i4Life Darwin Core Archive Profile

Version 1.6, 17 August 2012

Editor: Richard White

Contributors: Wouter Addink, Kwok Cheung, Alastair Culham, Viktoras Didziulis, Markus Döring, Anton Güntsch, Andrew Jones, Patrick Leary, Cherian Mathew, Andreas Müller, Magda Sitko

Capacities Programme of Framework 7: EC e-Infrastructure Programme – Virtual Research Communities - INFRA-2010-2

Grant Agreement No:	261555
Project Co-ordinator:	Dr Alastair Culham
Project Homepage:	http://www.i4Life.eu
Duration of Project:	36 months
Start Date:	1 November 2010
End Date:	31 October 2013

Contents

Introduction

It is desirable to have a common format for the exchange of checklists among partners in i4Life and in the wider biodiversity informatics community. It has been agreed to use GBIF's Darwin Core Archive (DwCA) format for this purpose, but the GBIF and TDWG documentation permits or does not precisely define a wide variety of options. This might make life easier for checklist producers, but it makes the task of importing a checklist more difficult. In particular, implementing an import facility requires precision about which features are required and how these required features will be specified and used, which features are desirable or optional, which features may be ignored by some importers, and which will cause failure of the import process. According to the TDWG "Simple Darwin Core" pages [1], "it is up to applications to enforce further restrictions [on the use of the DwCA format] if appropriate, and it is up to the stakeholders of those applications to decide what the restrictions will be for the purpose the application is trying to serve." Thus further clarification of this format is required to permit its use as a reliable common format for producing and receiving checklists in the i4Life project, which will allow not only CoL data but checklists from other partners to be represented without data loss and with reasonable ease of use.

Here, we provide suggestions for using the DwCA profile which was previously agreed for the CoL Download Service, with extensions and minor deviations [2], to achieve a common checklist format for the exchange of checklists, and propose to define more precisely how some optional features should be handled. This document should be read in conjunction with the GBIF and CoL Download Service documentation, which contain more detailed explanations of the construction of DwCA files.

This document will form the basis for a common checklist exchange format that we hope will become a standard but presently should not be regarded as a rigid definition. Rather, it is intended to provide a means to help identify and describe the differences which may exist among the checklists being transferred between partners, and to help reduce unnecessary differences. It is not "cast in stone" and may be revised if clarifications or improvements are identified which do not cause undue difficulties for some users. Some differences in usage may be inevitable as the needs of partners vary and significant effort would be required by checklist producers and importers to adopt a single standard for all purposes without some variations however we expect partners to do all they can to meet the standard in the medium term.

Furthermore, not all producers may possess all the data elements described in this document. It is intended as a guide to good practice in providing the data which is available; in some cases fields may be empty or omitted. It is beyond the scope of this document to recommend a minimum data set, and in any case different partners in i4Life have different objectives and modes of working, so that a data type which is vital for one partner may be irrelevant to another. Each partner in the i4Life project is free to decide, and importantly to document, the extent to which files they produce comply with these suggestions and the use of any additional data elements they include.

> *Note: Except for possible additional fields relating directly to taxon names, this document does not discuss any additional data which might be included in a checklist for certain purposes by particular users. Any such data is assumed to follow DwCA principles and guidelines, which mandate that additional data tables are used within the DwCA zipped archive file. It is assumed that any checklist import software will use or ignore such extra tables, depending on its requirements. If a previously received DwCA*

file is passed on to a third party recipient, after possible modification, it is likely to be desirable that the additional data tables remain or are reinserted in the DwCA file for use by the third party. The use of the cross-mapping tools in the "new taxa awareness" ("piping") workflow has been recognised as an example of such a scenario.

Darwin Core Archive format

The description and specification of DwCA by GBIF is spread over a number of documents, but GBIF has recently published a guide to their use supported by the i4Life project [3], which provides instructions intended for biodiversity data administrators on sharing species checklists. It provides a step-by-step overview of how to publish species checklists, serves as a quick reference for getting started and provides links to other GBIF data publishing documents that provide more details on specific components of the data publishing workflow. It describes the main GBIF technical options for constructing checklists, including the Integrated Publishing Toolkit (IPT), the Spreadsheet Processor which offers limited means to produce DwCA via spreadsheet templates, and some resources to assist in manual assembly (the DwCA Validator, the DwCA Assistant and the GBIF Metadata Profile schema). Other GBIF references are also cited in the present document.

Darwin Core Archive (DwCA) [4, 5] is a very general specification documented by GBIF on a web page [6] as an application of the Darwin Core text file format described by TDWG [7]. The format for taxonomic checklists discussed in the present document is also related to the 'GNA profile' [8], a particular expression (or profile) of the DwCA format.

The minimum requirement is that the DwCA file is an archive conforming to the "zip file" industry standard, containing at least a 'core taxon file' representing a table in which each record denotes a "taxon name usage", that is either the accepted name or synonym of a taxon. The record containing the accepted name is effectively also the record for the taxon concept itself. Taxa may be at any level in the taxonomic hierarchy, not just species. The GBIF Darwin Core documentation for the core taxon file [9] gives two alternative ways to specify the fields used, one in which the fields are named in the first record of the core taxon file and another using an optional 'archive descriptor' file (named "meta.xml").

The archive descriptor file is, if present, stored in the same zip archive, and can be used as a map to describe the core taxon file and any extensions held in other files. Each field or column of each file is identified and described so that the whole schema can be interpreted. Each field is identified by means of the name of a term in a known vocabulary, frequently the TDWG Darwin Core vocabulary, but Dublin Core and other vocabularies are used, especially for extensions. The archive descriptor file, if used, can be prepared once and does not need to change if the archive format remains the same but the data contents change, for example from one edition of a provider's checklist to the next.

Users

The following users of a checklist transfer format are envisaged, and it is desirable that the checklists they exchange are fully compatible:

Producers

- CoL/i4Life WP4 Enhanced Download Service [10]
- i4Life partners who want to send or publish checklists for cross-mapping (IUCN, EBI, etc.)

- The i4Life cross-mapping service providers, when supplying lists similar to checklists, such as lists of unmatched names and taxa for the GSD 'piping' service intended for distributing new names or taxa to GSDs. (These lists are different from the cross-maps linking two or more checklists, since cross-maps require a different format.)
- Other producers of checklists, such as regional hubs supplying checklists to the CoL Regional Network as envisaged in the 4D4Life Work Package 4 and OpenBio projects

Importers

- Operators of the GSD 'piping' service, as mentioned above
- The i4Life cross-mapping service providers, for importing checklists for cross-mapping
- i4Life partners who want to receive checklists (from the CoL Download Service or other providers)
- Other consumers of checklists (for example, cross-mapping in the CoL Regional Network, OpenBio and EDIT [11])

Components of the 4D4Life "e2" architecture also use a DwCA format [12] when transferring data between data stores.

Requirements

Based on the needs of the users listed above, it is assumed that a common format will include or permit the following features:

- It will support the accepted names and synonyms of taxa, including species and higher- and lower-level taxa. "Synonyms" are to be interpreted broadly as referring to any name linked to a taxon, which may include misapplied names, homonyms and incorrectly classified taxa.
- It will include any unique identifiers used by the providers for the taxa included in the checklist. These may include both globally unique identifiers, if present, and internal database identifiers (which are often needed by users to access further information about a taxon, including records in other tables in the DwCA).
- It will include any unique identifiers in use by the providers for names, if available.
- It will permit the inclusion of generated unique identifiers for taxa (and names if required) if they were not supplied by the original checklist provider, and warn users they have been generated rather than supplied by the provider.
- It will permit the transfer of lists of names, in other words names some or all of which are not assigned to any taxon.
- It will permit the inclusion of metadata about the checklist, including information on its source, the type of internal and external identifiers present, etc.

Recommendations

Here we summarise the main recommendations. Some of their advantages and disadvantages and further notes are explored further in the following "Discussion" section.

File structure

Checklists may be exchanged as Darwin Core Archive zip files. The DwCA format specifies a "zip" compressed archive file; it is recommended that the file name extension should be ".zip". It should contain, at least, a core taxon table as a text file, which if other files are present should be named "taxa.txt". This file should include a header row (described as an optional record in [7]) naming all the fields used in the table. Other possible tables are not discussed in this document.

Fields required

The information given for the core taxon table in [6] implies that all fields are optional, but in our application some fields are essential, and these have been identified in the discussion of the various issues below, and are listed in the field tables in Appendix I.

In the i4Life project, because of the current lack of interpretation of the archive descriptor file, if present, support for variations in the columns used in the core taxon table is limited. The first record of the core taxon file in the CoL Download format is currently used to specify which fields are present and in what order. This approach requires the recommended field names to be used so that the fields can be identified. However there is an intention within the i4Life project to use the archive descriptor file in the medium term.

> *Note: The archive descriptor "meta.xml" file can be used to specify variations in the format, but this should not be relied on at present as it requires some programming support and is currently ignored by some import software. Fields from Darwin Core and Dublin Core profiles other than those listed below may be present, but will be ignored. Tools for reading DwCA files exist which might help in developing more flexible import software.*

Character set

Unicode (UTF-8) text files should be used, if possible without a "Beginning of Message" (BOM) sequence.

> *Note for importers: If a BOM sequence is present and it is not automatically swallowed by your file reading software, it may appear as three unexpected characters.*
>
> *Note: The reference [7] discusses the character set used in the core taxon file. It is highly recommended to declare the character encoding used for each file in the meta.xml descriptor.*

Field separator

We propose that tab characters be used to separate fields in the core taxon file.

> *Note: A tab character (as used in the CoL Download Service) is more useful than the common alternative use of a comma character (as in the description referred to in [10] and in the examples in [7]), as the latter requires that strings which may contain a comma (such as authority strings) be enclosed in string quotation characters. It also permits us to recommend the use of a comma character as a separator for lists inside fields, e.g. 'higherClassification'.*
>
> *Note: Any sequence of tab or newline characters present in a data value should be replaced by a single space. If a tab character is accidentally included in a field value it may not be visible to a human editor and will cause errors on import, but the GBIF validator [13] can be used to test for this.*

Note: Since some import software may interpret quote characters (") as enclosing a text string (and remove pairs of them, possibly ignoring any field separators between them), any such characters need to be removed or escaped in some way.

Record separator

Both sources [10] and [7] imply that records are separated by the operating system-specific record terminator. This is acceptable as most software which reads data files will accept Windows, Unix and Macintosh record terminators (which may be a carriage return, a line feed, or both). A Unix-style line-feed character is recommended, but it may be an issue with certain simple text editors such as Windows Notepad (use WordPad instead).

Synonyms

The field 'acceptedNameUsageID' should be used to link a synonym record to its corresponding accepted name (which will have a matching 'taxonID' value). A 'taxonID' value may be any string, it is not required to be numeric. An accepted name should have a unique 'taxonID' value and an empty 'acceptedNameUsageID' field. A synonym (or similar name linked to a taxon) should have a value in the 'acceptedNameUsageID' field which is the value in the 'taxonID' field of the accepted name record to which it is linked. A synonym should also ideally have a unique identifier in the 'taxonID' field (so that records in other optional files, such as bibliographic references, may be linked to it), but the field may be left empty if synonym identifiers are not available.

Note: An alternative field name is 'id' for what is referred to in this document as 'taxonID', since in the application described here the value is not necessarily always a taxon identifier: it could be a name identifier in the case of a synonym record. In practice import software could easily recognise either alternative name.

Note: If a provider has no taxon identifiers, they can be generated by the provider for use within the checklist (in which case the metadata should indicate this, and they may not available for, say, retrieving data about this taxon from the provider, but the 'source' field is available for this purpose anyway). If an identifier has to be generated, it might for example be an arbitrary integer, or a string constructed from the scientific name in a way known to be unique. For example, it could be the scientific name or concatenated binomial or trinomial name, if the checklist does not contain duplicate accepted names. (A non-alphabetic character is needed to separate the concatenated elements, otherwise confusion could occur.) This suggestion is about generating an identifier from the accepted name, which might just happen to have the same string value as the accepted name. This is not the same as using the accepted name itself as the linking value. According to the DwCA documentation, the field 'acceptedNameUsage' may be used to contain the full accepted name of the taxon, but this is strongly discouraged as it raises many issues about how that name should be represented.

A field 'taxonomicStatus' should contain a string representing the status of the name in the record as an accepted name, synonym, etc. At a minimum this should contain either "accepted" or "synonym", normally consistent with the 'acceptedNameUsageID' field being empty or filled (but see the second note below).

Note: The values available for use in the 'taxonomicStatus' field need further discussion. The values "accepted", "provisional", "synonym", "doubtful", "ambiguous", "proparte", "misapplied", "homonym" and "misclassified" are suggested as a starting point. "unknown" has also been suggested, but often an empty value is expected to indicate an unknown value. We need to discuss the use of the GBIF/GNA vocabulary [14].

Note: An example of a misclassified taxon (said to be a common problem in fungi and micro-

organisms) is a binomial which appears to belong to a genus included in the checklist, but has actually been reclassified into a completely different taxon. If this latter taxon is not included in the taxonomic scope of the checklist, the incorrectly classified taxon name needs to be listed for accurate data integration and to prevent the name being repeatedly considered for inclusion. Note that an incorrectly classified taxon name does not refer to an accepted taxon within the checklist, so its 'acceptedNameUsageID' field would be empty. It would presumably also appear as a normal synonym in some other checklist, so this is not an issue for "all taxon" checklists of all organisms.

Scientific names

(a) The 'scientificName' field should be used for the entire name for a taxon, including the authority if available. The taxon may be at any rank, so the name may be a uninomial, binomial, trinomial, polynomial or complex hybrid name. The rank of the taxon concept denoted by the name is placed in the field 'taxonRank'. (If a synonym, this is not necessarily the same as the rank of the accepted taxon name to which it refers.) If the authority is known and can be separated from the rest of the scientific name, the authority string should also be placed in the 'scientificNameAuthorship' field.

Note: The reason for including the authority in the 'scientificName' field is that it is not necessarily the last element of a scientific name (e.g. botanical autonyms, hybrids), so keeping it separate and then concatenating it with the "Latin" part of the name is not reliable.

(b) If possible, the elements of the scientific name for species and lower taxa should also be given in the separate fields provided for each element ('genus', 'subgenus', 'specificEpithet' and 'infraspecificEpithet', in addition to the use of the 'scientificName' field. If the authority cannot be separated from the last element of the scientific name (in the 'specificEpithet' or 'infraspecificEpithet' for binomials and trinomials respectively or in the 'scientificName' field) it can remain appended to that field.

Note: It is difficult to judge which method (a) or (b) is more likely to lead to accurate name transfer, especially for display purposes. Option (a) will handle most cases but display of the names so transferred with italic font in appropriate places will be unreliable. Option (b) may result in some inaccuracies when importing some taxa with unusual names, but could be used to validate the correct application of display fonts. Similarly it is difficult to predict which method will be easier or more reliable for software to import. Hence the suggestion that both versions be provided by producers if they have atomised names in their database or a reliable algorithm for atomising their names. However, there is a possibility that records might be created in which the two methods are inconsistent.

Note: An issue which needs further discussion is whether the 'genus' field for a synonym should contain the generic name of the synonym or of the accepted name to which it refers.

(c) Scientific names of genera, which are composed of a single element, should appear in both the 'scientificName' field and in the 'genus' field. This is compatible with both atomised and concatenated names at the species level and below.

Note: The practice for subgenera needs further discussion. The DwCA documentation is confusing and unhelpful on this point.

See Appendix I for more details of the fields required.

Taxon ranks

The values kingdom, phylum, class, order, family, genus, species, subspecies, variety and form are suggested as a starting point, together with the appropriate super-, sub- and infra-forms of higher ranks.

Note: The correct values for DwC are the usual higher taxa plus "species", "subspecies", "variety" or "form"; additional less frequently used ranks are specified in [15] and they can also be expressed in Latin(!) Unfortunately this permitted variation is not convenient when trying to import checklists, so agreement is needed on the rank names which should be used, especially at infra-specific levels. See also the GBIF rank vocabulary [16].

Note: The values available for use in the 'taxonRank' field need further discussion, especially for infra-specific levels. If it is decided to recommend a particular set of standard values for the 'taxonRank' field, any values not in this set could be represented by storing the original values in the 'verbatimTaxonRank' field.

For synonyms, the field 'taxonRank' (and 'verbatimTaxonRank' if used) should describe the rank of the name, that is the taxonomic rank of the taxon for which it was published, not the rank of the accepted taxon concept to which it is applied as a synonym in the checklist.

Taxonomic hierarchy

(a) All taxa within the scope of the checklist should if possible be included as separate records, including higher taxa. The field 'parentNameUsageID' of the accepted name record for a taxon is used to refer to the 'taxonID' value of the parent taxon at the next higher taxonomic rank included in the checklist.

Note: In this way a tree can be specified, where higher taxa such as genera, families, etc. have their own records in the core taxon file; it is insufficient to use the additional columns provided for 'kingdom' etc. in records for species as these columns provide no way to specify identifiers and other information for higher taxa, and do not permit the inclusion of higher taxa at ranks not recognised by the CoL. If there is no parent included in the checklist, because the "top of the tree" has been reached, then this field should be empty to indicate this.

As with species and infra-specific taxa, a higher taxon may or may not have additional records for any names considered to be synonyms of the taxon. The tree should only include taxa which actually exist and have names and identifiers: if a rank is missing from the hierarchy for a particular taxon, there should be no record for a so-called "unassigned" taxon as this creates ambiguities.

Note: The field 'parentNameUsageID' of a synonym record will typically be ignored and therefore should normally be empty; the 'acceptedNameUsageID' field is used to link the synonym to the taxon to which it refers.

In addition, appropriate higher taxon names should be placed in the relevant columns from the set used to provide the "spreadsheet style" of hierarchy representation, especially if the checklist is to be supplied to the Piping Tools. These fields are 'kingdom', 'phylum', 'class', 'order', 'family' and 'genus'. The fields 'superfamily' and 'subgenus' are also allowed (although not used in the Piping process) as they are used in the CoL Download Service. These should of course be consistent with the higher taxon records, if provided, and if a rank is missing from the hierarchy for a particular taxon, the corresponding field for this "unassigned" rank should be left empty.

Note: The field 'superfamily' is not listed in [9] and there may be support for an official request for it to be added to the DwC standard. Some users may require additional taxonomic rank values. These do not cause any problem when used in the 'taxonRank' field, as described above, or in the 'higherClassification' field for the Piping Tools, but there is an issue if additional columns are used to hold the names of higher taxa at additional ranks, such as 'suborder' or 'superkingdom'. Inserting new fields is not encouraged as the practice may complicate importing checklists (although such fields can be ignored) and it would cause a core taxon file expressed in or converted to XML to fail an XML validation test.

Tracking updates

When re-importing updated checklists, where an earlier version has already been imported, it may be desirable for some checklist importers to know which parts of the checklist have changed. This is certainly the case for the cross-mapping tools, because re-importing and repeat cross-mapping of a checklist is an expensive process. Thus a "datestamp" is needed.

> *Note: It has been suggested that the date of last modification of the checklist file could be used, but as this is not a permanent part of the text file contents, it is easily lost as files are transferred.*

It is recommended that the 'modified' field in the core taxon table be used, since this can provide a separate datestamp for each name. (This is a Dublin Core data type, with string values in the format "2012-07-18": this format allows dates to be compared and sorted correctly.)

> *Note: Initially, providers could simply insert the date of creation of the file for all records, as this would have some use, but ideally the date of last update of each individual record should be used. We need to decide what constitutes a change to a name or taxon. For example, does it include a change to a common name or distribution in another table linked to the taxon?*

Discussion

The following sections provide further observations on some of the aspects where the DwCA profile requires further clarification and agreement, in accordance with the principle expressed in [1].

Where there are two (or more) alternatives for the issues discussed above, we need to decide how to proceed. Having two (or more) different formats would limit the possibilities for exchanging data. Within a single data format, we can distinguish two different possibilities, either

- both alternatives must be provided (which is good for importers, but bad for producers), or
- the choice is optional – either alternative could be provided (which is good for producers who already provide one of the alternatives, but bad for importers, such as the CoL GSD Piping Service, which may not be in a position to handle all the options)

The consensus view is that where two alternative ways to provide the same information are available, as with scientific names and hierarchies, checklist exporters should be encouraged to provide both alternatives. Including duplicated data values in different formats should ensure the format's wide usefulness to consumers with varying import needs and software. However, it is explicitly acknowledged that for certain specialised purposes, such as the transfer of new taxa and names to the Piping Service, some of the alternatives may not be required and some restrictions may exist, as discussed elsewhere in this document.

Representation of scientific names

Scientific names, of course, are composed of a number of elements followed by an author string which is sometimes omitted. The process of parsing such strings into their component parts, necessary for accurate comparison and presentation, is complex and may sometimes be incomplete and therefore liable to introduce errors. So it is desirable to present scientific names in a fully "atomised" form. But some checklist producers may not have their names stored this way.

Format	Advantages	Disadvantages
Names stored ***concatenated in one field*** including the authority string	• Can handle complex names such as trinomials and quadrinomials	• Representation may be inconsistent (leading to matching difficulties) • Interpretation may be difficult and unreliable (also leading to matching difficulties) • Identifying author strings may be unreliable • Display with italic font where appropriate is difficult
Names stored in ***two strings*** (the ***concatenated*** Latin scientific name part and the authority string in separate fields)	• Can handle complex names such as trinomials and quadrinomials • The author string is easily recognised	• Representation of the scientific name part may be inconsistent for infra-specific taxa (leading to difficulties in matching and display with italic font) • Interpretation may be difficult and unreliable (also leading to matching and italic font difficulties)
Fully ***atomised*** names (with genus, specific epithet, any infra-specific epithets and the authority string in separate fields)	• No ambiguities • Display with italic font etc. is easy • Matching can be controlled better	• May be difficult for some providers to produce • Might not handle quadrinomials (depending on the schema chosen) • May be difficult to reconstruct an entire correctly formatted name in complex cases

Note: Hybrid formulae (for example, Hypoxis parvula *Baker var.* albiflora *B.L. Burtt* × Rhodohypoxis baurii *(Baker) Nel) and chimaeric taxa pose special difficulties in all alternatives, and may require compromises to be accepted. They can be handled when names are stored in one or two strings, but interpretation for matching and display with italic font will be difficult. Even the fully atomised name model cannot handle them unless a very complex schema is used. The Download Service specification embeds the multiplication symbol in the epithet string, which is a workable compromise.*

Linkage of synonyms to accepted names of taxa

There is more than one way to link synonyms to their accepted names. The 'acceptedNameUsageID' field is intended to be used to link a synonym record to its corresponding accepted name (the record with a matching 'taxonID' value), but its usage has varied in past practice. In the case of records for accepted names, the 'acceptedNameUsageID' field is recommended to be empty, although it may contain a duplicate of the 'taxonID' value, but in either case this indicates that the record is for an accepted name. Similarly, in the case of a synonym record in different example files, the 'taxonID' field may be empty or contain the identifier of this name record or contain the identifier of the matching accepted name record; in the latter cases a separate field ('taxonomicStatus' or 'acceptedNameUsageID' or preferably both) needs to indicate that this is a synonym.

However, the use of these fields, whilst sufficient for certain checklists, is insufficient to convey more than the distinction between accepted names and synonyms. The status of

accepted names may be certain or "provisional" and synonyms may actually have a variety of statuses, including further information about the type of synonymy (doubtful, ambiguous, *pro parte*, etc.) or the names of taxa which need to be distinguished from the present taxon but may be confused with it (such as misapplied names, homonyms and incorrectly classified taxa). Therefore in many cases a separate field 'taxonomicStatus' is required to provide this status information.

> *Note: CoL practice is to provide more than one record for a* pro parte *synonym, so that each can refer to a different accepted taxon.*

Representation of the taxonomic hierarchy

Three alternative representations of a taxonomic hierarchy are commonly used:

1. A model which GBIF [8] calls "**database-style**" (also known in the database literature as an "adjacency list") in which each node (or entity, a taxon in the case of a taxonomic hierarchy) is stored in a separate record, each with a link to its "parent" (next higher taxon in the hierarchy).
2. A model which GBIF calls "**spreadsheet-style**" is used in the Download Service format and needed for the Piping Tools, in which higher taxa are not treated as entities in their own right, but instead selected higher taxon ranks are treated as attributes of species. So rather than being stored in separate records, they are given in a number of additional columns, one column per selected rank.
3. A variant of the above is a "path enumeration" in which a single field contains the names or identifiers of all higher level taxa. This has many disadvantages for general use, but the DwCA 'higherClassification' field can be used as a "path enumeration" field in files intended for the piping tool to convey arbitrary levels of classification.

The advantages and disadvantages of the first two alternatives can be summarised:

Format	**Advantages**	**Disadvantages**
Higher taxon names stored in separate records ("***database style***")	• Can store all taxonomic ranks • Can be designed to handle synonyms and alternative hierarchies • Easily handles missing higher taxa	• May require potentially more complex or time-consuming retrieval or conversion procedures for certain purposes
Higher taxon names stored in the same record in separate columns ("***spreadsheet style***")	• Treats higher ranks as attributes of species, useful if, as with CoL checklist assembly, processing is species- rather than hierarchy-oriented	• No identifiers for higher taxa • Cannot handle synonyms of higher taxa or alternative hierarchies • No guarantee of consistency because data is duplicated (see note below) • Loses names of higher taxa which do not correspond to the chosen columns • Difficult for importers with optimised data schemas

> *Note: Both formats may be difficult for some providers to produce, depending on how they store*

their data; in the spreadsheet-style variant, inconsistency is possible in which a single higher taxon may be named in multiple records for lower taxa, where it may have different parents. It is also unclear how to link infra-specific taxa to their "parent" species in this model.

Metadata

It will be seen from the foregoing that at various points decisions have to be made by checklist providers concerning how they will present their checklist information in the resulting exported files. There is also other information about a checklist which a consumer would like to know. Some of this information or metadata about the checklist might be conveyed separately by human interaction using email, telephone, etc., but ideally it should be captured within the checklist file itself, so that it is not lost as the file is distributed. This topic is not discussed further in this document, but possibly the archive descriptor file in DwCA or the "Bisby Core" approach might be used. Possible metadata includes:

- which optional fields and other checklist features are present,
- whether features have been generated automatically, or found to be inconsistent (for example by an automatic consistency checker),
- whether identifiers have been generated during processing (i.e. after the checklist was supplied by the original provider),
- properties of the data as a whole, including the taxonomic and geographical scope of the checklist, its author, date etc.,
- aspects of the IPR relating to the data.

Notes on checklist testing

We recommend the use of appropriate tools to perform checklist consistency checking to identify any problems before uploading a DwCA file to the cross-mapping service or sending it to any other recipient, especially if new software or significantly different data has been used in the preparation of the file. This would allow checklist producers to test their output and it would assist checklist importers to investigate the reasons for any failure to import.

GBIF provides a test archive [17] and a very informative online service to validate a Darwin Core Archive [13]. However, this is only available for small files (< 20Mb), and some of the issues reported by the validation service can be disregarded, as it may flag as errors certain features (such as superfamily) which our standard permits. Conversely, some archives which pass all the GBIF validation tests may fail to be read correctly by import software because they do not follow the recommendations in this proposal. However, the GBIF tool does provide much useful information. An example of its output is provided [18].

A more specific checklist consistency checking tool should be implemented, perhaps by adapting the GBIF tool or an import program. The OpenCSV project provides checking tools.

Note: GBIF has developed highly tested basic Java libraries to read any Darwin Core Archives. Refer to the dwca-reader project in particular, which is used in GBIF software for reading and writing DwC archives. It does all the meta.xml parsing and provides a simple interface to DwC archives.

References

[1] "Simple Darwin Core" http://rs.tdwg.org/dwc/2009-12-07/terms/simple/index.htm

[2] Viktoras Didziulis, Kwok Yin Cheung, Magda Sitko, David Remsen, Ruud Altenburg & Wouter Addink (12 Sept 2011) Download Service specification: Download (version 3) and Piping Tools (version 2) Specifications, Deliverable 2.1 – updated version, Workpackages 2, 4, 6, 12

[3] Remsen D., Döring M., Robertson, T., Ko, B. (2011). Best Practices in Publishing Species Checklists, Copenhagen: Global Biodiversity Information Facility, 10 pp, accessible online at http://links.gbif.org/checklist_how_to

[4] Wieczorek J, Döring M, De Giovanni R, Robertson T, Vieglais D (2009) Darwin Core. Available: http://www.tdwg.org/standards/450/

[5] Wieczorek J, Bloom D, Guralnick R, Blum S, Döring M, et al. (2012) Darwin Core: An Evolving Community-Developed Biodiversity Data Standard. PLoS ONE 7(1): e29715. doi:10.1371/journal.pone.0029715

[6] http://code.google.com/p/gbif-ecat/wiki/DwCArchive

[7] http://rs.tdwg.org/dwc/terms/guides/text/index.htm

[8] GBIF (2010). GBIF GNA Profile Reference Guide for Darwin Core Archives, version 1.2, released on 1 April 2011, (contributed by Remsen D.P., Döring, M, Robertson, T.), Copenhagen: Global Biodiversity Information Facility, 28 pp,. ISBN: 87-92020-25-9 Accessible at http://links.gbif.org/gbif_gna_profile_reference_guide or at http://www.gbif.org/orc/?doc_id=2822

[9] http://rs.gbif.org/core/dwc_taxon.xml

[10] i4Life-project deliverable report "D 2.1v3.2.pdf" (available from www.i4life.eu) describes the format which the Catalogue of Life Download Service uses.

[11] http://dev.e-taxonomy.eu/trac/wiki/CoL2EDITPipeline

[12] "Structure of e-2 DwC-A file.pdf" and two example DwCA files in the 4D4Life project Subversion repository

[13] Online service to validate a Darwin Core Archive: http://tools.gbif.org/dwca-validator/

[14] http://rs.gbif.org/vocabulary/gbif/taxonomic_status.xml

[15] http://code.google.com/p/darwincore/wiki/Taxon#taxonRank

[16] http://rs.gbif.org/vocabulary/gbif/rank.xml

[17] GBIF test archive: http://darwincore.googlecode.com/svn/trunk/dwca-reader/src/test/resources/archive-tax.zip

[18] http://litchi.cs.cf.ac.uk/resources/tools.gbif.org/dwca-reports/116-1850942067830013310.html

Appendix I: Recommended fields and their names

The following tables list the Darwin Core and Dublin Core fields available for the core taxon ("taxa.txt") table. Three tables are given, for fields which are ***required***, ***optional***, and ***ignored*** (for the purposes of checklist import) respectively.

Required fields

These are fields which are essential or strongly desired. "desired" refers to fields which should be provided if possible, although some of them may be omitted in certain circumstances for limited purposes (for example, columns for higher taxon ranks not present in the checklist). Certain "essential" fields may be omitted if they have no use in checklists for limited purposes (for example 'acceptedNameUsageID' if no synonyms are present and 'parentNameUsageID' if a hierarchy is lacking).

Field name	Required	Existing usage
taxonID (or id)	essential	CoL Download Service, Piping tool input (compulsory)
acceptedNameUsageID	essential	CoL Download Service, Piping tool input (compulsory for synonyms)
parentNameUsageID	essential	CoL Download Service, Piping tool input (compulsory for infra-specific taxa)
scientificName	essential	CoL Download Service (with authorship if available)
scientificNameAuthorship	desired	CoL Download Service, Piping tool input (compulsory where available)
kingdom	see note 1	CoL Download Service, Piping tool input (compulsory where appropriate)
phylum	see note 1	CoL Download Service, Piping tool input (compulsory where appropriate)
class	see note 1	CoL Download Service, Piping tool input (compulsory where appropriate)
order	see note 1	CoL Download Service, Piping tool input (compulsory where appropriate)
superfamily	desired	CoL Download Service (not in DwC)
family	see note 1	CoL Download Service, Piping tool input (compulsory where appropriate)
genus	see note 1	CoL Download Service, Piping tool input (compulsory)
subgenus	desired	CoL Download Service
specificEpithet	see note 1	CoL Download Service, Piping tool input (compulsory)
infraspecificEpithet	see note 1	CoL Download Service, Piping tool input (compulsory where available)
taxonRank	essential	CoL Download Service
taxonomicStatus	essential	CoL Download Service, Piping tool input (compulsory but restricted to: accepted, synonym, unknown)
modified	desired	CoL Download Service (but should use the Dublin Core format)

Note 1: This field is required in checklists prepared for the Piping Tools, if the rank is represented in the data set.

Optional fields

These are fields which may be useful when interpreting checklists or for use by importing software, but may be omitted if they cannot easily be provided with data values. Some of these fields marked "metadata" may be used to associate metadata information with the checklist for other purposes, for example for IPR.

> *Note: Metadata fields would not be useful or appropriate in the core taxon table if their values are the same for all records. In that case the information, if required, should be provided as a "default attribute" in the DwC archive descriptor table. In "all taxon" checklists such as the CoL, on the other hand, these values might differ between constituent GSDs in the same core taxon file, and thus merit inclusion. This argument can also be applied to justify omitting the 'kingdom', 'phylum', etc. where these ranks are invariant in checklists of taxonomic subsets.*

Field name	**Required**	**Existing usage (or GBIF explanation)**
nameAccordingTo		CoL Download Service
namePublishedIn		CoL Download Service, Piping tool input (compulsory where appropriate)
scientificNameID		CoL Download Service (for ITIS TSNs)
verbatimTaxonRank		CoL Download Service, Piping tool input (compulsory where appropriate)
taxonRemarks		CoL Download Service, Piping tool input (compulsory where appropriate)
source		CoL Download Service, Piping tool input (compulsory where appropriate)
taxonConceptID		CoL Download Service (as 'identifier'), Piping tool input (as 'taxon id', compulsory where appropriate)
language	metadata	A language of the resource. Recommended best practice is to use a controlled vocabulary such as RFC 4646.
rights	metadata	Information about rights held in and over the resource. Typically, rights information includes a statement about various property rights associated with the resource, including intellectual property rights.
rightsHolder	metadata	A person or organisation owning or managing rights over the resource.
accessRights	metadata	Information about who can access the resource or an indication of its security status. accessRights may include information regarding access or restrictions based on privacy, security, or other policies.
datasetID	metadata	CoL Download Service
datasetName	metadata	CoL Download Service
higherClassification	see note 1	Piping tool input: provides GSDs with information about the taxonomic hierarchy used in the original data source

Fields to be ignored

These fields are defined as available for use in the core taxon file as part of the DwCA standard, but have no planned use for checklist exchange in the i4Life project. If present, they will be ignored by importing software.

Note: Fields marked "don't use" are especially deprecated for our purposes, because the use of alternative fields which use identifiers is preferable to the use of names which take more space and are slower and less reliable to match.

Field name	Use	GBIF explanation
acceptedNameUsage	don't use	The scientificName of the taxon considered to be the valid (zoological) or accepted (botanical) name for this nameUsage.
parentNameUsage	don't use	The scientificName representing the direct, most proximate higher-rank parent taxon (in a taxonomic classification) of this taxon's scientificName.
originalNameUsageID		A unique identifier for the nameUsage instance in which the name was originally established, under the rules of the associated nomenclaturalCode (i.e., within the namePublishedIn reference). The basionym (botany) or basonym (bacteriology) of the scientificName or the senior/earlier homonym for replaced names. If provided the nameAccordingTo value returned should match the namePublishedIn value for this record.
originalNameUsage	don't use	The equivalent of the scientificName as it originally appeared when the name was first established under the rules of the associated nomenclaturalCode (i.e., within the namePublishedIn reference). The basionym (botany) or basonym (bacteriology) of the scientificName or the senior/earlier homonym for replaced names.
nameAccordingToID		A unique identifier that returns the details of a nameAccordingTo reference.
namePublishedInID		A preferably resolvable, globally unique identifier that refers to namePublishedIn.
namePublishedInYear		(not in GNA profile or in proposal)
vernacularName		A common or vernacular name.
nomenclaturalCode		The nomenclatural code under which the scientificName is constructed.
nomenclaturalStatus		The status related to the original publication of the name and its conformance to the relevant rules of nomenclature.
bibliographicCitation		Citation information specified by the data publisher. Citation information is inherited downward by all child taxa if no other citation is included.
informationWithheld		Additional remarks as to information not published, but available.