



Building a knowledge-base

Documentation for the FSC Identikit



Dr Richard Burkmar
BioLinks Digital Development Officer
Field Studies Council
Head Office
Montford Bridge
Shrewsbury
SY4 1HW

r.burkmar@field-studies-council.org

Development funded by the Esmée
Fairbairn Foundation and the
Heritage Lottery Fund



1 Contents

1	Contents.....	2
2	Introduction	4
3	Building a knowledge-base	4
3.1	The taxa worksheet.....	5
3.1.1	General rules for the taxa worksheet	5
3.1.2	Text character state values	5
3.1.3	Ordinal character state values	6
3.1.4	Special character state values.....	7
3.1.5	Describing taxonomy in a knowledge-base	8
3.1.6	NBN Mapping.....	9
3.2	The characters worksheet.....	10
3.2.1	The Group column	10
3.2.2	The Label column	11
3.2.3	The Help and HelpShort columns	11
3.2.4	The Status column.....	11
3.2.5	The ValueType column.....	12
3.2.6	The ControlType and Params columns	13
3.2.7	The Weight column.....	13
3.2.8	The Strictness column	14
3.2.9	The Latitude column	15
3.3	The values worksheet	17
3.3.1	Providing help on character state values.....	19
3.3.2	Translating character state values	19
3.3.3	Ranking ordinal character state values	20
3.3.4	Specifying order for text character values in drop-down lists	20
3.3.5	Specifying state groups	21
3.4	The media worksheet	21
3.4.1	Images to supplement help text	22
3.4.2	Images to illustrate taxa.....	23
3.4.3	The UseFor and TipStyle columns	24
3.4.4	HTML files to provide further information on taxa	25
3.4.5	General comments on the media worksheet	26



3.4.6	Checking media files	26
3.5	The config worksheet.....	26
3.5.1	Identikit configuration (type 'config')	27
3.5.2	Knowledge-base metadata	28
3.5.3	Release history.....	29
3.6	Including an information file about your knowledge-base.....	29
3.7	Dealing with sexual dimorphism.....	30
3.8	The macros worksheet.....	31
3.9	Using other spreadsheet features	32

2 Introduction

This document is a reference for building knowledge-bases for use with the FSC Identikit (referred to in this document as ‘the Identikit’). Most of the examples referred to are from the example ‘biscuits’ knowledge-base which is supplied with the Identikit (‘biscuits.xlsx’). We recommend that, before you start building your own knowledge-base, you work through this documentation using the ‘biscuits’ example. This documentation will make most sense to you if you have the biscuit knowledge base open and refer to it as you go.

3 Building a knowledge-base

The Identikit uses a spreadsheet to represent taxonomic knowledge. The example spreadsheet provided with the Identikit is an Excel workbook, but you can use any type of spreadsheet you like to build a knowledge-base that can be used by the Identikit – the only requirement is that it can save to CSV (Comma Separated Value) files – since this is what the Identikit actually reads.



Each knowledge-base is represented by five CSV files with the following names:

- taxa.csv
- characters.csv
- values.csv
- media.csv
- config.csv



In the example Excel biscuits workbook, there is a separate worksheet for each of these. A sixth worksheet – called ‘macros’ – has a button on it which invokes a macro to save the current workbook and export each of the five worksheets – taxa, characters, values, media and config – to the CSV files that are read by the Identikit. This makes it easy to generate all the CSVs required by the Identikit at the push of a button (but you can save each of them manually if you don’t want to use the macro).

Further sections in this document deal with each of the five CSV files (which will be referred to here as ‘worksheets’).

On all worksheets, placing a hash character (‘#’) as the first character of the first column in a row effectively ‘comments out’ the whole row – it won’t be read by the Identikit. This is a useful mechanism for knowledge-base authors to provide notes on their knowledge-bases or to make rows invisible to the Identikit without removing them from the knowledge-base.

Note that, in general, text in the knowledge-base is treated case-sensitively by the Identikit, so, for example, ‘Shape’ is not the same as ‘shape’. So be sure to have regard for case when creating knowledge-bases.

The sections below tell you how to construct a knowledge-base. It’s easy to make mistakes but, fear not, when you run the Identikit, it will check the integrity of your knowledge-base and indicate problems to you. This is documented in the separate document ‘Deploying your visualisations’.



3.1 The taxa worksheet

The example below shows the taxa worksheet from the biscuits knowledge-base.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Taxon	Shape	BiscuitColour	Coating	Holes	Pattern	Words	SingleDouble	FillingColour	Width	Length	ChocTaste	Availability	TopTen
2	Rich Highland Shortie	Round	Pale brown	None	Yes	Yes	No	Single	n/a	57	57	none	[jan-dec]	no
3	Romany Vanilla Cream	Round	Dark brown	None	No	Yes	No	Double	White	47	47	none	[jan-dec]	no
4	Shortcake	Rectangular	Pale brown	None	Yes	Yes	No	Single	n/a	37	61	none	[jan-dec]	no
5	Custard Cream	Rectangular	Pale brown	None	Yes	Yes	Yes	Double	White	34	47	none	[jan-dec]	yes
6	Bourbon Cream	Rectangular	Dark brown	None	Yes	No	Yes	Double	Dark brown	30	62	little	[jan-dec]	yes
7	Happy Face	Round	Pale brown	None	No	Yes	No	Double	White	47	47	none	[jan-dec]	no
8	Milk Chocolate Digestive	Round	Pale brown	Chocolate	No	No	No	Single	n/a	58	58	very	[jan-dec]	no
9	Milk Chocolate Finger	Cigar	Pale brown	Chocolate	No	No	No	Single	n/a	15	78	very	[jan-dec]	yes
10	Jam Sandwich Cream	Round	Pale brown	None	No	Yes	No	Double	White and red	47	47	none	[jan-dec]	no
11	Choc Chip Cookie	Round	Pale brown	None	No	No	No	Single	n/a	57	57	moderate	[jan-dec]	no
12	#Rich's fantasy biscuit 1	Triangle	Dark brown Pale brown	Toffee	Yes	No	No	Single	n/a	[80-90]	[90-100]	very	[jan-dec]	no
13	Rich's fantasy biscuit 2	Hexagonal	Pink Dark brown Pale brown	Chocolate	Yes	No	?	Double	Dark brown	[80-90]	[90-100]	[little-very]	[may-jun]	no
14	Rich's fantasy biscuit 3	Round	Pale brown	Chocolate	No	No	No	Double	n/a	50	50	very moderate	dec	no
15	Rich's fantasy biscuit 4	Cigar	Dark brown	Chocolate	Yes	Yes	No	Single	n/a	10	70	very [none-little]	[dec-feb]	no

3.1.1 General rules for the taxa worksheet

These are the general rules to bear in mind when specifying knowledge on the taxa worksheet:

- There *must* be a column called 'Taxon'. This column holds the names of the taxa as they will appear to users of the visualisations.
- The order of the columns is unimportant (and the 'Taxon' column needn't be the first column).
- Further columns represent attributes of the taxa – e.g. morphological characters.
- No spaces are allowed in the column names.
- The column names are not what appears to users (see section on character worksheet) so you only need to use names that are meaningful to you as the knowledge-base author.
- Apart from the first row (header row), which must contain the column names, each row represents a different taxon.
- A taxon can be included in the knowledge-base but excluded from the visualisations by being 'commented out' with the hash character '#' – as is the case for 'Rich's fantasy biscuit 1' in the biscuits knowledge-base.
- The values in the cells of a morphological character column contain the character state values for each of the taxa.
- Character state values can 'commented out' by using a leading hash character ('#').
- There's no practical limit on the number of taxa (rows) or characters (columns) that can be specified.
- There are two broad types of character state values: text and numeric.

11	Choc Chip Cookie	Round
12	#Rich's fantasy biscuit 1	Triangle
13	Rich's fantasy biscuit 2	Hexagonal

3.1.2 Text character state values

Text values are expressed with normal alphanumeric characters. In fact you can use most keyboard characters in text state values (but they have not all been tested exhaustively). However, one character – '|' – has a reserved meaning. It is interpreted as 'or' by the Identikit so you should only use this to separate multiple alternatives for a given taxon/character. An example in the biscuits knowledge-base is the value for the 'Colour' character and 'Rich's fantasy biscuit 2' taxon which is specified as 'Pink | Dark brown | Pale brown'. That means that this type of biscuit can be either pink, dark brown or light brown and would score as a match for that character if the user specified any one of those three colours with the Identikit's character state input tools. (See also the section on 'special character state values'.)

BiscuitColour
Pale brown
Dark brown Pale brown
Pink Dark brown Pale brown



There is no equivalent 'and' character. So you can't specify, for example, that a biscuit is both pink *and* dark brown in a way that would match a multiple selection by the user (see section on the characters worksheet for an explanation of multiple state selection). Instead you would have to specify 'Pink and dark brown' as the character state – the word 'and' would have no special meaning to the Identikit, but this character state would be considered by the Identikit to be completely different from both 'Pink' and 'Dark brown' and presented to the user as a separate possible value in its own right.

Normally, when constructing a knowledge-base, if you find yourself wishing to use 'and' a lot to combine character states, it is an indication that you might be trying to specify a character which would be better disaggregated into two or more different characters.

Sometimes you may find that you need to supply a lot of alternatives for a value and, furthermore, that you have to specify the same long list of alternatives for several taxa. In such cases, you can use the values worksheet to group values under a 'state group' name and instead of supplying the individual values, specify the state group name instead. An example from the biscuits knowledge-base is the value of 'Availability' for 'Rich's fantasy biscuit 2' which is set the the group 'even-months'. (Actually this is an ordinal character but it works in the same way for both text and ordinal characters.)

Numeric values are expressed in any of the commonly accepted ways (except scientific notation), so any of the following examples are valid: 1000, 1, 1.0, 0.01 and 123.123. It is frequently necessary to provide a numeric range and this is accomplished using the notation '[n1-n2]' where n1 and n2 are numbers specified in a valid format. Examples from the biscuits knowledge-base include the values of the 'Width' and 'Length' characters for 'Rich's fantasy biscuit 2'. That means that the size of 'Rich's fantasy biscuit 2' is variable and will match against any number specified by a user which is within the specified range (unusual for a biscuit but common for biological organisms!).

	A	J	K
1	Taxon	Width	Length
11	Choc Chip Cookie	57	57
12	#Rich's fantasy biscuit 1	[80-90]	[90-100]
13	Rich's fantasy biscuit 2	[80-90]	[90-100]

3.1.3 Ordinal character state values

Ordinal characters are a bit of a hybrid of text and numeric – having some properties of each of them. They look like text values – being comprised of words rather than numbers – but the essential feature of ordinal character values is that they can be ranked – like numbers.

For example imagine a character that can take the following values:

- 'hairless'
- 'few hairs'
- 'moderately hairy'
- 'very hairy'
- 'downy'

These values are comprised of words, but they can be ordered – e.g. 'moderately hairy' is less than 'very hairy' but greater than 'few hairs'. Why is this important? If this character was a 'text' type, then a value of 'few hairs' specified by a user would not match 'downy' taxa or taxa that are



‘moderately hairy’. But in reality these types of character states are hard to judge and they can often also be variable in nature; under these circumstances, you might want taxa that are ‘moderately hairy’ to score *something* when the user enters ‘few hairs’ and, furthermore, to score more than taxa that are ‘downy’. In such cases, you specify characters as type ‘ordinal’ (which will be described in the section on the ‘characters worksheet’).

The way you can specify ordinal character values on the taxa worksheet has something in common with text characters and something with numeric characters. Like text characters, values are just specified by entering their text values. Alternatives can be specified by using the ‘or’ character – ‘|’ – as with text characters, but because they are ranked, it is normally better to specify an ordinal range, in a similar fashion to numeric characters. For example if you wanted to specify a hairiness range for a certain taxon, you could specify it thus: ‘[hairless-moderately hairy]’.

Just like text characters, you can use a state group value to conveniently indicate a large number of alternative states. You cannot use both an ordinal range and a state group for a single taxon character state.

An example from the biscuits knowledge-base is shown here. (Note however, that although the ‘or’ character has been demonstrated, there are probably few realistic examples where you’d need to use this – normally a range will suffice.)

ChocTaste
none
none
none
none
little
none
very
very
none
moderate
very
[little-very]
very moderate
very [none-little]

3.1.4 Special character state values

As well as the text and numeric state values indicated above, there are three other ‘special’ values that can be used in the character/taxon character state spreadsheet cells. These are:

- “
- ‘?’
- ‘n/a’
- novalue

The first one is an empty cell, i.e. no value at all. This is allowed in a knowledge-base and is treated as a missing value. Missing values neither score for or against a taxon when matching against user input. Obviously the performance of knowledge-bases with lots of missing values is poorer than those that are well-populated, but it is imperative that visualisations can operate ‘normally’ on knowledge-bases with missing values so that knowledge-base authors can see progress as their knowledge-bases mature.

The second – the question mark – also denotes missing data and is treated in exactly the same way as an empty cell by the Identikit. However it can be used by the knowledge-base author as a marker in the knowledge-base, for instance as a reminder that some work is needed on that character state.

The value ‘n/a’ represents ‘not applicable’. This should be used for characters that are not applicable for a given taxon. For example in the biscuits knowledge-base the value ‘n/a’ is

	A	I
1	Taxon	FillingColour
2	Rich Highland Shortie	n/a
3	Romany Vanilla Cream	White
4	Shortcake	n/a
5	Custard Cream	White
6	Bourbon Cream	Dark brown
7	Happy Face	White
8	Milk Chocolate Digestive	n/a



used for the character 'FillingColour' for those taxa that have a value of 'single' for the character 'SingleDouble', indicating that they do not have fillings. If the user specifies a value for the character 'FillingColour' then those taxa with a value of 'n/a' will score negatively for that character.

The meaning of final one - 'novalue' – is 'not manifest'. This means that none of the possible states for this character are manifest for this taxon. It is an important way to distinguish between empty cells – representing missing values – and those where the character does not manifest itself for the taxon because they score differently. If the user selects a state for a character those taxa with missing values for that character state will score neutrally – i.e. zero - but those for which the character is not manifest score negatively.

This value 'novalue' is most useful for characters which describe highly distinctive features of certain taxa. For example, in the Harvestman knowledge-base there is a character 'Other distinctive trident features' designed to pick out a handful of trident features exhibited by just a few taxa. For all other taxa, these features aren't present. Those taxa which do not have a trident will correctly specify a value of 'n/a' for this character. But those that have a trident which does not exhibit any of the possible character states can use 'novalue'. In fact 'n/a' and 'novalue' score similarly, so what you use is really up to you, but there is a true semantic difference.

3.1.5 Describing taxonomy in a knowledge-base

Some visualisations make use of taxonomic (as opposed to morphological) knowledge expressed in a knowledge base (e.g. the Circle-pack key). So for example you could record, for each taxon in your knowledge base, the genus, sub-family, family and any other higher level taxonomic groupings that you like.

To do that, you add columns to the taxa worksheet in the normal way – each column representing a taxonomic rank – and populate the cells with the value of that rank for each taxon. The example below shows a section from a knowledge-base on harvestmen.

	A	B	C	D
1	Taxon	Family	SubFamily	Genus
2	Nemastoma bimaculatum	Nemastomatidae		Nemastoma
3	Mitostoma chrysomelas	Nemastomatidae		Mitostoma
4	Nemastomella bacillifera	Nemastomatidae		Nemastomella
5	Trogulus tricarinatus	Trogulidae		Trogulus
6	Anelasmacephalus cambridgei	Trogulidae		Anelasmacephalus
7	Sabacon viscayanum ramblaianum	Sabaconidae		Sabacon
8	Homalenotus quadridentatus	Sclerosomatidae	Sclerosomatinae	Homalenotus
9	Oligolophus tridens	Phalangiidae	Oligolophinae	Oligolophus
10	Oligolophus hansenii	Phalangiidae	Oligolophinae	Oligolophus
11	Paroligolophus agrestis	Phalangiidae	Oligolophinae	Paroligolophus
12	Paroligolophus meadii	Phalangiidae	Oligolophinae	Paroligolophus
13	Lacinius ephippiatus	Phalangiidae	Oligolophinae	Lacinius
14	Odiellus spinosus	Phalangiidae	Oligolophinae	Odiellus

The order in which you specify the columns does not matter – the taxonomic order of the ranks is indicated on the characters worksheet. See 'The group column' section for the characters worksheet



to complete your understanding of how to specify taxonomy. Note that the taxonomy you specify must be expressible as a strict hierarchy – so all taxa that have the same value for a given taxonomic rank, must also have the same values at all higher taxonomic ranks. This is a general truism of all hierarchical taxonomies in the tree of life. If you break this rule, you can expect unexpected results!

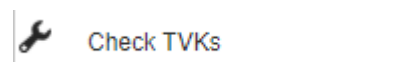
3.1.6 NBN Mapping

From version 1.7.0 of the Identikit, some features (e.g. taxon details dialog) and visualisations (e.g. the ‘full details’ visualisation) are able to show distribution maps for taxa by using NBN web services. To make use of this feature, all you have to do is supply a column on the Taxon worksheet called ‘TVK’ (note sure it’s upper case). TVK stands for ‘taxon version key’ and is a unique identifier, for each taxon, used by the NBN. For every taxon for which you wish to display an NBN distribution map, you must supply its TVK in this column. The example below is from the Harvestman knowledge-base.

	A	B	C	D	E	F	G	H
1	Taxon	Family	SubFamily	Genus	TVK	Niche	Synanthropic	AdultSeason
2	Nemastoma bimaculatum	Nemastomatidae		Nemastoma	NBNSYS0000013844	ground	no	spring summer autumn winter
3	Mitostoma chrysomelas	Nemastomatidae		Mitostoma	NBNSYS0000013845	ground	no	spring summer autumn winter
4	Nemastomella bacillifera	Nemastomatidae		Nemastomella	NHMSYS0001590852	ground	yes	autumn
5	Trogulus tricaratus	Trogulidae		Trogulus	NHMSYS0001590934	ground	no	spring summer autumn winter
6	Anelasmacephalus cambridgei	Trogulidae		Anelasmacephalus	NBNSYS0000013843	ground	no	spring summer autumn winter
7	Sabacon viscayanum ramblaia	Sabaconidae		Sabacon	NBNSYS0000033230	ground	no	summer autumn
8	Homalenotus quadridentatus	Sclerosomatidae		Homalenotus	NBNSYS0000013846	ground	no	spring summer autumn winter
9	Oligolophus tridens	Phalangidae	Oligolophinae	Oligolophus	NHMSYS0001590898	ground herbaceous	no	summer autumn
10	Oligolophus hansenii	Phalangidae	Oligolophinae	Oligolophus	NBNSYS0100004303	shrub wall	no	summer autumn winter
11	Paroligolophus agrestis	Phalangidae	Oligolophinae	Paroligolophus	NBNSYS0000013853	herbaceous shrub wall	yes	summer autumn winter
12	Paroligolophus meadii	Phalangidae	Oligolophinae	Paroligolophus	NHMSYS0001590914	ground	no	summer autumn
13	Lacinius ephippiatus	Phalangidae	Oligolophinae	Lacinius	NHMSYS0001590860	ground herbaceous	no	summer autumn
14	Odiellus spinosus	Phalangidae	Oligolophinae	Odiellus	NBNSYS0000013865	ground	no	summer autumn
15	Mitopus morio var morio	Phalangidae	Oligolophinae	Mitopus	NHMSYS0001590870	herbaceous shrub wall	yes	summer autumn
16	Mitopus morio var ericaeus	Phalangidae	Oligolophinae	Mitopus	NBNSYS0000013849	herbaceous	no	summer autumn
17	Phalangium opilio	Phalangidae	Phalangiinae	Phalangium	NBNSYS0000013856	herbaceous shrub wall	yes	spring summer autumn
18	Opilio parietinus	Phalangidae	Opilioninae	Opilio	NHMSYS0001590904	shrub wall	yes	summer autumn
19	Opilio canestrinii	Phalangidae	Opilioninae	Opilio	NBNSYS0100004330	shrub wall	yes	summer autumn winter
20	Opilio saxatilis	Phalangidae	Opilioninae	Opilio	NHMSYS0001590907	herbaceous	no	summer autumn
21	Megabunus diadema	Phalangidae	Platybuninae	Megabunus	NBNSYS0000013860	shrub wall	no	spring summer autumn winter
22	Platybunus triangularis	Phalangidae	Platybuninae	Platybunus	NHMSYS0001590926	herbaceous	no	spring summer
23	Platybunus pinetorum	Phalangidae	Platybuninae	Platybunus	NHMSYS0020978833	shrub wall	yes	spring summer
24	Lophopilio palpalis	Phalangidae	Platybuninae	Lophopilio	NBNSYS0000013854	ground herbaceous	no	summer autumn winter
25	Dicranopalpus ramosus	Phalangidae	Gyantinae	Dicranopalpus	NHMSYS0021178904	shrub wall	yes	summer autumn winter
26	Dicranopalpus caudatus	Phalangidae	Gyantinae	Dicranopalpus	NHMSYS0021178916	?	no	?
27	Leiobunum rotundum	Phalangidae	Leiobuninae	Leiobunum	NBNSYS0000013861	herbaceous shrub wall	yes	summer autumn
28	Leiobunum blackwalli	Phalangidae	Leiobuninae	Leiobunum	NBNSYS0000013862	herbaceous shrub wall	no	summer autumn winter
29	Leiobunum tisciae	Phalangidae	Leiobuninae	Leiobunum	NBNSYS0000013863	shrub wall	yes	summer autumn
30	Leiobunum sp. A	Phalangidae	Leiobuninae	Leiobunum		shrub wall	no	summer autumn winter
31	Nelima gothica	Phalangidae	Leiobuninae	Nelima	NBNSYS0000013864	herbaceous shrub wall	no	summer autumn

You can get TVK values, also known as the ‘NBN ID code’, one at a time, by searching on the UKSI (UK Species Inventory) here: <http://www.nhm.ac.uk/our-science/data/uk-species/species/index.html>. But if you have a large number to do, you might find it more convenient to use the experimental bulk search here: <http://nbn-sd-dev.nhm.ac.uk/taxonmatch.php>. You can also search and check TVKs using the NBN Atlas website (<https://nbnatlas.org/>).

Most aspects of knowledge-base checking are done ‘up front’ before any visualisations are set (if the top level ‘checkKB’ option is set from the calling web page - see the document ‘Deploying your visualisations’ for details). However, checking for the TVKs can potentially take some time and therefore is therefore not routinely done at startup. But if you have the ‘checkKB’ option set to true, then Identikit puts an additional item on the ‘Select a tool’ drop-down – ‘Check TVKs’ (only if you have a ‘TVK’ column on the taxa worksheet). If you select this item, a report will be generated to indicate whether or not the TVKs referenced in your knowledge-base can be on the NBN.



3.2 The characters worksheet

The example below shows the character worksheet from the biscuits knowledge-base.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Group	Character	Label	Help	HelpShort	Status	ValueType	ControlType	Params	Weight	Strictness	Source
2	Taxonomy	Taxon	Taxon									
3	Structure	Shape	Shape	The actual shape c	The actual shape c	key	text	single		10		
4	Structure	SingleDouble	Single or Double B	Is the biscuit a single layer, or is it a c	key	key	text	single		10		
5	Structure	Coating	Coating	Does the biscuit have a chocolate coa	key	key	text	single		8		
6	Decorations	Holes	Small holes on top	If the biscuit has small holes (e.g. the key			text	single		8		
7	Decorations	Pattern	Decorative pattern c	Is there any sort of <i>deliberate</i> c	key	key	text	single		8		
8	Decorations	Words	Words stamped on	As part of the decoration, some biscui	key	key	text	single		10		
9	Colour	BiscuitColour	Biscuit Colour	The base colour of the biscuit, ignorin	key	key	text	multi		5		
10	Colour	FillingColour	Filling Colour	The colour of the filling which sandwi	key	key	text	single		5		
11	Size	Width	Width	The width of the biscuit, measured in	key	key	numeric	spin	1,100,1	8	0	
12	Size	Length	Length	The length of the biscuit, measured in	key	key	numeric	spin	1,100,1	8	0	
13	Taste	ChocTaste	How chocolatey?	Indicate how <i>chocolatey</i> the bis	key	key	ordinal	single		10	0	
14	Other	Availability	Availability	When was the biscuit 'found'?!	key	key	ordinalCircular	multi		10	0	
15	Other	TopTen	Top ten biscuit?			display						http://i
16												

Regardless of the columns specified on the taxa worksheet, they will only be considered as true characters by the Identikit if they have an entry on the characters worksheet – i.e. they appear under the 'Character' column. So every character that you want to use in the visualisations must have a line on this worksheet.

3.2.1 The Group column

The first column – 'Group' – is used to group characters in the visualisations on the state input controls. The illustration on the right shows the appearance of the state input controls for the biscuits knowledge-base. Here the 'Decorations' tab is selected causing only those characters marked with this group on the characters worksheet to be shown.

All

Structure

Decorations

Colour

Size

Taste

Small holes on top surface?
select option

Decorative pattern on top surface?
select option

Words stamped on top surface?
select option

If the value under 'Group' for every character is set to 'None' (except for taxonomy characters – see below), then the Identikit's state input controls do not group characters at all and the grouping tabs shown in the illustration here are completely hidden. Groups are useful when you are specifying a knowledge-base with lots of characters but can just be a hindrance if there are only a few, so it's useful to be able to specify 'None' in such cases.

The special group name 'Taxonomy' is used to indicate to the Identikit that the character represents a taxonomic rank and that the values in the taxa worksheet represent the taxonomic group to which each taxon belongs for this rank. The order in which taxonomy ranks are specified the on characters worksheet is critical. The higher ranks must appear at the top and the lowest (Taxon) must appear at the bottom. An example for harvestmen is shown below.

	A	B	C	D	E	F	G	H	I	J	K
1	Group	Character	Label	Help	Status	ValueType	ControlType	Params	Weight	Strictness	Source
2	Taxonomy	Family	Family		display						
3	Taxonomy	SubFamily	Sub-family		display						
4	Taxonomy	Genus	Genus		display						
5	Taxonomy	Taxon	Species								

Note that the values of taxonomy characters can also be displayed to users along with morphological characters, or even be used for key input, just like any other character (see section on ‘The status column’ below).

3.2.2 The Label column

The ‘Label’ column is where you specify how the character is actually represented to users of the visualisations. So while the character name, as

Character	Label
SingleDouble	Single or Double Biscuit

specified on the taxa worksheet (and in the ‘Character’ column of the character worksheet) is short and succinct and convenient for you as a knowledge-base author, it may not be that meaningful to a user of the visualisations. So in the ‘Label’ column you indicate the text that will appear to a user in place of the character name used in the knowledge-base. If you like, you can frame these labels as short questions – as in the example for the decoration characters in the biscuits knowledge-base. Alternatively you can use something simpler and leave the questions for the next column we will discuss – the ‘Help’ column.

3.2.3 The Help and HelpShort columns

Character	Label	Help
Pattern	Decorative pattern on top surface?	Is there any sort of <i><i>deliberate</i></i> decorative pattern on all or part of the top surface of the biscuit? Natural ‘cracking’ during baking or an uneven texture does not count as a pattern.

The ‘Help’ column is where you can specify a much lengthier explanation of the character – including instructions on how to use it – to users of the visualisations. When you specify text in this column it is presented to users when they click on the character’s label in the state input controls as shown on the right.

Decorative pattern on top surface?

select option

Character help and information

Decorative pattern on top surface?

Is there any sort of *deliberate* decorative pattern on all or part of the top surface of the biscuit? Natural ‘cracking’ during baking or an uneven texture does not count as a pattern.

You can use plain text in this column, but you can also use simple HTML mark-up if you like, e.g. to embolden or italicise text.

But help on a character is also displayed to a user in a tooltip as they move the mouse over the label for the character. The text displayed will be that in the ‘Help’ column, but if you want a shorter version for the tooltip help, then put that in the ‘HelpShort’ column and that will be used instead.

Note that for numeric characters, you should indicate to the user what units the character is specified in. Sometimes you may want to do that in the ‘Label’ column, but more usually you would do it in the ‘Help’ column.

3.2.4 The Status column

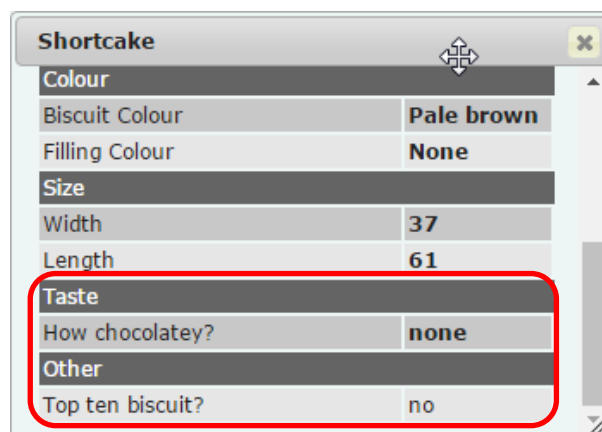
The ‘Status’ column is where you can indicate how the character is to be treated by the visualisations. There are currently only two recognised values for this: ‘key’ and ‘display’. A character with any other value in this column, or no value, will be ignored by the visualisations. (The exception is the ‘Taxon’ character which must appear on the characters worksheet, but which does not need to have a value specified here.)

If you specify a value of 'key' in the 'Status' column, the character against which you specify it will be included in the character state input controls for the visualisations which use them (e.g. multi-access keys). This is the usual state of affairs, but sometimes you might specify a character that you want users to be able to see –

because it contains useful information – but not use for character state input.

For such characters, use a value of 'display'. In the biscuits knowledge-base, the 'TopTen' character is specified with a status of 'display', so it does not appear in the state input controls, but users can see its value elsewhere in the visualisations, for example when clicking on a taxon name in either of the two multi-access key visualisations. Doing this invokes a dialog showing the knowledge-base values for the taxon as shown in the illustration on the right. Here you can see values for all characters including those that can be used for character state input (values in bold) and those which are display only (values in normal font weight).

	A	B	C	D	E
1	Group	Character	Label	Help	Status
13	Taste	ChocTaste	How chocolatey?	Indicate how <i>choc</i>	key
14	Other	TopTen	Top ten biscuit?		display



Shortcake	
Colour	
Biscuit Colour	Pale brown
Filling Colour	None
Size	
Width	37
Length	61
Taste	
How chocolatey?	none
Other	
Top ten biscuit?	no

3.2.5 The ValueType column

Under the 'ValueType' column you specify the data type of the character. In the section on the taxa worksheet we noted that there were three types of character state values – text, numeric and ordinal – these are indicated in the 'ValueType' column simply by entering the words, 'text', 'numeric' and 'ordinal' respectively.

But there is a special class of ordinal values – circular ordinals – that we need to think about. Consider an ordinal character that whose values could take the name of any month of the year. This could be a 'peak emergence' character or something like that. You can see why it would be specified as ordinal – if the user specifies a value of 'june', then taxa with the values 'may' and 'july' should score equally and both would be a closer match than taxa with values of 'april' or 'august'. But what if the user specified a value of

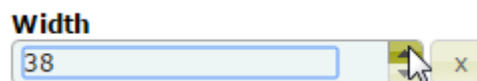
'december'? Taxa with values of 'november' would score a close match, but those with a value of 'january' would score a low match because that's at the other end of the ordinal scale. To account for situations like these, you can specify a 'ValueType' of 'ordinalCircular'. This effectively joins the two extremities of the ordinal range to each other – i.e. a 'circular ordinal' – and in this case, taxa with a value of 'january' would score the same as those with a value of 'november'.

B	F
Character	ValueType
Words	text
BiscuitColour	text
FillingColour	text
Width	numeric
Length	numeric
ChocTaste	ordinal
Availability	ordinalCircular

(Note that for both ordinal and circular ordinal characters, we have to specify the ranking order – this is done on the values worksheet.)

3.2.6 The ControlType and Params columns

The 'ControlType' column is used to specify the type of input control used by the character state input controls on the visualisations. Currently this can only accept one value – 'spin' – for characters with a 'ValueType' of 'numeric'. So any character with a 'ValueType' of 'numeric' must also have a 'ControlType' of 'spin'. This creates a 'spin' input control like that shown on the right. For each spin control you specify, you also need to provide a value in the 'Params' column to indicate the minimum and maximum values that the spinner will travel between when you use the up and down arrows and an increment value that indicates how much the value will increase/decrease when you click the up/down arrows.



The params value is specified in the form 'min,max,incr' – with no spaces. So the value for the character 'Width' in the biscuits knowledge-base – '1,100,1' – indicates that the control will spin between the values of 1 and 100 and increase/decrease by a value of 1. Note that fractional values are permissible for any of these values, e.g. '5.0,15.0, 0.25'. Currently, this is the only use of the 'Params' column.

The 'ControlType' column can accept one of two values – 'single' or 'multi' – for characters with a 'ValueType' of 'text' or 'ordinal'. Both 'single' and 'multi' control types are drop-down lists from which the user can select the appropriate value. The difference between them is that for 'single' controls, only one value can be specified at once whilst for 'multi' controls, more than one value can be specified. The illustration on the right shows the 'multi' type control specified in the biscuits knowledge-base for the character 'BiscuitColour'.



Normally it is best to specify 'single' for 'ControlType' – it makes for a clearer specification of knowledge and more transparent working of the visualisations. But it can be handy to allow the user to 'hedge their bets' for text characters that are particularly variable or subjective such as colour. (See separate document for details on the scoring/matching mechanism.)

3.2.7 The Weight column

Each character with a 'Status' of 'key' can be given a value in the 'Weight' column of between 1 and 10 to indicate its relative importance/reliability as an identification character. The matching scores for each character are adjusted by this value to reflect this relative importance. (See separate document for details on the scoring/matching mechanism.)

Your most reliable characters should be given a weighting of 10 and the weighting of all the others specified relative to this. In the biscuits knowledge-base, the characters with a weighting of 10 are 'Shape', 'SingleDouble' and 'Words' – all characters with are not variable and very unlikely to be misinterpreted by a user. (There's a good argument for giving the 'Width' and 'Length' characters of the biscuits a 'Weight' of 10 since they are likely to vary very little, however it is considered that a user is as likely to

B	I
Character	Weight
Taxon	
Shape	10
SingleDouble	10
Coating	8
Holes	8
Pattern	8
Words	10
BiscuitColour	5
FillingColour	5
Width	8
Length	8
ChocTaste	3

estimate these values as measure them, so they have been given a lower weighting.)

3.2.8 The Strictness column

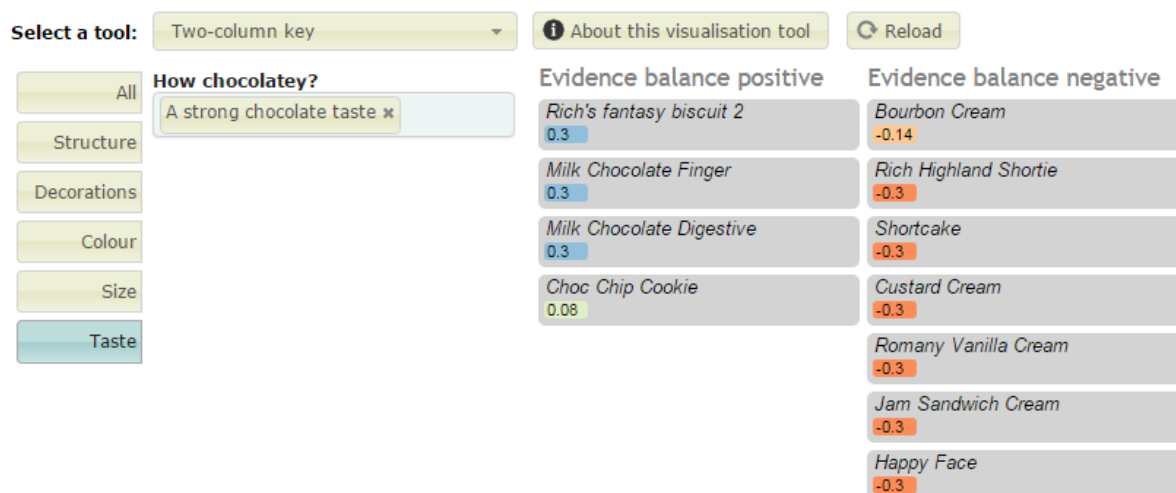
Note that from version 1.7.0, the 'Strictness' column is deprecated in favour of the new 'Latitude' column. Something that's deprecated will still work, but support for it might be removed in a future release. Specifying 'Latitude' and predicting the results of the values you supply is much easier than for 'Strictness'. Documentation on 'Strictness' is currently still supplied below, but you are urged to change to 'Latitude'.

The 'Strictness' column is only relevant for characters with a 'ValueType' of 'numeric', 'ordinal' or 'ordinalCircular'. You can use it to specify how much latitude to give to user-specified values that are not exact matches for a taxon but are relatively close. A strictness value of 10 – the strictest value – requires an exact match between character values specified by the user and those recorded in the knowledge-base for a taxon to score any sort of match for that character. But with a lower strictness value, values that are close but not an exact match can also score. The lower the strictness value, the more latitude there is for a match of some sort.

Imagine a character 'BodyLength' in a knowledge-base for Harvestmen (Opiliones). Suppose that you could populate this field from the literature, but the best data at your disposal was an average body length. You don't have the information required to specify a 'normal' range for the species, so you just use the average body length. Say for species A this was 5.5 mm and for species B 11.0 mm. If you specified a 'Strictness' value of 10 for this character, then a value of 10.5 mm specified by a user for the character would not match either taxon. But with strictness value of 0, species B would score a partial match. Species A might also score a partial match but the score for species B would be much higher.

Similarly, for the ordinal character 'ChocTaste' in the biscuit knowledge-base, a user specified value of 'very' would only score a perfect match for taxa with a value of 'very'. And if the 'Strictness' value for this character was 10, then those taxa with a value of 'moderate' would score nothing – just like those with a value of 'none'. But with a 'Strictness' value of 0, those taxa with a value of 'moderate' would score less than those with a value of 'very' but more than those with a value of 'none'. (See separate document for details on the scoring/matching mechanism.)

B	F	J
Character	ValueType	Strictness
Width	numeric	0
Length	numeric	0
ChocTaste	ordinal	0



The illustration above of the ‘two-column key’ visualisation working on the biscuits knowledge-base shows how each taxon (biscuit) is scored when a single character – ‘ChocTaste’ – is specified by the user as ‘strong’ (the value ‘strong’ has been translated to ‘A strong chocolate taste’ – see the section on the values worksheet for an explanation of that).

The top three matches have the maximum score 0.3 (this is a low maximum because the character has a low weighting of 3). But because this is an ordinal character with a low strictness value of 0, even those values which are not strict matches can score positively, e.g. ‘Choc Chip Cookie’ which has a value of ‘moderate’ for this character state.

You can experiment with different values of the ‘Strictness’ column but, in practice, 10 and 0 are likely to be most useful.

Note that from version 1.7.0, the ‘Strictness’ column is deprecated in favour of the new ‘Latitude’ column and you are urged to change to use ‘Latitude’ instead of ‘Strictness’.

3.2.9 The Latitude column

The ‘Latitude column is only relevant for characters with a ‘ValueType’ of ‘numeric’, ‘ordinal’ or ‘ordinalCircular’. You can use it to specify how much latitude to give to user-specified values that are not exact matches for a taxon but are relatively close. A latitude value of 0 requires an exact match between character values specified by the user and those recorded in the knowledge-base for a taxon to score any sort of match for that character. But with a higher value for latitude, values that are close but not an exact match can also score if they fall within the range expressed by latitude value. The idea is best illustrated by example so carefully examine the examples below.

Character	ValueType	Latitude
Width	numeric	10
Length	numeric	10
ChocTaste	ordinal	2
Availability	ordinalCircular	4

Imagine a character ‘BodyLength’ in a knowledge-base for Harvestemen (Opiliones). Suppose that you could populate this field from the literature, but the best data at your disposal was an average body length. You don’t have the information required to specify a ‘normal’ range for the species, so you just use the average body length.

Say for species A this was 8.5 mm. If you specified a ‘Latitude’ value of 3.5 for this character, then:



- Species A will score the maximum value if the user specifies a value of exactly 8.5.
- Species A will score something (but not the maximum) if the user specifies a value that is greater than 5 ($8.5 - 3.5$) and less than 12 ($8.5 + 3.5$).
- Species A will score nothing if the user specifies a value that is equal to or less than 5 ($8.5 - 3.5$) or equal to or greater than 12 ($8.5 + 3.5$).
- User specified values that are closer to 8.5 than 5 will result in higher scores for Species A.
- User specified values that are closer to 8.5 than 12 will result in higher scores for Species A.

Supposing for species B we have some information that allows us to specify a 'normal' range of body sizes and its body size is expressed as [8-10], then:

- Species B will score the maximum value if the user specifies a value of exactly between 8 and 10 inclusive.
- Species B will score something (but not the maximum) if the user specifies a value that is greater than 4.5 ($8 - 3.5$) and less than 13.5 ($10 + 3.5$).
- Species B will score nothing if the user specifies a value that is equal to or less than 4.5 ($8 - 3.5$) or equal to or greater than 13.5 ($10 + 3.5$).
- User specified values that are closer to 8 than 4.5 will result in higher scores for Species B.
- User specified values that are closer to 10 than 13.5 will result in higher scores for Species B.

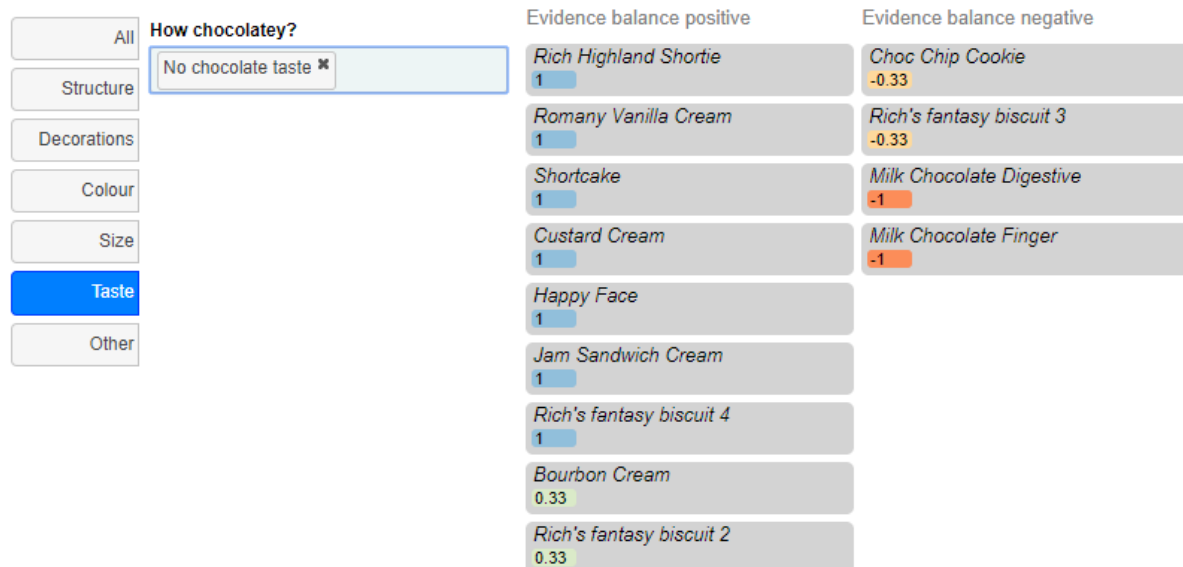
Latitude works similarly for ordinal and ordinalCircular characters, but instead of specifying a value corresponding to absolute values (as you do for numeric characters), you specify a whole number that corresponds to a number of *ranks*.

Consider the ordinal character 'ChocTaste' in the biscuit knowledge-base there are four possible values as shown on the values worksheet and illustrated on the right.

Character	CharacterState
ChocTaste	none
ChocTaste	little
ChocTaste	moderate
ChocTaste	very

If a value of 1 is specified for Latitude for this character and the user selects a value of 'moderate', then taxa with the value 'moderate' will score the maximum value, but taxa with values of 'little' and 'very' will also score something because they are within 1 rank of the selected value. Taxa with a value of 'none' will score nothing.

If a value of 2 is specified for Latitude for this character and the user selects a value of 'none' then taxa with a value of 'none' will score maximally, but those with 'little' and 'moderate' will also score because their ranks are both within 2 of the specified value. In this case taxa with the value 'little' will score more than those with 'moderate' since they are closer to the rank of the selected value. Taxa with a value of 'very' will score nothing because the rank of 'very' is more than 2 outside that of the selected value.



The illustration above of the 'two-column key' visualisation working on the biscuits knowledge-base shows how each taxon (biscuit) is scored when a single character – 'ChocTaste' – is specified by the user as 'none' (the value 'none' has been translated to 'No chocolate taste' – see the section on the values worksheet for an explanation of that).

The taxa which have a value of 'none' for this character have the maximum score 1. But because this is an ordinal character with latitude of 2, the taxa with a value of 'little' also score (0.33) because they are only a single rank away. Even the taxa with a value of 'moderate' score -0.33, which is higher than the worst score of -1, since they are within 2 ranks of the specified value. Only the taxa with values of 'very' score the lowest value (-1) since they are three ranks away from the specified value.

If you want to understand how the scores are actually calculated from using the latitude then consult the document 'Character scoring'.

3.3 The values worksheet

The image below shows the values worksheet from the biscuits knowledge-base.



	A	B	C	D	E	F	G	H
1	Character	CharacterState	CharacterStateTranslation	StateHelp	StateHelpShort			
2	ChocTaste	none	No chocolate taste					
3	ChocTaste	little	A slight chocolate taste					
4	ChocTaste	moderate	A moderate chocolate taste					
5	ChocTaste	very	A strong chocolate taste					
6	BiscuitColour	Dark brown		The overall colour of the bi	The overall colour of the biscuit is dark brown.			
7	BiscuitColour	Pale brown		The overall colour of the biscuit is pale brown.				
8	Coating	Chocolate		The biscuit has a chocolate coating.				
9	Coating	None		The biscuit does not have a coating.				
10	Shape	Round		The biscuit is round.				
11	Shape	Rectangular		The biscuit is rectangular.				
12	Shape	Hexagonal		The biscuit is hexagonal.				
13	Shape	Cigar	Cigar shaped	The biscuit is cigar shaped.				
14	FillingColour	Dark brown		The bisuit has a dark brown filling.				
15	FillingColour	White		The biscuit has a <i>white</i> filling.				
16	FillingColour	White and red		The biscuit has a filling of two layers - one red and one white.				
17	Holes	Yes		The biscuit has small holes on the top surface.				
18	Holes	No		The biscuit has no holes in top surface.				
19	Pattern	Yes		The biscuit has a clear an deliberate pattern on the top surface.				
20	Pattern	No		The biscuit has no clear or deliberate pattern (words or holes may be present)				
21	SingleDouble	Double		There are two separate layers of biscuit.				
22	SingleDouble	Single		There is only a single layer biscuit.				
23	Words	Yes		The biscuit has words stamped on top.				
24	Words	No		There are no words on the biscuit (pattern or holes may be present).				

It should be noted right away that a knowledge-base will work without any rows in the values worksheet. You *do not* have to enter every character state into the values worksheet, but there are some circumstances under which doing so is necessary or just desirable. These are the reasons you might want to make entries in the values worksheet:

- To provide help to the user on a particular character state value.
- To translate the character state text that appears on the taxa worksheet to something more meaningful to users of the visualisations.
- To rank character state values for an ordinal character.
- To specify an order for text character state values in drop-down lists.

Each of these situations is dealt with below.

3.3.1 Providing help on character state values

	A	B	D	E	F	G	H
1	Character	CharacterState	StateHelp	StateHelpShort			
19	Pattern	Yes	The biscuit has a clear an deliberate pattern on the top surface.				
20	Pattern	No	The biscuit has no clear or deliberate pattern (words or holes may be present).				

To provide help on character state values, it is first necessary to ensure that the character itself has some help text specified on the characters worksheet. Providing help text for one or more character states gives maximum help to users of the visualisations who are trying to input character state values for a specimen. To provide help for a character state, enter the character and character state values (as they appear in the taxa worksheet) in the columns 'Character' and 'CharacterState' respectively and then enter the help text under 'StateHelp'.

The illustration on the right shows the effect of doing this for the character states 'Yes' and 'No' for the 'Pattern' characters. The relevant entries on the values worksheet are also shown.

Note that like help text for characters themselves, you can use either plain text in the 'StateHelp' column of the values worksheet, or simple HTML mark-up, e.g. to embolden or italicise text.

Decorative pattern on top surface?

select option

Character help and information

Decorative pattern on top surface?

Is there any sort of *deliberate* decorative pattern on all or part of the top surface of the biscuit? Natural 'cracking' during baking or an uneven texture does not count as a pattern.

Yes: The biscuit has a clear an deliberate pattern on the top surface.

No: The biscuit has no clear or deliberate pattern (words or holes may be present).

	A	B	D	E	F	G	H
1	Character	CharacterState	StateHelp	StateHelpShort			
	BiscuitColour	Dark brown	The overall colour of the biscuit is dark brown. There may be other colours present, but this is the dominant colour.	The overall colour of the biscuit is dark brown.			
6							
7	BiscuitColour	Pale brown	The overall colour of the biscuit is pale brown.				

As well as displaying help text for state values in the help dialog window as shown above, the values are also displayed in tooltips when a user moves the mouse over a selected value in the state input controls. If you want to provide a briefer version of the state help for a tooltip, then specify it in the 'StateHelpShort' column (as has been done for the 'Dark brown' state for the 'BiscuitColour' character in the Biscuits knowledge-base shown above).

3.3.2 Translating character state values

When entering character state values into the taxa worksheet, wordy text character values can be unwieldy and really make it hard for the author to work with the knowledge-base. And yet, since users of the visualisations see and select character state values, they must be meaningful to them. To help with this there is a facility whereby succinct character state values specified on the taxa worksheet can be translated into more meaningful text that the users actually see.

To provide a translated value for a character state, enter the character and character state values (as they appear in the taxa worksheet) in the columns 'Character' and 'CharacterState' respectively and then enter the translated text for the character state under 'CharacterStateTranslation'.

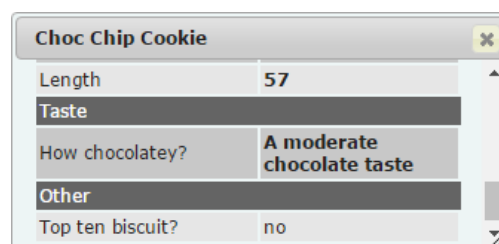
	A	B	C
1	Character	CharacterState	CharacterStateTranslation
2	ChocTaste	none	No chocolate taste
3	ChocTaste	little	A slight chocolate taste
4	ChocTaste	moderate	A moderate chocolate taste
5	ChocTaste	very	A strong chocolate taste

How chocolatey?

select option

No chocolate taste
A slight chocolate taste
A moderate chocolate taste
A strong chocolate taste

In the biscuits knowledge-base, the character states for the character 'ChocTaste' have been provided with translated values. So whilst it is the succinct values of 'none', 'little', 'moderate' and 'very' that appear in the knowledge-base on the taxa worksheet, these are translated, by means of the values worksheet, to the values 'No chocolate taste', 'A slight chocolate taste', 'A moderate chocolate taste' and 'A strong chocolate taste' respectively.



Choc Chip Cookie

Length: 57

Taste: [dropdown menu]

How chocolatey?: A moderate chocolate taste

Other: [dropdown menu]

Top ten biscuit?: no

The images show the effects of this translation in the character state selection control and the knowledge-base values display dialog.

3.3.3 Ranking ordinal character state values

Ordinal characters have been described under the relevant parts of the section on the characters worksheet. But ordinal (and circular ordinal) characters can only operate if a ranking is assigned to each possible character state value and this is done on the values worksheet. A number is not explicitly assigned to define rank; instead, rank is implicitly assigned on the basis of the order in which the values are listed on the values worksheet.

The relevant part of the worksheet is shown on the right for the 'ChocTaste' character in the biscuits knowledge-base. It does not matter whether the values are ranked in 'ascending' or 'descending' order - the effect on scoring will be the same. The important thing is that they appear in rank order. (However the exact order does dictate the order in which they are presented to the user in drop-down lists - see below.)

	A	B
1	Character	CharacterState
2	ChocTaste	none
3	ChocTaste	little
4	ChocTaste	moderate
5	ChocTaste	very

3.3.4 Specifying order for text character values in drop-down lists

When the Identikit creates drop-down lists of character states for use in the visualisation's state input controls, the lists are created, by default, simply by adding state values in the order that they are encountered as the taxa worksheet is read from the first row to the last. Therefore the order in which the character states appear in the lists is, essentially, random. Sometimes this is okay –

especially if there are very few possible character state values. But sometimes this can lead to ugly and unintuitive lists, as in the case of the left-most list above which shows the drop-down list generated by default from a harvestman knowledge-base.

The list on the right was generated after all the possible character state values for this character were specified on the values worksheet in the order in which the knowledge-base author wanted them to be presented to the user. As you can see, this makes for a more intuitively navigable list.

Ocularium tubercle type

select option(s)

- ☐ small spines
- ☐ broad topped tubercles
- ☐ irregularly sized tubercles
- ☐ hair-tipped tubercles
- ☐ blunt hair-tipped tubercles
- ☐ small hairs
- ☐ small hair-tipped tubercles
- ☐ pointed tubercles
- ☐ small blunt tubercles
- ☐ prominent tubercles
- ☐ acute black-tipped tubercles
- ☐ extremely long spines

Ocularium tubercle type

select option(s)

- ☐ none
- ☐ small hairs
- ☐ small spines
- ☐ small hair-tipped tubercles
- ☐ small blunt tubercles
- ☐ blunt tubercles
- ☐ blunt hair-tipped tubercles
- ☐ hair-tipped tubercles
- ☐ pointed tubercles
- ☐ acute black-tipped tubercles
- ☐ extremely long spines
- ☐ robust sharp tubercles

Remember that unless this is an 'ordinal' or 'circularOrdinal' character, the order has no effect whatsoever on the scoring of the character, only the order in which the state values appear to visualisation users.

3.3.5 Specifying state groups

You can group a large number of alternative state values for a single character into a single 'state group' which you can use in the taxa worksheet instead of listing all the alternatives individually (separated by the '|' – or – character). This is really useful if more than one taxon needs to reference the same large group of alternatives. To group character state values for a character into a single state group, just give them the same value in the 'StateGroup' column of the worksheet. In the example biscuits knowledge-base, the character state values for the 'Availability' character have been grouped into one of two state groups – 'odd-month' or 'even-month' and the value 'even-month' has been used to on the taxa worksheet for 'Rich's fantasy biscuit 2'.

3.4 The media worksheet

The image below shows part of the media worksheet from the biscuits knowledge-base.

	A	B	C	D	E	F	G	H	I	J
1	URI	ImageWidth	Type	Priority	Caption	Taxon	Character	State	UseFor	TipStyle
2	# 'Words' character help images									
3	resources/BourbonTop.jpg	100%	image-local		Words on the top of a Bourbon		Words			
4	# 'FillingColour' character state help									
5	resources/CusCreamFilling.jpg	100%	image-local		The filling of a Custard Cream		FillingColour	White		
6	resources/CusCreamXSection.jpg	100%	image-local		White filling in a Custard Cream		FillingColour	White		
7	resources/BourbonFilling.jpg	100%	image-local		The filling inside a Bourbon		FillingColour	Dark brown		
8	resources/BourbonXSection.jpg	100%	image-local		Dark brown filling in a Bourbon		FillingColour	Dark brown		
9	resources/JSCreamXSection.jpg	100%	image-local	2	White and red filling in a Jam Sandwich Cream		FillingColour	White and red		
10	resources/HappyFaceFilling.jpg	100%	image-local	1	White and red filling inside a Happy Face		FillingColour	White and red		
11	# Taxon images...									
12	resources/MCC1.jpg	100%	image-local		A photo showing a top down view	Choc Chip Cookie				
13	resources/MCC2.jpg	100%	image-local		A photo showing an oblique view	Choc Chip Cookie				
14	# Taxon HTML files...									
15	resources/DigestivesHist.htm		html-local	2	History of the Digestive	Milk Chocolate Digestive				
16	resources/DigestivesRecipe.htm		html-local	1	Recipe for a Digestive	Milk Chocolate Digestive				
17	resources/Bourbon.htm		html-local	1	About the Bourbon biscuit	Bourbon Cream				

The media worksheet is where media files are associated with taxa, characters or character states. Identikit visualisations can then display these files in appropriate situations. Situations where you might want to specify media files with the current visualisation tools include the following:

- Images to supplement help text for characters for visualisations that use the state input controls.
- Images to supplement help text for character state values for visualisations that use the state input controls.
- Images to illustrate taxa for visualisations capable of displaying.
- HTML files to provide further information on taxa.

Each of these situations is dealt with below.

3.4.1 Images to supplement help text

We have seen how to use the characters worksheet and the values worksheet to specify help text for characters and character state values respectively. You can also provide images to be displayed in the appropriate places on the same help dialog by specifying the images on the media tab.

If you specify a value of 'image-local' for the 'Type' column, the image must be available from your web server. If you specify a value of 'image-web' for the 'Type' column, you will direct the Identikit to get the image from another website. Make sure that the images you are retrieving are accessible and that you have permission to use them.

If you specify 'image-local' for the 'Type' column then the referenced image should be stored in a sub-folder of the folder which contains your knowledge-base (and therefore CSV files) and the 'URI' column is used to specify the relative path, from this folder, to the image file. If you specify 'image-web' for the 'Type' column, the 'URI' column is used to specify the full web address (i.e. starting with 'http') of the image.

An image is associated with a character by specifying it in the 'Character' column (the 'State' column must be blank to associate an image with the character itself).

To associate an image with a specific character state value, then values must be supplied for both the 'Character' and the 'State' columns.

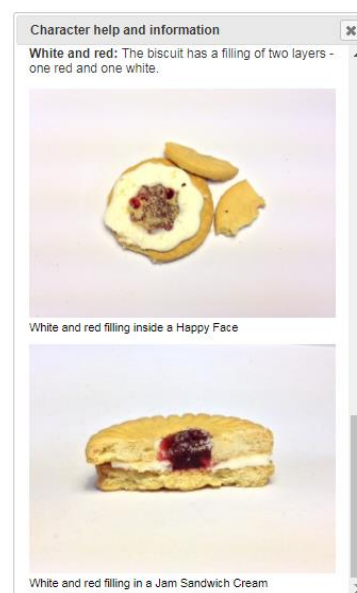


It is normally best to use the value of '100%' in the 'ImageWidth' column. This will ensure that the image is resized to fit the help dialog. Without this value, the image is shown at its natural size.

The 'Priority' column can be used to specify the order in which images are presented when more than one is specified for a given character or character state value. Lower numbers have the highest priority. Fractional numbers are permitted.

The 'Caption' column is used to specify a caption for an image. It's a useful place to credit copyright holders. You can use plain text or basic HTML mark-up in captions.

The top image on the right shows an image specified in the biscuits knowledge-base for the 'Words' character with no images specified for either of the valid character states.



The bottom image on the right shows the two images specified in the biscuits knowledge-base for the 'White and red' state value for the 'FillingColour' character. Although the Happy Face image is listed on the row below the Sandwich Cream image on the media worksheet, it has a higher priority (i.e. '1' in the 'Priority' column) and is therefore displayed first.

The image below shows the rows from the media worksheet responsible for specifying these images.

	A	B	C	D	E	F	G	H	I	J
1	URI	ImageWidth	Type	Priority	Caption	Taxon	Character	State	UseFor	TipStyle
2	# 'Words' character help images									
3	resources/BourbonTop.jpg	100%	image-local		Words on the top of a Bourbon		Words			
4	# 'FillingColour' character state help									
5	resources/CusCreamFilling.jpg	100%	image-local		The filling of a Custard Cream		FillingColour	White		
6	resources/CusCreamXSection.jpg	100%	image-local		White filling in a Custard Cream		FillingColour	White		
7	resources/BourbonFilling.jpg	100%	image-local		The filling inside a Bourbon		FillingColour	Dark brown		
8	resources/BourbonXSection.jpg	100%	image-local		Dark brown filling in a Bourbon		FillingColour	Dark brown		
9	resources/JSCreamXSection.jpg	100%	image-local	2	White and red filling in a Jam Sandwich Cream		FillingColour	White and red		
10	resources/HappyFaceFilling.jpg	100%	image-local	1	White and red filling inside a Happy Face		FillingColour	White and red		

3.4.2 Images to illustrate taxa

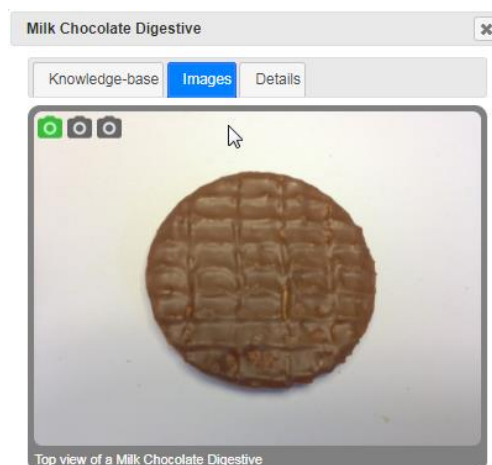
Images to illustrate taxa are specified in exactly the same way as those to illustrate character and state values, except that instead of specifying a value for the columns 'State' and/or 'Character' (which must both be left blank), you specify the name of the associated taxon in the 'Taxon' column.

The 'Priority' column is used in the same way. Make sure that if you have more than one image associated with a taxon the highest priority (lowest number) is associated with an image that is suitable to illustrate the taxon under most circumstances since, if only one image is required by a visualisation (e.g. the 'two-column key' visualisation), this is the one it will use.

The image on the right shows an image group viewer (included in a number of the visualisations) for a taxon (Choc Chip Cookie) for which two images have been specified in the biscuits knowledge-base.

The first image displayed is either that with the highest priority (lowest priority number) or, if no priorities are specified, that listed first in the knowledge-base.

The image below shows the rows from the media worksheet responsible for specifying the images for the 'Choc Chip Cookie' taxon.



	A	B	C	D	E	F	G	H
1	URI	ImageWidth	Type	Priority	Caption	Taxon	Character	State
11	# Taxon images...							
12	resources/MCC1.jpg	100%	image-local		A photo showing a top down view of a Chc Choc Chip Cookie			
13	resources/MCC2.jpg	100%	image-local		A photo showing an oblique view of a Chc Choc Chip Cookie			

From version 1.7.0 of the Identikit onwards you can optionally specify values in the columns 'SmallURI' and 'LargeURI' on the Media tab. The 'SmallURI' can be used to specify a thumbnail image that will be used by the *Galleria* image control. If you don't specify a value for 'SmallURI' the image specified under 'URI' will be used to generate a thumbnail, but specifying your own can improve performance. The *Galleria* control allows users to pop out a full-resolution image. If you specify a value in the 'LargeURI' column, then this image will be used when displaying these full-resolution images to the user (otherwise that specified under the 'URI' column is used). Careful preparation of different versions of your images and use of these columns helps you to generate the most responsive experience for your users.

3.4.3 The UseFor and TipStyle columns

Images specified against particular characters (as opposed to character states) will, by default, be displayed to the user in the character help tooltip. If you wish to restrict an image defined for a character so that it is only displayed in the full help window, then set the value of the 'UseFor' column to 'full'. If you want to restrict an image defined for a character so that it is only displayed in the help tooltip, then set the value of the 'UseFor' column to 'tip'. To display in both situations, you can either leave the column blank, or set the value to 'full, tip' (or 'tip, full').

By default, images displayed in tooltips take up the full width of the tooltip. You can restrict them to part of the width using the 'TipStyle' column. For example, a value of 'left-40' means that the image will only take up 40% of the width of the tool tip and be placed on the left-hand side. Similarly, a value of 'right-50' means that the image will take up 50% of the width of the tool tip and be placed on the right-hand side.

3.4.4 HTML files to provide further information on taxa

HTML files (Hypertext Markup Language files) sound complicated, but they're not. They are simple text files with short tags ('markup') that web browsers understand. For example, if a web browser finds this text in an HTML file - *Anelasma cephalus cambridgei* is relatively small with very short legs - it will know that it needs to italicise the scientific name and embolden the text 'very short legs'.

You can use HTML files to provide detailed information on taxa. The information you supply is up to you – it could be very short ID tips or very long ecological accounts.

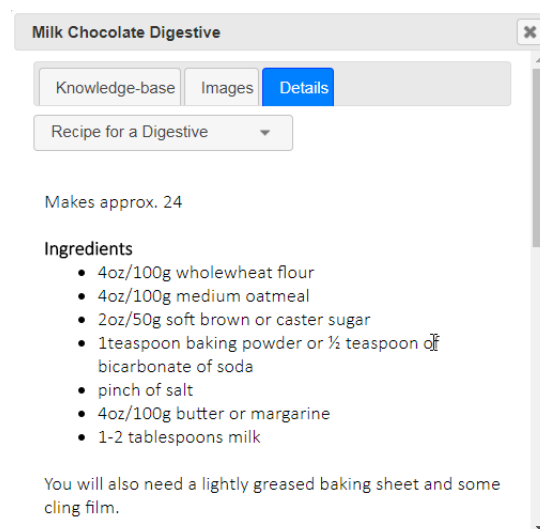
HTML files should be stored in sub-folders of the folder which contains your knowledge-base (and therefore CSV files). The 'URI' column is used to specify the relative path, from this folder, to the HTML file.

Using the media worksheet to specify HTML files associated with taxa is very similar to specifying images associated with taxa. Follow the same steps as for images, but use a value of 'html-local' in the 'Type' column. The value of the 'Priority' column is used by the Identikit to decide which file to show first if more than one is specified for a taxon.

A few simple examples have been included with the biscuits knowledge-base. Two files are specified for the 'Milk Chocolate Digestive' taxon and one for the 'Bourbon' taxon.

The image on the right is from the 'taxon details' dialog box that appears when you select the name of a taxon from any of the default visualisations. The image shows one of the two text files specified for Chocolate Digestive. The other can be displayed by selecting it from the drop-down list (shaded darker green and currently showing 'History of the Digestive').

The lines from the media worksheet which are responsible for specifying the text images for the 'Milk Chocolate Digestive' taxon are shown below.



	A	B	C	D	E	F	G	H
1	URI	ImageWidth	Type	Priority	Caption	Taxon	Character	State
14	# Taxon HTML files...							
15	resources/DigestivesHist.htm		html-local	2	History of the Digestive	Milk Chocolate Digestive		
16	resources/DigestivesRecipe.htm		html-local	1	Recipe for a Digestive	Milk Chocolate Digestive		

Prior to version 1.7.0 of the Identikit, this document suggested using Microsoft Word to create files and use the 'Save As' facility to save them as HTML files. We no longer recommend that approach since Word adds a lot of markup and additional information which is not necessary and can also adversely affect the appearance of the end result in the Identikit.

We have created a 'template' HTML file called 'template.html' which is in the 'resources/text' folder under the 'biscuits' knowledge-base folder. It contains examples of simple HTML markup. This file is linked on the 'media' worksheet of the biscuits knowledge-base to the 'Rich Highland Shortie'



taxon. So you can see what it looks like in the visualisations by using either the 'full details' visualisation from the drop-down menu, or by invoking the full details dialog from and looking on the 'details tab' (for the 'Rich Highland Shortie').

You can open and edit the template file in any plain text editor, such as Microsoft's Notepad, but we recommend another lightweight and free text editor called Notepad++. This is much more user-friendly for editing HTML – it will colour text and tags different and make it much less likely that you will make mistakes. You can download it here: <https://notepad-plus-plus.org/download/v7.5.6.html>.

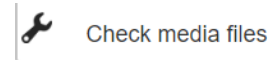
3.4.5 General comments on the media worksheet

Blank rows and those that start with a hash symbol ('#') in the first column – to denote a comment – can be used to organise the information on the media worksheet to help you as a knowledge-base author keep track of what's on there.

Where a single image is used for several purposes – e.g. illustrate a taxon *and* illustrate a character state value – there must be a separate entry (i.e. a separate row) on the media worksheet to specify each way in which the image is used.

3.4.6 Checking media files

Most aspects of knowledge-base checking are done 'up front' before any visualisations are set (if the top level 'checkKB' option is set from the calling web page - see the document 'Deploying your visualisations' for details). However checking for the existence of media files, e.g. local images, web images and html files, can potentially take some time and therefore is therefore not routinely done at startup. But if you have the 'checkKB' option set to true, then Identikit puts an additional item on the 'Select a tool' drop-down – 'Check media files'. If you select this item, a report will be generated to indicate whether or not the files referenced in your knowledge-base can be found.



3.5 The config worksheet

The image below shows the config worksheet from the biscuits knowledge-base.

	A	B	C	D	E	F	G	H
1	Key	Type	Mandatory	Value	Date	Notes		
2	#Tools - those default tools to exclude and non-default tools to include							
3	excludedDefaultTools	config	no			e.g. 'vis1, vis3' to remove the 'Two-column key' and 'Side-by-side comparison' visualisations		
4	otherIncludedTools	config	no			e.g. 'visEarthworm' to add a custom visualisation (must be installed to work)		
5	selectedTool	config	no	vis2		e.g. 'vis2' to start the framework with the 'Single-column key' visualisation selected		
6	checkValidity	config	no	yes		If value set to 'yes' (lower case) validity checks and reporting will be carried out by the visualisation software.		
7	defaultControlGroup	config	no	Structure		Set to the name of a group specified on the characters tab to select this group by default (leave blank for group 'All')		
8	#Main metadata about the key (e.g. for citing the key)							
9	title	metadata	yes	Family Circle Biscuits				
10	year	metadata	yes	2016				
11	authors	metadata	yes	Bell, C.				
12	publisher	metadata	no	Field Studies Council				
13	location	metadata	no	Preston Montford, Shrewsbury				
14	contact	metadata	no					
15	version	metadata	yes	1.0				
16	#Releases - keep track of the release history below this line. Most recent releases at the top. The version code in the 'value' column below this line should match that of the current release above it.							
17	release	history	no	1.0	22/11/2016	Modified by RB to illustrate some of the different features of knowledge representation.		
18	release	history	no	0.1	11/11/2016	CB published an initial test version		
19								

The config worksheet is used to specify some configuration options for the Identikit and metadata about the knowledge-base itself.

The basic format of entries in on this worksheet is simple key/value pairs corresponding to the columns 'Key' and 'Value'. Any keys with the value of the 'Mandatory' column set to 'yes' must be specified in order for the Identikit to function correctly – those set to 'no' are optional. The 'Type' column indicates the type of key/value pair corresponding to the three sections below.



3.5.1 Identikit configuration (type 'config')

Note that from version 1.6.0, all of the 'config' type options are deprecated in favour of Identikit visualisation options specified in the hosting HTML page. See the 'Deploying your visualisations' document for details of how to use the new initialisation options. Something that's deprecated will still work, but support for it might be removed in a future release.

The key '**excludedDefaultTools**' (deprecated from v1.6.0 – replaced by `tombiovis.opts.tools`, see 'Deploying your visualisations') can be used to remove one or more of the default visualisations from the Identikit when used with your knowledge-base. There are currently four default tools:

- Two-column key (vis1)
- Single-column key (vis2)
- Circle-pack key (vis5)
- Side by side comparison (vis3)

The codes in parentheses can be used in the 'Values' column for the '**excludedDefaultTools**' key to exclude the visualisation when the Identikit starts. For example set the value to 'vis1, vis3' to remove the 'Two-column key' and 'Side-by-side comparison' visualisations from the 'Select a tool' drop-down list in the Identikit.

The key '**otherIncludedTools**' (deprecated from v1.6.0 – replaced by `tombiovis.opts.tools`, see 'Deploying your visualisations') can be used to specify non-default visualisation tools that you have access to. You are unlikely to need to use this unless you have access to a non-default visualisation, e.g. you are a programmer who has created a new visualisation for the Identikit.

The key '**selectedTool**' (deprecated from v1.6.0 – replaced by `tombiovis.opts.selectedTool`, see 'Deploying your visualisations') can be used to specify the visualisation that you want the Identikit to show upon initialisation. By default this is the first included visualisation – 'Two-column key (vis1)' if you haven't excluded it. For example to instruct the Identikit to initialise with the 'Single-column key' visualisation, set the value of this key to 'vis2'.

The key '**checkValidity**' (deprecated from v1.6.0 – replaced by `tombiovis.opts.checkKB`, see 'Deploying your visualisations') is a very important one if you are creating a new knowledge-base. Setting the value of this to 'yes' instructs the Identikit to carry out and report on some validity checks on your knowledge base when it is initialising. The image below shows some example output for these checks.



Reload
 Continue

First fix these knowledge-base problems...

Some problems were found with the knowledge-base. They should be easy enough to fix. Read on for more details and guidance.

When you've fixed one or more problems, use the *Save worksheets as CSV* button on the KB to regenerate the CSVs and then click the *Reload* button above.

The problems are colour-coded according to the schema shown below:

- These are serious problems that could cause the visualisation software to malfunction.
- These problems are not likely to cause the visualisation software to malfunction, but you might not see what you expect.
- These are for information only - it may be what you intended to do, but if not, you may as well sort them out. These will not cause the visualisation software to malfunction.

On the characters worksheet...

- There is no row on the *characters* worksheet for the character 'BiscuitColour' represented by a column (column 3) on the *taxa* worksheet. All columns on the *taxa* tab must be represented by a row in the *characters* worksheet regardless of whether or not they are used. Names are case sensitive.
- There is no column on the *taxa* worksheet for the character '#BiscuitColour' which is represented in the *characters* worksheet.
- You must specify a 'Weight' value for 'Coating' because it has a 'Status' value of 'key'.
- '1,100,' is an invalid spin control 'Param' value for 'Width'. It must have the form 'n,n,n' (where n can be any valid numeric value, including decimal).

On the values worksheet...

- There is no row on the *characters* worksheet for the character 'BiscuitColour' represented in the *values* worksheet.

Values on the taxa worksheet...

- The value 'Pink' listed on the *taxa* worksheet for the character 'BiscuitColour' is not specified on the *values* worksheet.

On the media worksheet...

- An image is specified for the character 'BiscuitColour' on the media worksheet, but that character is not on the characters worksheet.
- An image is specified for the character 'BiscuitColour' on the media worksheet, but that character is not on the characters worksheet.
- An image is specified for the character 'BiscuitColour' on the media worksheet, but that character is not on the characters worksheet.

The validity check is very valuable whilst you are developing a knowledge-base and should always be enabled. However once you have developed your knowledge-base, you may want to set the value of this key to 'no' in order to speed up the initialisation of the Identikit.

The key '**defaultControlGroup**' (deprecated from v1.6.0 – replaced by **tombiovis.opts.selectedGroup**, see '**Deploying your visualisations**') allows you to indicate the name of a control group to select by default when your visualisation starts up. The value you specify must correspond to one of the value in the 'Group' column of the characters worksheet. In the biscuits knowledge-base, the value is set to 'Structure' when means that when the visualisation starts it will do so with the 'Structure' group selected rather than the 'All' group (which is the default).

3.5.2 Knowledge-base metadata

Knowledge-base metadata provides your users some important information about your knowledge base. It is used to build a citation for users of your knowledge-base.

The citation is available from the 'Get citation text' option of the 'Select a tool' drop-down list in the Identikit.

The metadata key values for the biscuits knowledge-base shown in the illustration here in the following citation:

#Main metadata about the key (e.g. for citing the key)				
title	metadata	yes	Family Circle Biscuits	
year	metadata	yes	2016	
authors	metadata	yes	Bell, C.	
publisher	metadata	no	Field Studies Council	
location	metadata	no	Preston Montford, Shrewsbury	
contact	metadata	no		
version	metadata	yes	1.0	result

Bell, C. (2016) *Family Circle Biscuits* (Version 1.0) [Knowledge-base] (for FSC Identikit). Field Studies Council. Preston Montford, Shrewsbury. Accessed Wed Nov 23 2016.
<http://localhost:54236/general.html>

Citation text is important because it allows people who make determinations based on your knowledge-base to reference it when required (e.g. for the purposes of verification).

The keys 'title', 'year', 'authors' and 'version' are mandatory whilst 'publisher', 'location' and 'contact' are optional. The value of the 'author' key will be used 'as is' in the citation so it is best to use a format, e.g. 'Surname, initial.', that is most frequently used in citations.

3.5.3 Release history

You can use 'release' keys to keep track of the release history of your knowledge-base. You can include any number of release keys – the value of each of them should be the version number of a milestone version of the knowledge-base. There are a couple of extra columns associated with 'release' keys that you should provide values for: 'Date' and 'Notes'. The date should correspond to a release date and you can use the notes to give a very brief description of the release. Unlike the other keys on the config worksheet, the 'Date' and 'Notes' column values are reported by the Identikit: a release history is displayed when the 'About the knowledge-base' option is selected from the 'Select a tool' dropdown list (see below).

Knowledge-base revision history

Current version: 1.0

Date	Version	Notes
22/11/2016	1.0	Modified by RB to illustrate some of the different features of knowledge representation.
11/11/2016	0.1	CB published an initial test version

The release history is simply presented in the order the keys are entered into the knowledge-base. It makes sense to have the most recent versions at the top of the list and the oldest at the bottom. Note that the 'current version' in the above report is taken from the value of the 'version' key – not the most recent 'release' key – so you should make sure that the value of the 'version' key always matches the value of the most recent 'release' key.

3.6 Including an information file about your knowledge-base

When a user selects the 'About the knowledge-base' option from the 'Select a tool' dropdown list the Identikit looks for an HTML file called 'info.html' in the same folder that the knowledge-base CSV files are found. If it finds the file, it is rendered before the release history table generated from the config worksheet (see previous section) as shown below for the biscuits knowledge-base.

The Biscuits knowledge-base

This knowledge-base was developed by Charlie Bell. It is a 'template' knowledge-base designed to demonstrate some of the features of the Tom.bio Framework for ID Visualisations. Most of the examples in the documentation describing how to build a knowledge-base refer to this knowledge-base. If you are building a completely new knowledge-base, consider making a copy of this template as a starting point for your work.



Knowledge-base revision history

Current version: 1.0

Date	Version	Notes
22/11/2016	1.0	Modified by RB to illustrate some of the different features of knowledge representation.
11/11/2016	0.1	CB published an initial test version



If you want to include external resources in your 'info.html' file, they need to be included in sub-folders of the folder containing your knowledge-base. Furthermore, you need to reference them in a particular way in your knowledge-base as shown below for the a logo image:

```

```

An important part in the above HTML image tag is the '##tombiokbpath##' token. The Identikit replaces this token with the URL of the knowledge-base folder. That means that the knowledge-base can be moved around – e.g. from local computer to a web-server – without having to edit this file every time.

3.7 Dealing with sexual dimorphism

You have a couple of options for representing knowledge for sexually dimorphic organisms. The first option is to create a separate line on the taxa worksheet of your knowledge base for each sex – sometimes referred to as a 'taxon-sex'. So, for example, if biscuits were sexually dimorphic, you would have one line on the taxa workbook for 'Shortcake male' and one line for 'Shortcake female'. An advantage of this approach is its simplicity and a disadvantage is that you need to maintain two lines in your knowledge base for each sexually dimorphic species, repeating much of the information that is common to both sexes.

The Identikit also supports a second approach which is to tag character state values on the taxa worksheet with sex where the value is specific to one sex or another. The example right shows part of the taxa worksheet for a knowledge-base on harvestmen showing how saddle visibility is represented for a number of different species and sexes within a species.

	A	M
1	Taxon	Saddle
22	Platybunus triangularis	indistinct
23	Platybunus pinetorum	indistinct distinct
24	Lophopilio palpinalis	indistinct distinct
25	Dicranopalpus ramosus	no (m) distinct (f)
26	Dicranopalpus caudatus	no (m) distinct (f)
27	Leiobunum rotundum	no (m) indistinct (m) distinct (f)
28	Leiobunum blackwalli	no (m) distinct (f)
29	Leiobunum tisciae	no indistinct
30	Leiobunum sp. A	no indistinct
31	Nelima gothica	no (m) indistinct (f)

If you represent sexually dimorphic characters in this way, then you need to have a character called 'Sex' (with an uppercase 'S') and the possible values must be 'male' and 'female' (all lower case). This text-type character must have a 'ControlType' of 'single' specified on the characters worksheet. Users will be able to specify sex just like any other morphological character.

To understand how this all works, consider the following scenarios for the harvestman knowledge-base shown above.

Scenario 1: user specifies a saddle visibility as 'indistinct' and does *not specify a value for sex*. The result for the 'Two-column visualisation' (or rather part of it) is shown below:



Un-used characters: <input type="button" value="show"/> <input type="button" value="hide"/> <input type="button" value="Reset all"/>	Evidence balance positive	Evidence balance negative
	<i>Homalenotus quadridentatus</i> 0.5	<i>Nemastoma bimaculatum</i> -0.5
	<i>Oligolophus tridens</i> 0.5	<i>Mitostoma chrysomelas</i> -0.5
	<i>Oligolophus hansenii</i> 0.5	<i>Nemastomella bacillifera</i> -0.5
	<i>Paroligolophus agrestis</i> 0.5	<i>Trogulus tricarinatus</i> -0.5
	<i>Paroligolophus meadii</i> 0.5	<i>Anelasmacephalus cambridgei</i> -0.5
	<i>Lacinius ephippiatus</i> 0.5	<i>Sabacon viscayanum ramblaianum</i> -0.5
	<i>Opilio parietinus</i> 0.5	<i>Odiellus spinosus</i> -0.5
	<i>Opilio canestrinii</i> 0.5	<i>Mitopus morio var morio</i> -0.5

Scenario 2: user specifies a saddle visibility as 'indistinct' and specifies sex as 'female'. The result (for the 'Two-column visualisation') is shown below:

Un-used characters: <input type="button" value="show"/> <input type="button" value="hide"/> <input type="button" value="Reset all"/>	Evidence balance positive	Evidence balance negative
	<i>Homalenotus quadridentatus</i> 0.5	<i>Nemastoma bimaculatum</i> -0.5
	<i>Oligolophus tridens</i> 0.5	<i>Mitostoma chrysomelas</i> -0.5
	<i>Oligolophus hansenii</i> 0.5	<i>Nemastomella bacillifera</i> -0.5
	<i>Paroligolophus meadii</i> 0.5	<i>Trogulus tricarinatus</i> -0.5
	<i>Opilio parietinus</i> 0.5	<i>Anelasmacephalus cambridgei</i> -0.5
	<i>Opilio saxatilis</i> 0.5	<i>Sabacon viscayanum ramblaianum</i> -0.5
	<i>Platybunus triangularis</i> 0.5	<i>Paroligolophus agrestis</i> -0.5
	<i>Platybunus pinetorum</i> 0.5	<i>Lacinius ephippiatus</i> -0.5

You will see that *Paroligolophus agrestis*, amongst others, has changed score. This because this species has the value of 'indistinct' for saddle visibility only for the female. So when no sex is specified (or if female had been specified), it scores positively for this character. But when the value of sex is set to 'male' it scores negatively for this character. Note that the character 'Sex' does not itself score.

3.8 The macros worksheet

At the start of this document the 'Save worksheets as CSV' macro was described. There are some other convenience tools on the macro worksheets that you might find useful.

- The 'Get unique values' macro was developed to ease the process of extracting unique values for a character from the taxa worksheet for use on the values worksheet.



- The 'Get max and min' macro was developed to ease the process of extracting min and maximum values for a numeric character for use with the 'Params' column of the characters worksheet.

To get the unique values for a character, select the cells from which you want to extract the values on the taxa worksheet and then go to the macros worksheet and click the 'Get unique values' button – the unique values are copied to the computer's buffer and you can simply paste them anywhere you like. (Don't select an entire column by clicking on the header – the macro won't work if you do that.) The macro takes care of sorting out cells where multiple values are specified with the 'or' character ('|') and it also strips out any '(m)' and '(f)' sex suffixes. Special values (e.g. 'n/a' and '?') are ignored.

To get the minimum and maximum values for a numeric character, select the cells from which you want to extract the values on the taxa worksheet and then go to the macros worksheet and click the 'Get max and min' button – the min and max values are copied to the computer's buffer (as a string – 'min,max,') and you can simply paste them anywhere you like – usually to the corresponding 'Params' column of the characters worksheet. (Don't select an entire column by clicking on the header – the macro won't work if you do that.) The macro takes care dealing with ranges specified as '[n1-n2]'. Special values (e.g. 'n/a' and '?') are ignored.

To use any of the macros in the knowledge-base, you need to ensure that your Excel security settings enable you to do so.

3.9 Using other spreadsheet features

You are free to use whatever features your particular spreadsheet (e.g. Excel) provides when constructing a knowledge-base – e.g. formulas, data ranges, comments, etc. None of these should interfere with the production of the CSV files and it is the CSV files that the Identikit actually reads.

In Excel, for example, you can create as many extra worksheets as you like containing supplementary information to help you as a knowledge-base author. For example if you are converting an existing spreadsheet of knowledge to the Identikit's format, it is convenient to copy the existing spreadsheet, as is, into extra worksheets. They won't interfere with the Identikit.

Another Excel feature that people like to make use of is creating a data range to use as a drop-down list when entering data. You could, for example, create a range from the possible state values for a character as specified on the values worksheet and use Excel's data validation feature to use these values in a drop-down list for the relevant column on the taxa worksheet.

That's an advantage of making use of a spreadsheet format for representing knowledge – it leverages all your existing experience of using spreadsheets and managing information within them.