	nvarchar (50)	Who was the last to update this dataset
LogUpdatedWhen	smalldatetime	The last time when this dataset was updated

Table **TaxonGeography**

TaxonGeography is containing the information about the geographic distribution according to the protologue.

Column	Data type	Description
NameID	int	ID of the name. Refers to the NameID of TaxonName (= foreign key).
PlaceURI	varchar (255)	URI (e.g. LSID) within e.g. the DiversityGazetteer for a place (which may have several names) as found in the protologue
PlaceNameCache	nvarchar (100)	The name of the place
LogInsertedBy	nvarchar (50)	Name of user who first entered (typed or imported) the data. *Default value: user_name()*
LogInsertedWhen	smalldatetime	Date and time when the data were first entered (typed or imported) into this database. *Default value: getdate()*
LogUpdatedBy	nvarchar (50)	Name of user who last updated the data. *Default value: user_name()*
LogUpdatedWhen	smalldatetime	Date and time when the data were last updated. *Default value: getdate()*

Table **TaxonHierarchy**

TaxonHierarchy contains information about the hierarchical position of the taxa. For each project, a name may be referred to a single parent taxon. The attribute "NameParentID" may, however, be Null, in which case "HierarchyPositionIsUncertain" should be true.

Column	Data type	Description
ProjectID	int	Each project can implement a different taxonomic hierarchy. Refers to the common project definition in the DiversityProjects module. *Default value: 0*
NameID	int	Unique NameID code of the higher taxon. Refers to the NameID code of LichenName (= foreign key).
IgnoreButKeepForReference	tinyint	If true, the record is ignored for all purposes of evaluation (because contradicted). It is kept only to maintain the

		cited reference. If no reference is given, it may be deleted instead. *Default value: 0*
NameParentID	int	Next higher taxon (e.g. the family or subfamily if this taxon is a genus)
HierarchyRefURI	varchar (255)	Reference URI: Source publication where the hierarchy is published (not publication of name!) as stored e.g. in DiversityReferences
HierarchyRefText	nvarchar (255)	Free citation, esp. where a HierarchyRefURI is missing. Source publication where the hierarchy is published (not publication of name!)
HierarchyRefDetail	nvarchar (255)	Esp. page number on which the hierarchy is published
HierarchyPositionIsUncertain	bit	If the position of this taxon within this taxonomic group is uncertain. *Default value: 0*
HierarchyTypistsNotes	nvarchar (255)	An internal note of the responsible person concerning this hierarchy. This information is NOT included in any report.
HierarchyListCache	nvarchar (1000)	CALCULATED FIELD: List of higher taxa for faster access: "div.; class; ord.; fam."
LogInsertedBy	nvarchar (50)	Name of user who first entered (typed or imported) the data. This is the operator (or typist) name, which may be different from the person responsible. *Default value: user_name()*
LogInsertedWhen	smalldatetime	Date and time when record was first entered (typed or imported) into this system. *Default value: getdate()*
LogUpdatedBy	nvarchar (50)	Name of user who last updated the data. This is the operator (or typist) name, which may be different from the person responsible. *Default value: user_name()*
LogUpdatedWhen	smalldatetime	Date and time when record was last updated. *Default value: getdate()*

Table **TaxonName**

TaxonName is the nomenclatural information. Most items are expected to be undisputed nomenclatural facts. Some data items include editorial opinion (nomenclatural status, but also the accepted spelling which may involve orthographical or grammatical corrections as required by ICBN). However, it is expected that within a project agreement can be achieved on the information contained in TaxonName, i.e. collaborators can "correct" information without consultation.

Column	Data type	Description
NameID	int	The ID of a name in DiversityTaxonNames_Fungi (primary key)
TaxonNameCache	nvarchar (255)	A full unique version of the name. Generated by the database, not entered by the user (candidate key)
Version	smallint	The version of a name record (revision number, internally filled by system) *Default value: (1)*
BasedOnNameID	int	The Basionym of this name, resp. the NameID of the Basionym. A name is a Basionym, if NameID = BasionymID.
CreationType	nvarchar (50)	E.g.: taxon based on new type, combination based on a previously publ. name ('comb. nov.'), new name ('nom.nov.') introduced to replace a homonym (may occur for genera!), validation of previously invalidly publ. taxon name ('ex'), or unknown.
TaxonomicRank	nvarchar (50)	Taxonomic rank of the taxon (var., subsp., species, genus, family, order, etc.). The rank must be selected from the associated list of ranks (= TaxonomyRank) *Default value: N'sp.'*
GenusOrSupragenericName	nvarchar (200)	If rank is above species: Name of taxon above species level (currently accepted spelling). Includes infrageneric taxon names, genera, families, etc.
SpeciesGenusNameID	int	If rank is species or below: NameID of the Genus name. Refers to same table with TaxonomicRank = genus.
InfragenericEpithet	nvarchar (200)	If rank below genus and above species: Name of

		infrageneric taxon above species level (currently accepted spelling). Includes subgenus, series, etc.
SpeciesEpithet	nvarchar (100)	The species name part of the species name, for example 'alba' in 'Abies alba'.
InfraspecificEpithet	nvarchar (100)	The epithet of the infraspecific entity
BasionymAuthors	nvarchar (100)	The authors of a newly created name. For all taxonomic ranks, only for 'comb. nov.' or 'nom. nov.': Author(s) of the basionym (will be displayed in '()', do not enter the parentheses), abbreviated according to authors standard
CombiningAuthors	nvarchar (255)	The names of the combining authors if the name is base on another older name (e.g. combined into a different genus)
PublishingAuthors	nvarchar (255)	If the authors of the taxon differ from the authors of the publication: the latter (...'in ' Publishing authors ...)
SanctioningAuthor	nvarchar (100)	Sanctioning is a special instrument available for fungi to allow the starting date of taxonomy to be identical with higher plants, but preserve names used by Fries (Fr.) and Person (Pers.)
NonNomenclaturalNameSuffix	nvarchar (200)	A suffix for concept names not included in any nomenclatural code. If present fields for authors and typification must be empty and NomenclaturalCode set to 'not under code'.
IsRecombination	bit	True if the name is a recombination *Default value: (0)*
IsHybrid	bit	If checked (or 'True'), the new taxon is a hybrid with or without a hybrid epithet. *Default value: (0)*
HybridNameID1	int	If IsHybrid is checked (or 'True'): Name of hybrid species 1. Refers to NomID code of publ. TaxonomyName (= foreign key).
HybridNameID2	int	If IsHybrid is checked (or 'True'): Name of hybrid species 2. Refers to NomID

		code of publ. TaxonomyName (= foreign key).
HybridNameID3	int	If IsHybrid is checked (or 'True'): Name of hybrid species 3. Refers to NomID code of publ. TaxonomyName (= foreign key).
HybridNameID4	int	If IsHybrid is checked (or 'True'): Name of hybrid species 4. Refers to NomID code of publ. TaxonomyName (= foreign key).
ReferenceTitle	nvarchar (600)	The title of the publication where the name was published. Note this is only a cached value where ReferenceURI is present
ReferenceURI	varchar (255)	URI (e.g. LSID) of Reference, referes to table ReferenceTitle in Database DiversityReferences: Source publication where name is published
Volume	nvarchar (20)	The volume of the journal
Issue	nvarchar (255)	The issue of the literature
Pages	nvarchar (50)	The pages within the literature
DetailLocation	nvarchar (200)	Additional information like plates etc.
DayOfPubl	tinyint	The day when the name was published
MonthOfPubl	tinyint	The month when the name was published
YearOfPubl	smallint	The year when the name was published
DateOfPublSupplement	nvarchar (200)	Verbal or additional date information, e.g. 'end of summer 1985', 'first quarter', '1888-1892'
YearOnPubl	smallint	The year cited on the original paper as year of publication
DateOnPublSupplement	nvarchar (200)	Verbal or additional date information, e.g. 'end of summer 1985', 'first quarter', '1888-1892'
Protologue	nvarchar (MAX)	Full text of the protologue. If protologue is provided in Latin and in another language languages, both only the Latin or several languages may be stored together.
ProtologueURI	varchar (255)	URI of externally available Protologue information (scanned image, full text from

		external provider, etc.)
ProtologueResourceURI	varchar (255)	The ResourceID of an image of the protologue as stored in the module DiversityResources. Primary key of table Resource in the database DiversityResources.(= Foreign key)
NameUsageReferences	nvarchar (255)	Indexing volumes like Index of Fungi or any Name usage that caused this name to be added to the database
OriginalOrthography	nvarchar (255)	The original spelling of the name
NomenclaturalCode	nvarchar (50)	Code of Nomenclature under which this taxon was created: 'Bacteriology', 'Botany' (incl. Mycology), 'Zoology', 'Biocode' (for future use), 'Non nomenclatural name'. Default value: 'Botany'
NomenclaturalStatus	nvarchar (50)	Categories for effective/valid/legitimate... esp. 'nom. illeg.', 'nom. inval.', 'nom. nudum', etc.
NomenclaturalComment	nvarchar (MAX)	Comments on the nomenclature. e.g. 'according to ICBN Art. 39.1'
Typification	nvarchar (50)	The status of the type specimen(s) as written in the protologue, e.g. holotype
TypificationDetails	nvarchar (255)	Details concerning the typification as written in the protologue, e.g. if just parts of a specimen were accepted as the type
TypificationReferenceTitle	nvarchar (255)	The title of the publication where the typification was published. Note this is only a cached value where Typification-LiteratureLink is present
TypificationReferenceURI	varchar (255)	URI (e.g. LSID) of Reference, refers to e.g DiversityReferences: Source publication where the typification is published
TypificationNotes	nvarchar (200)	Notes concerning the typification
TypeSubstrate	nvarchar (255)	The substrate the type was growing on as written in the protologue
TypeLocality	nvarchar (255)	The locality where the type was found as written in the

		protologue
TypeSpecimenNotes	nvarchar (200)	Notes about the type specimen, e.g. the herbarium where a type specimen is located, Collector, collection date/number, etc.
AnamorphTeleomorph	nvarchar (255)	If the name is related to a anamorph or a telemorph as written in the protologue
TypistNotes	nvarchar (200)	Additional notes and problems
RevisionLevel	nvarchar (50)	The level of the revision of the taxonomic name, e.g. 'unchecked', 'fully checked'
IgnoreButKeepForReference	bit	If true, the record is ignored for all purposes of evaluation (because contradicted). It is kept only to maintain the cited reference. If no reference is given, it may be deleted instead. *Default value: (0)*
DataWithholdingReason	nvarchar (255)	If the dataset is withhold, the reason for withholding the data, otherwise null
LogInsertedBy	nvarchar (50)	Name of user who first entered (typed or imported) the data. This is the operator (or typist) name, which may be different from the person responsible. *Default value: user_name()*
LogInsertedWhen	smalldatetime	Date and time when record was first entered (typed or imported) into this system. *Default value: getdate()*
LogUpdatedBy	nvarchar (50)	Name of user who last updated the data. This is the operator (or typist) name, which may be different from the person responsible. *Default value: user_name()*
LogUpdatedWhen	smalldatetime	Date and time when record was last updated. *Default value: getdate()*

Table TaxonNameExternalDatabase

TaxonNameExternalDatabase document the sources of the names.

Column	Data type	Description
ExternalDatabaseID	int	An ID to identify an external data collection of plant names (primary key, the ID has no

		meaning outside of the DiversityWorkbench system)
ExternalDatabaseName	nvarchar (255)	The name of the data collection that has been integrated or can be linked to for further analysis
ExternalDatabaseVersion	nvarchar (255)	The version of this data collection (either official version number, or dates when the collection was integrated)
Rights	nvarchar (500)	A description of copyright agreements or permission to use data from the external database
ExternalDatabaseAuthors	nvarchar (200)	The persons or institutions responsible for the external database
ExternalDatabaseURI	nvarchar (300)	The URI of the database provider resp. the external database
ExternalDatabaseInstitution	nvarchar (300)	The institution responsible for the external database
InternalNotes	nvarchar (1500)	Additional notes concerning this data collection
ExternalAttribute_NameID	nvarchar (255)	The table and field name in the external data collection to which TaxonNameExternalID refers
PreferredSequence	tinyint	For selection in e.g. picklists: of several equal names only the name from the source with the lowest preferred sequence will be provided.
Disabled	bit	If this source should be disabled for selection of names e.g. in picklists

Table TaxonNameExternalID

TaxonNameExternalID document the source of a name together with the ID of the source.

Column	Data type	Description
NameID	int	The ID of a name in DiversityTaxonNames (foreign key + part of primary key: the name string associated with NameID may occur in multiple external databases)
ExternalDatabaseID	int	The ID of an external taxonomic data collection as defined in

		TaxontNameExternalDatabase (foreign key + part of primary key)
ExternalNameURI	varchar (255)	The URI (e.g. a LSID) of the external name as defined in the external database

Table TaxonNameList

The table is containing the names listed in e.g. a checklist corresponding to the project.

Column	Data type	Description
NameID	int	ID of the name. Refers to the NameID of TaxonName (= foreign key).
ProjectID	int	Each project can may contain one taxon list. Refers to the common project definition in the DiversityProjects module.
Notes	nvarchar (MAX)	An note concerning this entry.
LogInsertedBy	nvarchar (50)	Name of user who first entered (typed or imported) the data. *Default value: user_name()*
LogInsertedWhen	smalldatetime	Date and time when the data were first entered (typed or imported) into this database. *Default value: getdate()*
LogUpdatedBy	nvarchar (50)	Name of user who last updated the data. *Default value: user_name()*
LogUpdatedWhen	smalldatetime	Date and time when the data were last updated. *Default value: getdate()*

Table TaxonNameListAnalysis

Analysis values for list entries in the database, "Red list category: R", "Time of observation: Sept. - Nov."

Column	Data type	Description
NameID	int	ID of the name. Refers to the NameID of TaxonName (= foreign key).
ProjectID	int	Each project can may contain one taxon list. Refers to the common project definition in the DiversityProjects module.
AnalysisID	int	ID of the analysis (= foreign key)

AnalysisValue	nvarchar (255)	The result of the analysis
Notes	nvarchar (MAX)	Notes concerning this analysis
LogInsertedBy	nvarchar (50)	Name of user who first entered (typed or imported) the data. *Default value: user_name()*
LogInsertedWhen	smalldatetime	Date and time when the data were first entered (typed or imported) into this database. *Default value: getdate()*
LogUpdatedBy	nvarchar (50)	Name of user who last updated the data. *Default value: user_name()*
LogUpdatedWhen	smalldatetime	Date and time when the data were last updated. *Default value: getdate()*

Table **TaxonNameListAnalysisCategory**

Analysis types used within the database, e.g. "Red list category", "Frequency"

Column	Data type	Description
AnalysisID	int	ID of the analysis (Primary key)
AnalysisParentID	int	Analysis ID of the parent analysis if it belongs to a certain type documented in this table
DisplayText	nvarchar (50)	Name of the analysis as e.g. shown in user interface
Description	nvarchar (MAX)	Description of the analysis
AnalysisURI	varchar (255)	URI referring to an external documentation of the analysis
Notes	nvarchar (MAX)	Notes concerning this analysis
LogCreatedWhen	datetime	The time when this dataset was created *Default value: getdate()*
LogCreatedBy	nvarchar (50)	Who created this dataset *Default value: user_name()*
LogUpdatedWhen	datetime	The last time when this dataset was updated *Default value: getdate()*
LogUpdatedBy	nvarchar (50)	Who was the last to update this dataset *Default value: user_name()*

Table **TaxonNameListAnalysisCategoryValue**

Value lists for analysis types with predefined values, e.g. "0, 1, 2, 3, ..." for Red list category. Includes description etc. for the values in the list.

Column	Data type	Description
AnalysisID	int	ID of the analysis (Primary key)
AnalysisValue	nvarchar (255)	The categorized value of the analysis
Description	nvarchar (500)	Description of enumerated object, displayed in the user interface
DisplayText	nvarchar (50)	Short abbreviated description of the object, displayed in the user interface
DisplayOrder	smallint	The order in which the entries are displayed. The order may be changed at any time, but all values must be unique.
Notes	nvarchar (500)	Internal development notes about usage, definition, etc. of an enumerated object
LogInsertedBy	nvarchar (50)	Name of user who first entered (typed or imported) the data. *Default value: user_name()*
LogInsertedWhen	smalldatetime	Date and time when the data were first entered (typed or imported) into this database. *Default value: getdate()*
LogUpdatedBy	nvarchar (50)	Name of user who last updated the data. *Default value: user_name()*
LogUpdatedWhen	smalldatetime	Date and time when the data were last updated. *Default value: getdate()*

Table **TaxonNameListArea**

The area defined for the whole taxon list

Column	Data type	Description
ProjectID	int	Each project can have a different opinion regarding the parameters defined for a list. Refers to the common project definition in the DiversityProjects module.
PlaceURI	varchar (255)	URI or identifier for a place (which may have several names) derived from e.g. TDWG
PlaceNameCache	nvarchar (255)	The name of the place

Notes	nvarchar (MAX)	An note concerning this entry.
LogInsertedBy	nvarchar (50)	Name of user who first entered (typed or imported) the data. *Default value: user_name()*
LogInsertedWhen	smalldatetime	Date and time when the data were first entered (typed or imported) into this database. *Default value: getdate()*
LogUpdatedBy	nvarchar (50)	Name of user who last updated the data. *Default value: user_name()*
LogUpdatedWhen	smalldatetime	Date and time when the data were last updated. *Default value: getdate()*

Table TaxonNameListCollectionSpecimen

The collection specimens on which the taxon list is based

Column	Data type	Description
NameID	int	ID of the name. Refers to the NameID of TaxonName (= foreign key).
ProjectID	int	ProjectID of the taxon list. Foreign key and part of primary key.
DisplayText	varchar (255)	The name of the collection specimen as shown e.g. in a user interface. Part of the primary key
CollectionSpecimenURI	varchar (255)	URI of the collection specimen. May refer to the module CollectionSpecimen
Role	nvarchar (50)	The role of the specimen in relation to the taxon list (= foreign key, see table TaxonNameListSpecimenRole_ Enum)
Description	nvarchar (MAX)	Description of the collection specimen
Notes	nvarchar (MAX)	Notes about the collection specimen
LogInsertedBy	nvarchar (50)	Name of user who first entered (typed or imported) the data.
LogInsertedWhen	smalldatetime	Date and time when the data were first entered (typed or imported) into this database.
LogUpdatedBy	nvarchar (50)	Name of user who last updated the data.
LogUpdatedWhen	smalldatetime	Date and time when the data

		were last updated.

Table TaxonNameListDistribution

The geographical distrbution for the organisms within the taxon list.

Column	Data type	Description
NameID	int	ID of the name. Refers to the NameID of TaxonName (= foreign key).
ProjectID	int	Each project can may contain one taxon list. Refers to the common project definition in the DiversityProjects module.
PlaceURI	varchar (255)	URI or identifier for a place (which may have several names) derived from e.g. TDWG
PlaceNameCache	nvarchar (255)	The name of the place
Notes	nvarchar (MAX)	Notes concerning this analysis
LogInsertedBy	nvarchar (50)	Name of user who first entered (typed or imported) the data. *Default value: user_name()*
LogInsertedWhen	smalldatetime	Date and time when the data were first entered (typed or imported) into this database. *Default value: getdate()*
LogUpdatedBy	nvarchar (50)	Name of user who last updated the data. *Default value: user_name()*
LogUpdatedWhen	smalldatetime	Date and time when the data were last updated. *Default value: getdate()*

Table TaxonNameListProjectProxy

The taxon lists defined within the database. Links to the module DiversityProject

Column	Data type	Description
ProjectID	int	ID of the Project, Primary key. May be related to module DiversityProjects
Project	nvarchar (50)	The name of the project. May be related to module DiversityProjets

Table **TaxonNameListReference**

The reference where the taxon list resp. the name is published

Column	Data type	Description
NameID	int	ID of the name. Refers to the NameID of TaxonName (= foreign key).
ProjectID	int	Each project can may contain one taxon list. Refers to the common project definition in the DiversityProjects module.
TaxonNameListRefText	nvarchar (255)	Free text, esp. where a TaxonNameListRefURI is missing. Source publication where distribution is published (not publication of name!)
TaxonNameListRefURI	varchar (255)	Reference URI: Source publication where the distribution is published (not publication of name!) as stored e.g. in DiversityReferences
TaxonNameListRefDetail	nvarchar (255)	Esp. page number on which the distribution is published
TaxonUsageContext	nvarchar (50)	The context of the usage of the taxonomic name within the reference, e.g. ecology
ReferenceType	nvarchar (50)	The type of the reference, e.g. a webpage
Notes	nvarchar (MAX)	An note concerning this entry.
LogInsertedBy	nvarchar (50)	Name of user who first entered (typed or imported) the data. *Default value: user_name()*
LogInsertedWhen	smalldatetime	Date and time when the data were first entered (typed or imported) into this database. *Default value: getdate()*
LogUpdatedBy	nvarchar (50)	Name of user who last updated the data. *Default value: user_name()*
LogUpdatedWhen	smalldatetime	Date and time when the data were last updated. *Default value: getdate()*

Table **TaxonNameProject**

TaxonNameProject documents the projects of the names and provides access to the data.

Column	Data type	Description

		The ID of a name in DiversityTaxonNames (foreign key + part of primary key: the name may occur in multiple projects)
NameID	int	
ProjectID	int	ID of the project. Refers to the common project definition in the DiversityProjects module. *Default value: 0*
LogInsertedWhen	datetime	The time when this dataset was inserted *Default value: getdate()*
LogInsertedBy	nvarchar (50)	Who inserted this dataset *Default value: user_name()*
LogUpdatedWhen	datetime	The last time when this dataset was updated *Default value: getdate()*
LogUpdatedBy	nvarchar (50)	Who was the last to update this dataset *Default value: user_name()*

Table TaxonNameTypification

The typification of a taxonomic name, e.g. where the type information of the name was published

Column	Data type	Description
NameID	int	The ID of a name in DiversityTaxonNames_Fungi (primary key)
TypificationReferenceTitle	nvarchar (255)	The title of the publication where the typification was published. Note this is only a cached value where Typification-LiteratureLink is present
TypificationReferenceURI	varchar (255)	URI (e.g. LSID) of Reference, refers to e.g DiversityReferences: Source publication where the typification is published
Typification	nvarchar (50)	The status of the type specimen(s) as written in the protologue, e.g. holotype
TypificationDetails	nvarchar (255)	Details concerning the typification as written in the protologue, e.g. if just parts of a specimen were accepted as the type
TypificationNotes	nvarchar (200)	Notes concerning the typification
TypeSubstrate	nvarchar (255)	The substrate the type was

		growing on as written in the protologue
TypeLocality	nvarchar (255)	The locality where the type was found as written in the protologue
TypeSpecimenNotes	nvarchar (200)	Notes about the type specimen, e.g. the herbarium where a type specimen is located, Collector, collection date/number, etc.
LogInsertedBy	nvarchar (50)	Name of user who first entered (typed or imported) the data. This is the operator (or typist) name, which may be different from the person responsible. *Default value: user_name()*
LogInsertedWhen	smalldatetime	Date and time when record was first entered (typed or imported) into this system. *Default value: getdate()*
LogUpdatedBy	nvarchar (50)	Name of user who last updated the data. This is the operator (or typist) name, which may be different from the person responsible. *Default value: user_name()*
LogUpdatedWhen	smalldatetime	Date and time when record was last updated. *Default value: getdate()*

Table **TaxonSynonymy**

TaxonSynonymy contains names, which are synonym to other names. Here a single name may be a synonym of multiple accepted names ("pro parte synonym"). The "pro parte" or corresponding sensu information is captured in the "ConceptSuffix" attribute, which is part of the primary key.
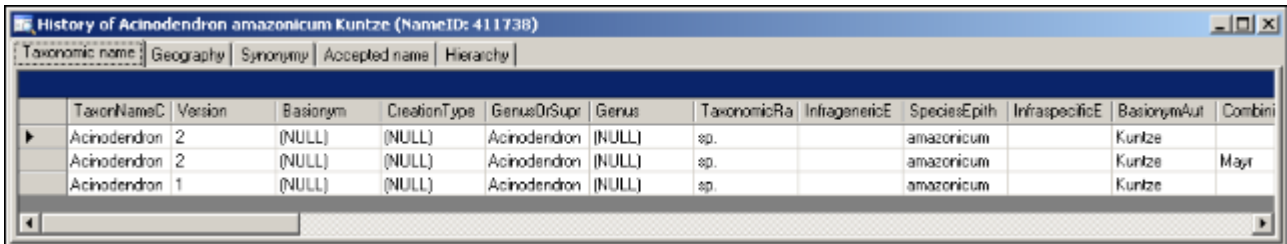
Column	Data type	Description
ProjectID	int	Each project can have a different opinion regarding synonmy. Refers to the common project definition in the DiversityProjects module. *Default value: (0)*
NameID	int	ID of the synonymized name. Refers to the NameID of TaxonName (= foreign key).
SynNameID	int	The ID of the synonym to which this name has been assigned to. Refers to the NameID of TaxonName (= foreign key).

IgnoreButKeepForReference	tinyint	If true, the record is ignored for all purposes of evaluation (because contradicted). It is kept only to maintain the cited reference. If no reference is given, it may be deleted instead. *Default value: (0)*
ConceptSuffix	nvarchar (200)	Empty for default and primary concept; else 's. lat.', 's. str.', 'sec.', 'sensu Muell.', 'emend.', '(Auct.)' etc. *Default value: ''*
ConceptNotes	nvarchar (500)	Notes on the concept of the name, e.g. incl. forms with black apothecia
SynRefURI	varchar (255)	Reference URI: Source publication where synonymization is published (not publication of name!) as stored e.g. in DiversityReferences
SynRefText	nvarchar (255)	Free text, esp. where a SynRefURI is missing. Source publication where synonymization is published (not publication of name!)
SynRefDetail	nvarchar (255)	Esp. page number on which the synonymization is published
SynTypistsNotes	nvarchar (255)	An internal note of the responsible person concerning this synonymization. This information is NOT included in any report.
SynType	nvarchar (50)	Type of the synonymization, e.g. 'heterotypic', 'homotypic' *Default value: 'unknown'*
SynIsUncertain	bit	If the synoymization is uncertain. Corresponding to =? for heterotypic synonyms and ? if the type of the synonymisation is unknown. Homotypic synonyms can not be uncertain. *Default value: (0)*
LogInsertedBy	nvarchar (50)	Name of user who first entered (typed or imported) the data. This is the operator (or typist) name, which may be different from the person responsible. *Default value: user_name()*
LogInsertedWhen	smalldatetime	Date and time when record was first entered (typed or imported) into this system.

		Default value: getdate()
LogUpdatedBy	nvarchar (50)	Name of user who last updated the data. This is the operator (or typist) name, which may be different from the person responsible. *Default value: user_name()*
LogUpdatedWhen	smalldatetime	Date and time when record was last updated. *Default value: getdate()*

History

To inspect the history of a dataset click on the ![button] button. A form will open, showing all former states of the data in the tables with the current dataset at the top.

The version will be set automatically. If a dataset is changed the version will be increased if the last changes where done by a different user or the last change is more than 24 hours ago (for further details see topic Logging).

Statistics

To see the activity and the state of revision within a project choose **Help - Statistics ...** from the menu. A form as shown below will open containing a summary of the project and for the current user.

Backup

If you need to backup your database, you have to use the functionality provided by SQL-Server. To do this, you need administration rights in the database you want to create a backup. Open the Enterprise Manager for SQL-Server, choose the database and detach it from the server as shown in the image below.
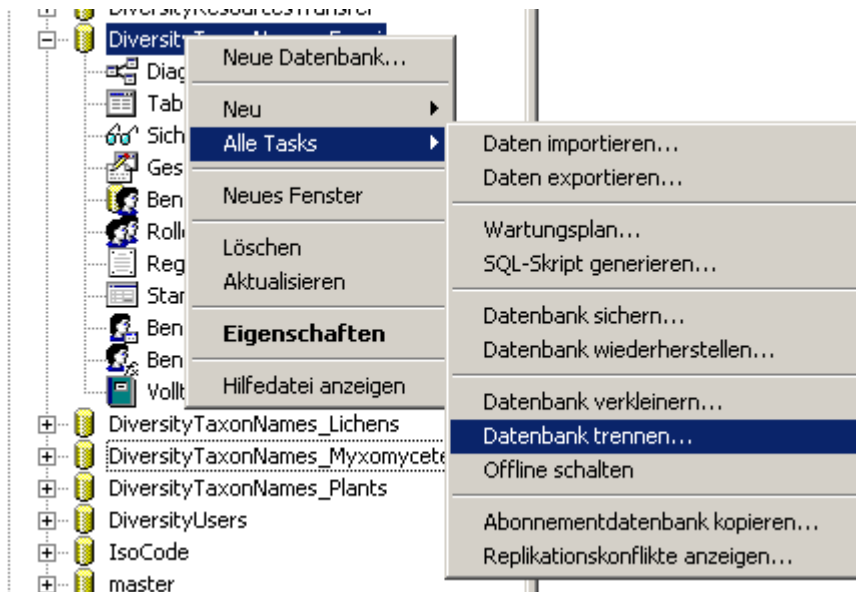
After detaching the database, you can store a copy of the ..._Data.MDF File to keep it as a backup.

After storing the backup you have to attach the database.
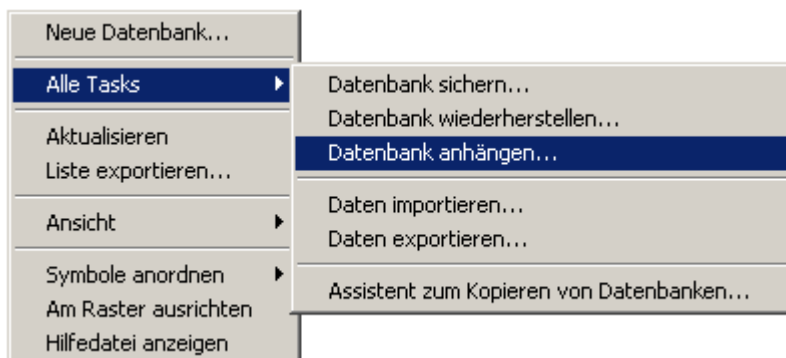
A dialog will appear where you have to select the original database file in your directory.

Datenbank anfügen - BSM5

Anzufügende MDF-Datei oder Datenbank:

`D:\Database\DiversityTaxonNames_Lichens2\DiversityTaxon` [...] Überprüfen

Ursprüngliche(r) Dateiname(n)	Speicherort für aktuelle Datei(en)
DiversityTaxonNames_Liche...	☑ D:\Database\DiversityTaxonNames_Lichens
DiversityTaxonNames_Liche...	☑ D:\Database\DiversityTaxonNames_Lichens

Anfügen als: `DiversityTaxonNames_Lichens2`

Datenbankbesitzer angeben: `BSM5\Administrator` ▼

[OK] [Abbrechen] [Hilfe]
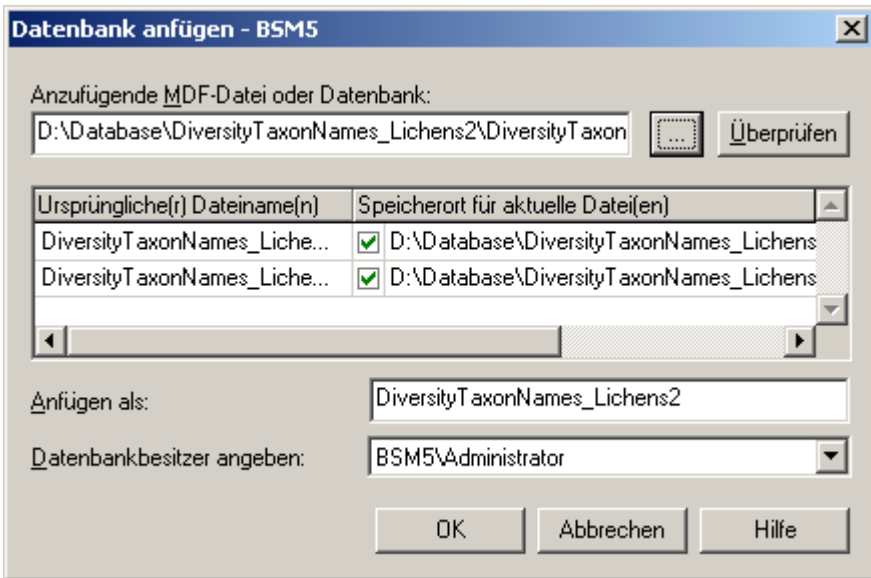
Logging

Changes within the database will be documented for each dataset with the time and the responsible user in the columns shown in the image below.

Spaltenname	Datentyp-Kurzform	Beschreibung
LogInsertedBy	nvarchar(50)	Name of user who first entered (typed or imported) the data. This is the operator (or typist) name, which may be different from the person responsible.
LogInsertedWhen	small datetime	Date and time when record was first entered (typed or imported) into this system.
LogUpdatedBy	nvarchar(50)	Name of user who last updated the data. This is the operator (or typist) name, which may be different from the person responsible.
LogUpdatedWhen	small datetime	Date and time when record was last updated.

All main tables have a corresponding logging table. If you change or delete a dataset the orignial dataset will be stored in this logging table together with information about who has done the changes and when it happend. To inspect the history of a dataset click on the button [image] (see topic [History](#) for further information)
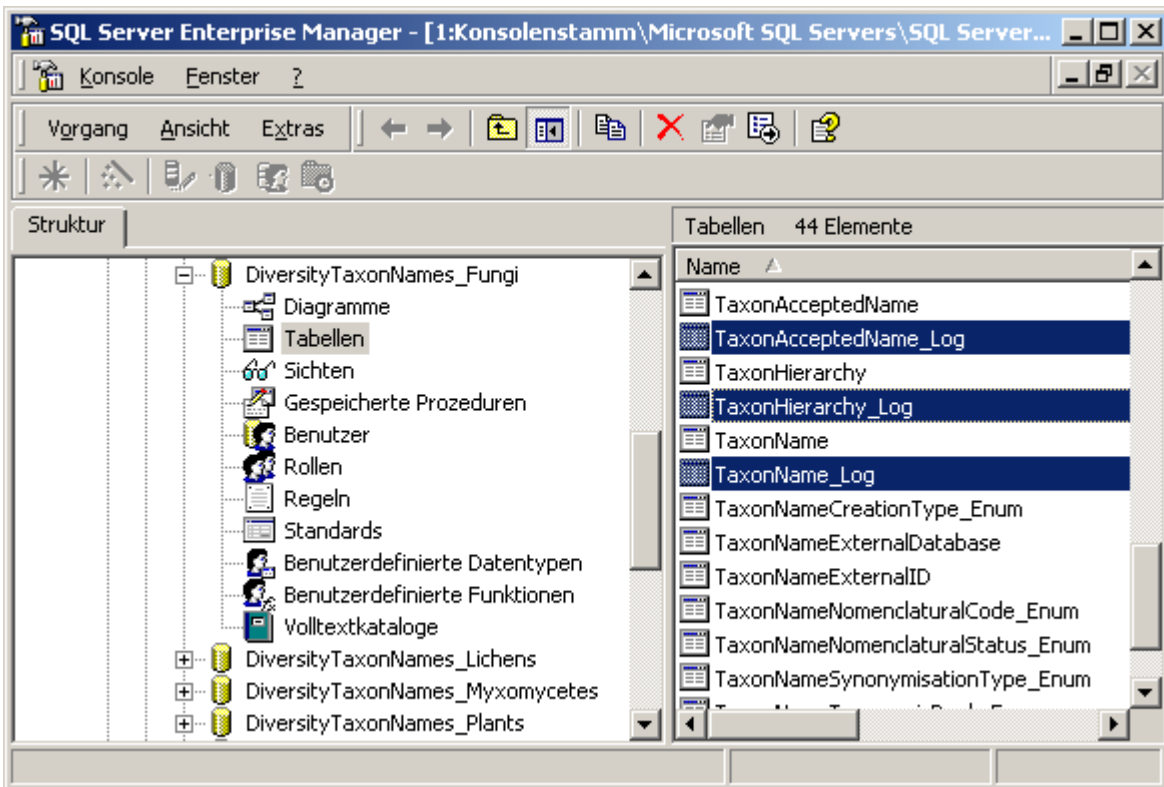
The logging tables (right side in image below) have the same structure as the data tables (left side in image below) and some additional fields (LogState, LogDate, LogUser, LogVersion and LogID) to document the logging as shown in the figure below.

LogInsertedWhen	datetime	LogInsertedWhen	datetime	The time when this dataset was inserted
LogInsertedBy	nvarchar(50)	LogInsertedBy	nvarchar(50)	The login of the user who is responsible for the insert of this dataset
LogUpdatedWhen	datetime	LogUpdatedWhen	datetime	The last time when this dataset was updated
LogUpdatedBy	nvarchar(50)	LogUpdatedBy	nvarchar(50)	The login of the user who is responsible for the last to update of this dataset
LogState	char(1)			The state of the change, U = Updated, D = Deleted
LogDate	datetime			The date when the original dataset was changed
LogUser	nvarchar(50)			The user who changed the original dataset
LogVersion	int			The version of the corresponding dataset in the main table
LogID	int			Primary key of the logging table

The field LogVersion is only present in tables dependent on a version main table. This main table keeps the version valid for the dataset including data in dependent tables. If a dataset in a dependent table is changed, the version of the main table will be updated and the entries in the logging table for the dependent table will record the version of the main table before the changes took place.