

Azquo Reference Manual

Draft 7 – Dec 2018



data made simple

Copyright

This document is copyright of Azquo Ltd. 2016. All rights reserved. It is supplied under the terms of an agreement with Azquo Ltd, and may not be used except as permitted by Azquo Ltd.

Azquo is a trademark of Azquo Ltd. Other products and company names herein may be the trademarks of their respective owners.



Contents

1.	Introduction	5
	About This Manual.....	5
	Logging On	5
	Creating Databases	6
2.	Azquo basics.....	8
	Attributes	8
3.	Importing	9
	Set Imports	9
	Values Import	9
	Heading Keywords.....	10
	Context Names	16
	General comments.....	16
	Keywords	16
	Importing from CSV Files.....	18
	Groovy pre-processing	18
4.	Report creation	19
	Data Regions	19
	Permute	20
	Loading a region	21
	Locked and unlocked cells	22
	MultiUser behaviour	25
	Selections	26
	Set Expression syntax	27
	Set operators	29
	Heading functions.....	29
	NAMECOUNT	30
	PATHCOUNT	30
	VALUEPARENTCOUNT.....	30
	VALUESET	31
5.	The admin panel menu	32



Reports	32
Schedules	32
Recipients	32
Databases	32
Upload	32
DB Management	33
Users	33
Az_AllowableReports.....	33
6. The Azquo Excel Plugin	35
7. The report popup menu	36



1. Introduction

Azquo is designed specifically to fill spreadsheets with audited data. From these sheets you can produce the reports you want, when you want them.

About This Manual

The *Azquo Reference Manual* describes how to set up Azquo databases and reports:

- The remainder of this section tells you how to get Azquo up and running.
- Section 2 provides some basic information about the types of data held by Azquo.
- Section 3 discusses the importing of data into a database.
- Section 4 gives details of how to create reports.
- Section 5 describes the administration options.

The separate *Azquo Tutorial* uses a sample database to demonstrate how to create databases within Azquo and generate new reports.

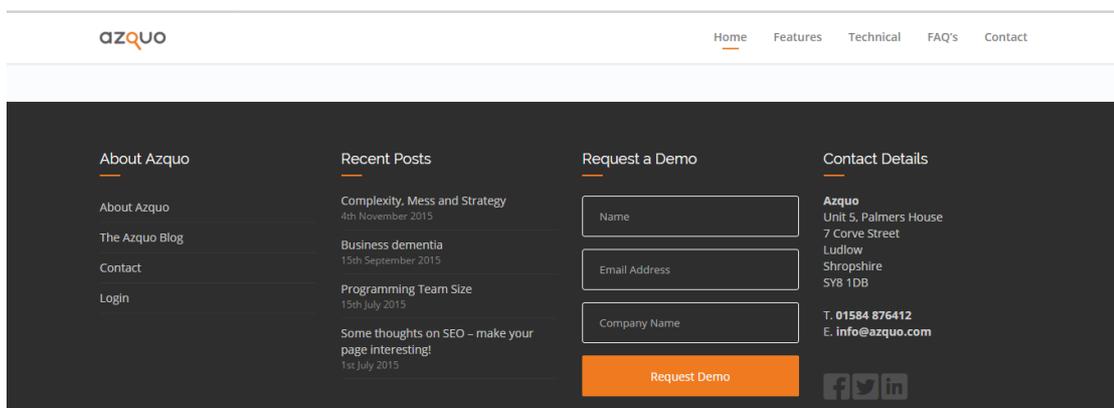
Information on using Azquo is given in the *Azquo User Guide*.

Logging On

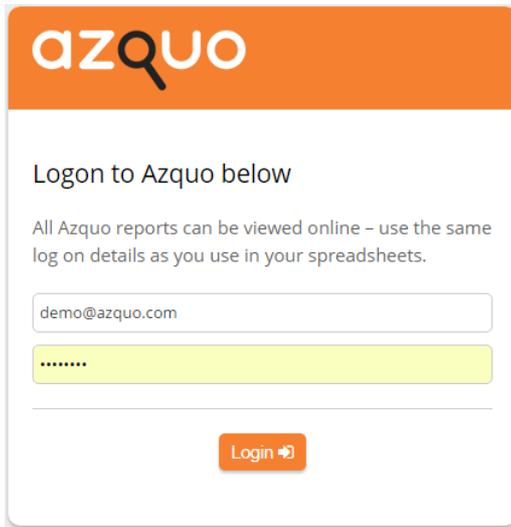
In order to use Azquo, you must have been set up as a user by an administrator, who will have given you a user name and password.

To log on:

1. Go to www.azquo.com. The Home page is displayed. Scroll to the bottom of the page.

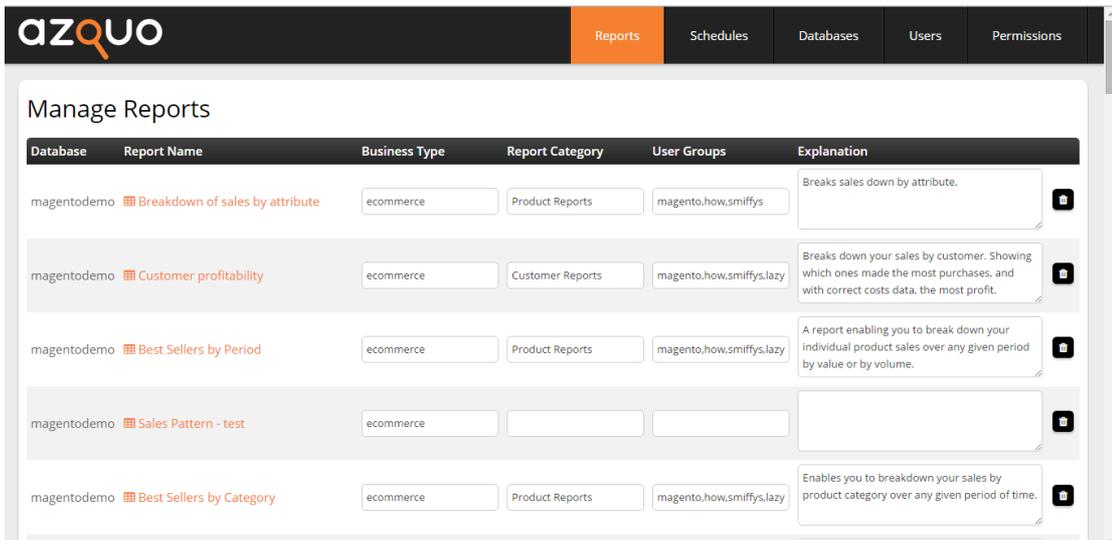


2. Click on Login in the bottom left-hand corner. Enter your username and password.



The login form features the Azquo logo at the top. Below it, the text reads "Logon to Azquo below" and "All Azquo reports can be viewed online - use the same log on details as you use in your spreadsheets." There are two input fields: one for the email address "demo@azquo.com" and another for the password, which is masked with dots. A "Login" button with a right-pointing arrow is located at the bottom.

3. Click on the  icon. You are now logged on and the Azquo reports page is displayed.



The dashboard shows a navigation bar with "Reports" selected. The main content area is titled "Manage Reports" and contains a table with the following data:

Database	Report Name	Business Type	Report Category	User Groups	Explanation
magentodemo	Breakdown of sales by attribute	ecommerce	Product Reports	magento,how,smiffys	Breaks sales down by attribute.
magentodemo	Customer profitability	ecommerce	Customer Reports	magento,how,smiffys,lazy	Breaks down your sales by customer. Showing which ones made the most purchases, and with correct costs data, the most profit.
magentodemo	Best Sellers by Period	ecommerce	Product Reports	magento,how,smiffys,lazy	A report enabling you to break down your individual product sales over any given period by value or by volume.
magentodemo	Sales Pattern - test	ecommerce			
magentodemo	Best Sellers by Category	ecommerce	Product Reports	magento,how,smiffys,lazy	Enables you to breakdown your sales by product category over any given period of time.

Note that the Users and Permissions menu options are only visible if you are logged on as an administrator.

Creating Databases

Having logged on, the following steps create an Azquo database:



1. Select Databases/DB Management.
2. Enter a new database name (database types are optional). If you are a major user you may also choose a server from a list. This creates a blank database.
3. You can now upload some setup information to this database. Choose Databases/Uploads to import data (see Section 3 for the syntax). You have a sample setup file.
4. After importing, you can go to Databases/DB Management and view the data you have imported, or you can empty the database and re-import, or you can delete the database entirely.

When a database is loaded, you also have the option of unloading it. This frees memory, but will cause a delay the next time the database is loaded

5. Having uploaded all the setup information and data you need, you then need to create report and upload them through the same mechanism as above. To create reports, see Section 4.

To view and execute the reports, choose Reports.

For the other uses of the Admin panel, see the 5.



2. Azquo basics

An Azquo database consists of **values** and **names**:

- Values are usually, but not exclusively numbers.
- Names are the labels attached to the values.

Names may be defined as **Sets** (e.g. 'Weekdays' is the set of 7 days). Sets may sit within sets, but there is no requirement for the same 'dimensionality'.

Attributes

Names may also have '**attributes**': information relating to the name which does not warrant a separate name but which is useful for some purpose

Attributes are case-insensitive.

Typical attributes might include ID (the ID of the name when importing from some external database) or Description. Some attributes have special meanings:

CALCULATION	Defines the name in terms of other names – e.g the calculation for 'Unit price' might be 'Price' / 'Quantity'. Calculations are automatically executed before any data is shown.
APPLIES TO	Used in conjunction with CALCULATION specifies a level to which the calculation applies. APPLIES TO usually works on one set, but for 'roll-up' dimensionality across more than one set, you can also use LOWEST which will roll up everything from the bottom level. e.g. Value = Price * Unit price applies to 'All Products' children
DEPENDENT ON	Used for calculations where only some of the names are relevant to that term e.g. 'Product cost'.DEPENDENT ON 'Product cost dependencies', where 'Product cost dependencies' is a set consisting of 'All products' and 'Periods'
INDEPENDENT OF	Used for calculations where one or more names may be omitted from a factor. The converse of DEPENDENT ON .
USE LEVEL	Used for calculations where the value at a low level may be inherited from a higher level. e.g. 'Wheel cost'.CALCULATION = 'Wheel model cost' * 'Number of wheels' 'Number of wheels'.USE LEVEL = 'Car models' children



3. Importing

Importing may be through workbooks or text files.

Text files may be separated by tab, comma or |

If the import is via workbooks, each workbook is treated as a text file, named the same as the spreadsheet.

The spreadsheet name is important. If the spreadsheet name starts with '**sets**' then the import is treated as a SET IMPORT; otherwise it is treated as a VALUES IMPORT.

Set Imports

The first field on each line is the set name. Other fields are the set members. Each set name is cleared before adding the set members, so the new set replaces any existing set.

The import continues until a blank line is encountered.

If the sheet name starts '**sets-**' (e.g. 'sets-date') then the names in the sheet are all considered to be local to the name specified in the title (e.g. 'date'). Thus if the 'sets-date' sheet contains the name 'May' that will not be confused with a customer of the same name.

Values Import

Here each line is considered to be the import of a mixture of structure and values information. For example, e.g. the headings:

Date, Customer, ProductId, ProductName, ProductCategory, Quantity, Price

would presage the import of a number of 'Quantity' and 'Price' values, recorded against Date, Customer and Product. At the same time Product would be considered part of Product Category, and the ProductId would need to be recognised as the definitive way of recognising the product (the ID would usually be an SKU and there might be many SKUs for one product name - e.g. different sizes).

Note that headings may include spaces but these have been omitted here for clarity.

In order to be able to recognise these requirements in import, heading definitions may be given to each column. These may either be written in at the top of the column (e.g. on a spreadsheet import) or referenced through an 'Import Interpreter' – see below.

Heading definitions are in the form:

<Heading keyword>{;<clause>}

Each <clause> consists of <clause name> <clause value>. For example:



CustomerId;language SKU;child of All customers

Product Category;parent of Product;child of All products

Two shortcut are allowed.

- The symbol '.' may be used to mean '**attribute**'
e.g. 'Customer;attribute Postcode' may be shortened to 'Customer.Postcode'
- The 'name' of the heading may be omitted if it is the same as the previous heading
e.g. Custome.name Customer.Postcode -> Customer.name .Postcode

Heading Keywords

Headings may contain the following keywords.

parent of

Include the heading named after the keyword in the set in this column.

e.g. Product Category;parent of Product

This will look for the column heading 'Product', and include the line value from that column in the set identified by the line value in the Product Category column

attribute

Specifies that this column is to be added as an attribute to a name in another column.

This may be shortened to '.' if the previous column has the same heading.

e.g.	Customer	.postcode	.telephone
	SMITH	AB12 AB1	01111101

If the attribute is enclosed in ```, then the attribute name specified is the value of the field with that heading on the import file.

e.g.	Customer	Field	Customer,`Field`
	SMITH	Telephone	01111101

Check <check list>

Checks the value in the cell against a list, separated by ','

E,g Zipcode;check letters = 5;number

Possible checks are:

Number



Letters ⇔ n where n is a number

The line is rejected if the check fails

child of

Put the line value into the set specified.

e.g. '; child of All customers' → the line value will be included in 'All customers'

clear

empties the 'child of' set mentioned in the heading before the import commences

Cleardata

If a cell is blank, remove any existing data from the database

composition

Sometimes you need to adjust the names sent before importing (e.g. to concatenate two names, extract part of a name or add text to a name) this allows you to create a 'new' heading (at the end of the existing headings).

e.g.

Month	Year	YearMonth
Jan	2015	Jan 2015

;composition `Month` `Year`;child of All Months

This would create a name 'Jan 2015', which it would include in 'All Months'.

The 'composition' treats any string enclosed by `` (note this is the quote mark usually above the tab key) as a heading name, while leaving the rest of the string as it stands.

e.g. 'composition The month of `Month` in `Year` → 'The month of Jan in 2015'

'composition' allows three string functions 'left', 'right', 'mid' which operate as in Excel, but when used here the quote marks must only surround the whole function.

e.g. composition Region number `mid(region, 4, 2)`

comment

Allows developers to comment their heading clauses – no effect.

e.g. heading;child of All headings;comment This is for testing only;language headings



composition calc

If the first word after 'composition' is 'calc' then the rest of the clause is taken to be a calculation

e.g. Total Price; composition calc `Quantity` * `Unit price`

default

If there is no value in the line, use this one.

e.g 'default Unknown'

exclusive

This can be used on its own or with a set name.

e.g. exclusive
exclusive All Regions

If no set (such as 'All Regions') is named, the set named in 'child of' is used. Names imported in this column will be removed from any other subset of the set named as 'Exclusive'.

For example, if 'Oxford' exists in the set 'Southern', but, from this heading would be inserted into 'Midlands', both being subsets of 'All regions' then the entry in 'Southern' would be deleted when Oxford was inserted into 'Midlands'.

existing

Only import this line if the value in this column already exists in the database.

Useful for filtering out subsets of large files where 'only' will not be suitable.

Ignore <list>

Do not import any lines with these values in this cell

e.g. ignore Total:, Subtotal:



language

This is similar to 'attribute' insofar as the value is treated as an attribute of the name, but in this case this is taken to be the identifying attribute.

e.g.	Product; language SKU	.name	.pattern
	PR1234	T-shirt	Coca-Cola

Notice that, in this case, '.name' is the same as 'Product' or 'Product.name'.

There is a special case of 'language' which has a different meaning: 'language date'.

language date, language us date

This specifies that the values in the column are dates. The system does its best to interpret the dates. Dates internally are held in the form yyyy-mm-dd.

local

Indicates that the names in the columns referenced are local to that context. (This is used in conjunction with 'parent of'.)

e.g.	Street	Town;parent of Street;local
	High Street	Anytown

This will distinguish this High Street from a High Street in another town.

nonzero

Ignore any 0 values.

only

Only import this line if the value satisfies the only condition.

e.g. only UK

This would only import if the value is 'UK'.

'only' may take wildcards e.g. '*UK' 'UK*' '*UK*'.

override

ignore what's on the line, use this one. – this will usually be a parameter passed in the file name

peers

This specifies that this column contains values, which are to be identified by the headings specified in 'peers'. The headings are enclosed in {} to indicate a set-date.



e.g.	Customer SMITH	Date 01/01/15	Purchases;peers {Customer, Date} 100
------	-------------------	------------------	---

removespaces

Remove any spaces in the cell value

replace

Usually, if a single import contains more than one value with the same 'names' then the values will be added. If 'replace' is specified, the latest value only will be accepted.

required

Do not import any lines without values in the cell

e.g. Sale Date 01/01/2018

Transaction No Product Quantity etc

split

This indicates that the cell may contain more than one name. This is used in categorisation – and must give the character that defines the split.

For instance, if the column heading is

'Product category;child of all categories;split |

Then you might expect to find, as a data cell

Category 1|Category 2|Category 3

To indicate that the product should be categorised into all three categories

topline

the field appears as a name/value pair at the top of the import

e.g.

Month Jan-18 Division wholesale

Field 1 Field 2 Field 3

In this example 'Month' and 'Division' could be referenced by 'topline'



you can also ignore {blank}

Composite Headings

classification <field name>

This is a shortened form of

Child of <heading name>;parent of <field Name>;exclusive

e.g. Product code;classification Transaction Line

is the same as

Product code;child of Product code;parent of Transaction Line;exclusive

lineheading

This is a shortened form of 'child of All Headings;parent of LineNo;language <headingname>

e.g. Customer;lineheading

This is the same as Customer;child of Customer;parent of LineNo;language Customer.name.

This also assumes that an extra field will be created:

LineNo;child of <importfilename> Lines;Composition Line LINENO;language LineNo

where LINENO is a keyword – see below – to be replaced by the line number in the file.

This is useful for pivot table data which will be recreated on every import (as the line number may change from one import to another).

linedata

This is a shortened form of ;peers {LineNo} – and can be used in conjunction with 'lineheading' above

e.g. Value;linedata;nonzero



Context Names

Sometimes in headings, it is useful to have additional headings that may be persistent.

e.g. Purchases Customer Jan Feb March

This can be expressed as

Customer Jan|Purchases;peers{Customer} Feb March Customer.telephone.

In this case | indicates a context name. There may be any number of context names, which persist until there is another column with context names.

(To remove reference to the last context names, you can use '.' as in the Telephone column above.)

General comments

It is not necessary to use the ` symbol for names unless there might be confusion. However including ` in names does not cause problems.

Keywords

LINENO	When used in 'composition' will be replaced by the line number in the import.
FILENAME	The name of the sheet or file (without the suffix) can also be used in 'composition'.

Importing using language tables

Sometimes you need to import equivalent data from more than one source which use different column titles – e.g consolidating data from different accounting systems. In this instance, you can store the language heading information in the Name itself, under the attribute

'HEADING <language>

So the field 'Supplier Billing Address' may have attributes

Accounts system1 <Billing Address> default `Supplier Delivery Address`



Notice that the field name from the import file is here enclosed in <> to distinguish it from Azquo field names, which are, as usual, in ``. The field name from the import file must be the first item, while subsequent text is additional field definition to the default definition defined in the second column below.

This method of import requires (as usual) a field in Azquo named

'DATAIMPORT <import name>

Where <import name> in this case is the first word from the name of a ZIP file containing one or more files, all to be consolidated.

The setup sheet for the consolidation can be created by having columns

Field Name;child of DATAIMPORT <import name> Field Name.HEADINGS <import Name>

Where the second column is the standard method of interpreting the headings assuming that the first column is the headings.

The third and subsequent columns will be labelled

Field Name.<system 1> Field Name.<system 2> etc.

An example of an import sheet using this method can be downloaded from our site.

Once imported, you can inspect All Import Sheets/DATAIMPORT <import Name> to see the children. Each field name there will have an attribute 'HEADINGS <import Name>, together with attributes for each 'language' imported (system 1, system 2 etc). There may also be attributes 'HEADINGS <import name> <language name> to store the additional heading information specific to the language.

Assumptions

Some of the import sheets will not contain all the information required. If the missing data is always the same value (e.g. the originator of the information) then this can be accommodated by setting the 'default' in the language table. However if this may differ from import to import then the system will also need to be fed a set of assumptions before the import. These can be made on the field called DATAIMPORT <specific import name> ASSUMPTIONS, where the <specific import name> is the full name, without suffix, of the zipped import file.

This must be a set, within which there will be names DATAIMPORT <specific import name> ASSUMPTIONS <language name>. This will have attributes that set values for fields on the import file. A sample assumptions table is on our sample language import sheet.



Importing a worksheet using a template

You can set up a template for an import worksheet that will use the presence of a particular name in a given position on the template to interpret the import worksheet. To do this, you clear all the potential data regions of one of the sheets you want to import, then name the 'key' cell 'az_ImportName' (usually this would be the report title, which, on a standard report upload would be named 'az_ReportName'. The report should then also include ranges 'az_DataRegion' where you will expect to find it in the template (make this big enough!), and 'az_ColumnHeadings, formatted as if this were a 'column headings only' upload when saving a report in the usual manner. You can also name other cells that will apply to ALL the data, but the same name as the '..Choice' and 'Chosen' names would be called in a standard report (i.e. if the headings would usually be 'DateChoice' and 'DateChosen', then the cell holding the date should be named 'date'.

Importing from CSV Files

Azquo will accept CSV files with comma, tab or pipe-separated fields. They may have headings as above, but if you first import in a spreadsheet with the same first name as the file you are importing, then the system will 'remember' the headings from the spreadsheet.

Once you have imported from a spreadsheet, you will see that the system creates a name 'Import files', which is a set with elements of the form 'Import <filename>'. Look at the attributes of these names and you will see that the headings from the spreadsheet import have been stored.

Note that this assumes that the CSV import has the same data fields as the original import. Azquo does have ways of matching headings, but it is best not to change the format of the import file if you want trouble-free importing.

For example, if the original spreadsheet was called 'sales', then any file named sales_<any name>, or sales <any name> will be imported using the template headings.

Groovy pre-processing

For files that need more processing than allowed above, you can create a 'Groovy' script to pre-process, but you will need to contact Azquo before attempting this.



4. Report creation

Reports are created in Excel (or LibreOffice) by creating named ranges.

Data Regions

All reports need a named range '**az_ReportName**' which indicates that this is a report, and also is used in the menus.

Each report must have one or more 'data regions' specified. These are specified as

`az_DataRegion<regionname>`.

The regionname can be blank. Data Regions will be loaded in alphabetic order of regionname.

Each DataRegion must, at minimum, have an associated region

`az_ColumnHeadings<regionname>`

which defines the column headings.

This will be a range of names. Some may be sets or set expressions (see below). Where there are sets, these will be expanded to fill the available space, and extend the data area if necessary.

Most, but not all will have an associated range

`az_RowHeadings<regionname>`

Where there are no RowHeadings, the region will be treated as an import sheet, and obey the rules above.

Otherwise row headings behave the same as column headings. Any cell in the data region is defined by the names found in the corresponding row and column headings, and in the context (below). If the headings are more than one 'layer' (e.g. there are two rows of column headings), and one or more layers is blank, the system looks to the left or up (according to whether these are column or row headings) to find a suitable name as heading.

Thus a single name may apply to many or all of the headings. The ambit of the name is only curtailed when the system encounters another name.

If you want to 'cancel' the name without introducing another, you can use '.' as a 'blank name'. It is treated as a null heading – it stops the system looking further without adding another name to the relevant heading list. A '.' on the rightmost row heading, or the lowest column heading ensures that no data will appear on that row or column.

`az_Context<Regionname>`



This contains the 'filters' that are applied to all data in the data region. The context cannot contain sets. If you want to filter a set, then you may need to use the 'as' command (below) to create a temporary set.

```
az_DisplayRowHeadings<RegionName>
```

```
az_DisplayColumnHeadings<RegionName>
```

This is the area where column headings, and sort arrows, will be shown. Cells in these regions already holding text are not filled.

Permute

If you create more than one row of column headings or more than one column of row headings, and the first element is a set, then Azquo will permute the sets. e.g. if the column headings are

```
`Year` children
```

```
`Shop` children
```

then you will have a heading for each combination of year and shop, regardless of whether there is data or not.

Azquo offers a different version of permuting which reflects a similar operation to Excel 'pivot' tables.

If, instead of separately specifying headings you specify a single cell:

```
permute(Year, Shop)
```

then only the columns/rows that have data will be displayed. But there are differences between 'permute' used as a column heading and as a row heading.

'permute(A,B,C) as a row heading will show all combinations of A, B, C children that have data, and show totals and subtotals. You can specify a particular format for total lines by formatting individual cells as 'aztotalformat1<region>' 'azTotalFormat2<region>' etc. If you do not specify a format, total lines are shown in bold.

'permute(A,B,C) as a column heading will always keep set C intact (this assumes that it contains disparate elements) whenever any element of C has a value. If you want to permute individual items, then you need to add a set at the end with a single element. Column heading permutes do not show totals.



ImageStoreName

If a cell is defined in this way, then the user will be allowed to upload images to the sheet (e.g. an image of an expense claim). The cell contents should reflect the parameters of the report.

Loading a region

The methodology for loading any region is as follows:

1. Interpret the headings, and expand where necessary (i.e. where sets are specified).
1. Expand the data region by inserting rows and columns into the second last row/column as required.
2. Load into memory (but not yet into the display) the values found.
1. Sort/filter the values found according to any optional information inserted in a region `az_Options<regionname>`.
2. Display the result. Row and Column headings will be displayed if there are named regions `az_DisplayRowHeadings`, `az_DisplayColumnHeadings`, but will not overwrite any text already written into these regions.

Data Regions may overlap, or superimpose, provided that the loading/saving is done in alphabetic order of regionname.

`az_Options<regionname>`

Each user region may have 'options' in a comma-separated list. These include whether the region is sortable, whether the region is locked for the user, etc. Options are

hiderows=<n> dividing the rows into sets of <n> rows, do not show any set which is completely clear.

hiderowvalues=<n> as above, but will ignore attributes

Hidecols=<n> As above for columns

IgnoreHeadingErrors Heading names that do not appear in the database will be ignored

sortable Allow the user to sort on any column in the data region. For this option to work, `DisplayColumnHeadings` must be assigned for the region

sortcolumn=<column number> specifies a column on which to sort the data before showing it. May also be defined as 'desc'. E.g. `sortcolumn=3 desc`

maxrows= <n> specifies a maximum number of rows to show. If no sorting is specified, then the rows shown are those where the total values on each row are maximum.

NoSave Do not allow the user to save any value in the region



UserLocked Do not allow any other user to save a value which would change the region. This will automatically lock all that data until either the user clicks on 'unlock' at the top of the screen, or the user clicks 'save', or the user loads another worksheet, or the lock times out after an hour. Note this differs from 'locked' and 'unlocked' on headings or context as it relates to other users.

Rowlanguage <language> Specifies that the row headings are to be considered to be in the language stated

ColumnLanguage <language> as above for columns

Locked and unlocked cells

By default, cells are only enterable (or 'Unlocked') if all the individual names that apply to that cell are childless, but if this condition applies, then they are enterable. This rule can be overridden by appending the heading with the suffix '**locked**' or '**unlocked**'. If this suffix is applied to individual headings this affects only the name or set to which the suffix is applied. If the suffix is applied to a context cell, then the override applies to the whole sheet, but may be overridden by suffices on the headings.

A similar useful suffix is '**split**'. Headings which are sets and have this suffix will not store away under the name of the set, but the value will be split equally among all the lowest level children of the name.

For example, values that are entered for '2016' might be split twelve ways among the months of 2016.

...see also 'userlocked' as an option

az_drilldown<region><suffix>

This must be a single cell named range. If this cell exists, then the 'Azquo menu' that appears on right-click of the mouse will include an item 'drilldown <suffix>' (underscores in the suffix will be replaced by blanks.)

For example, 'azDrilldownEntryRegion_by_Business_Unit' would appear on the menu as 'Drilldown by Business Unit'.

The cell should contain a script of the form:

```
in spreadsheet `<reportname>` with <choice> = <value>;{<choice2> = <value2>}
```

The <value> can either be set by the spreadsheet formula, or be [rowheading]
[rowheading<n>] [columnheading] [columnheading<n>]

For example, in spreadsheet 'Input values' with 'Business name' = [rowheading2]

Choice names in any sheet are persistent (see below), so need not be named specifically if they are inherited from the source sheet.



There can be any number of drilldown range cells for any single data region.

az_AllowableReports

Please see 'User' setup below



az_Execute, az_Followon

This must be a single cell range. It contains text with an Azquo 'routine'. If the range is 'az_execute' then a button at the top of the page will allow the user to execute the routine on demand. If the range is 'az_followon' then the routine will be executed automatically each time that the data is saved.

Azquo routines cause this or other sheets to execute using very simple syntax

for each <chosen> in <choice set>

Executes all the lines which have an indentation greater than the 'for each' line.

do <report name>

Executes the report named.

For example:

```
for each [businessUnit] in `All business units` children
  for each [month] in `2016 months` children
    do Calculate sales margins
  do Calculate total margin
```

NOTICE THAT [] indicates user choices, `` indicates Azquo names

set <query>

operates exactly as a 'query' cell in a worksheet. You can refer to user choices (e.g. from 'for each' in [])

.

For example:

```
for each [businessUnit] in `All business units` children
  set `All employees` children + `[businessUnit] employees` as `All employees`
  set `All employees` children * `[businessUnit]` as `[businessUnit] employees`
```

if <set> contains <name>

```
<if commands>
else
  <else commands>
```

operates exactly as a 'query' cell in a worksheet. You can refer to user choices (e.g. from 'for each' in [])



For example:

```
If Weekdays contains [date]
  do Weekday worksheet1
  eo Weekday Worksheet2
else
  do Weekend worksheet
```

delete <name>

useful for tidying up. Notice that the name and all data associated with the name will be deleted

Region cloning

There are some instances where you want to repeat the same region many times with a slightly varied context (e.g. you want a data region for every day of the week). In this case, you can specify a '**az_RepeatRegion**', to repeat within a '**az_RepeatScope**' for each item in a '**az_RepeatList**', where the item is to be placed in the cell '**az_RepeatItem**' on each cloned version. All four ranges are required.

In this scenario, the '**az_RepeatRegion**' defines the size of the full range to be repeated, and the '**az_RepeatScope**' defines the width (i.e the number of horizontal repeats) allowed. If the **az_RepeatScope** needs to be extended vertically, Azquo does so, filling the entire **az_RepeatScope** with clones of **az_RepeatRegion**.

Sheet cloning

Similar to the repeat region, but naming two cells on a sheet '**az_RepeatSheet**' and '**az_RepeatItem**' tells Azquo to clone individual sheets in a workbook to create a target workbook. Using this facility, you can produce report workbooks, which have a front sheet being the summary, and backing sheets that represent all the individual cases

MultiUser behaviour

Default behaviour

Whenever a user saves data, Azquo checks that the data in the spreadsheet has not been changed since it was loaded. This includes ALL visible data in the spreadsheet, not just cells that have changed. If any data has been changed, the system shows a message, and aborts



the save. Users would then need to copy their changes, and reload the sheet before any save would be allowed

As an alternative, a template designer may specify that a region on a template is 'UserLocked'. This will result in all data that has been used to create that sheet being 'locked' by that user – no other user may change any of that data until the user releases the lock (see 'UserLocked' for how the user releases the lock). Any other user loading spreadsheets based on locked data will be told the name of the user who is locking it.

Selections

Most reports will need some selection parameters, and maybe some calculated names.

Method 1

Selection parameters are recognised by '**<selectionname>choice**' and '**<selectionname>chosen**' fields.

The 'choice' field contains the set expression from which to choose, and the 'chosen' field is the field used for the selection. The 'choice' field is optional. If no choice is specified, then the 'chosen' field becomes a simple text entry.

In some instances where choices are made within a data table, the selection list for one column may be dependent on a value chosen in another cell on the row. In this case, there is a function '**contents(<rangename>)**' (e.g. contents(regionchosen)) which allows you to specify the dependent range choice.

An option to select multiple values is also available. If, instead of '**<selectionname>chosen**', there is a range '**<selectionname> multi**' (the range consisting of at least three cells), then the selection is saved away in the database under <selectionname> and can be used as a set within the report.

If the <selectionname>choice set contains the suffix '**showparents**' then each of the items in the list will show the parents. This is useful when there might be confusion

e.g. `Product Attributes` level 2 showparents
Width→1cm
Width→2cm
Length→3cm
Length→4cm

Choices can be set to apply to fields in a data region (e.g. one column of an expenses sheet may be chosen from a list of expense types) In this case, the 'chosen' field is a range covering the column in the data region. This can also be extended to cases where two columns of the data region are choices, and the second depends on the input in the first (e.g. where the first is a category, and the second an element of that category). In this case, the 'choice' cell can contain the Azquo function '**contents**'.

e.g. `All products` children * contents(CategoryChosen)



For this to work, Azquo needs to set up lists of all the possible options internally before loading the sheet. This will include using three cells in unused columns to the right of the data region. You will need to specify a blank column as **<option>list** (e.g. ProductList). You can hide this column and the two columns to the right of it so that the user does not see the 'workings'. You can see an example of this in our 'timeandexpenses' demo on the 'Expenses Input Sheet' (the project chosen is dependent on the customer chosen).

An option to select multiple values is also available. If, instead of `<selectionname> multi`, then the selection is saved away in the database under **<selectionname>** and can be used as a set within the report. Multiple selections are shown in the cell when there are two or less, or all but two or less. Other selections are recorded as 'many'.

Method 2

This is a 'mass production' method that makes the query work in a similar fashion to Microsoft Pivot Tables.

az_ContextFilter specifies a single cell that holds a comma separated list of query names (e.g. Customer, product, date). These will be treated as if they were filters in a spreadsheet pivot table.

az_ContextHeadings specifies the position of the first heading. According to the number of rows in **az_ContextHeadings**, headings are created consisting of heading name, heading value and a blank cell. The format is copied from the three horizontal cells on the first row of **az_ContextHeadings**, regardless of the width of that data region, which is preferably a single column.

In addition to you may also specify '**<name>Query**' for set expressions including 'AS' (see below) which remake sets before the regions are loaded.

Selections are saved so that individual user selections recur the next time the sheet is loaded.

Set Expression syntax

The simplest syntax is '**<name>` children**', which indicates the set of immediate elements of that name.

For example, `2014` children might return the set `Jan 2014`, `Feb 2014` etc. (depending on your setup)

The ` marks are only required when there is a space in the `name`.



e.g. `Last year` children

There are variations on 'children' specified as '**level**' where 'level 1' = 'children'. 'level 2' = grandchildren etc. Level -1 = parents, level -2 = grandparents

Options are **level <n>**, **level lowest**, **level all**, **'from'**, **'to'**, **'count'**, **'sorted'**.

These are self explanatory.

'from' and 'to' can take either the name of an element or its position in the set as an integer.

e.g. 2015 children from `Feb 2015` to 11

Sorting is alphabetic.

If 'to' is specified without 'from' the set is presented in inverse order.

where <condition>

Generally <condition> consists of one or more boolean expressions referring to attributes (NOT data!) separated by **'and'**.

The boolean expressions can use the operators = >= <= > < <>.

e.g. `All bookings` where cancelled = ""

as <name>

asglobal <name>

When added to the end of a set expression, this saves away the result to the <name>. If 'asglobal' is used, the result will be available to all who use the system. Otherwise the result is saved for the particular user.

Attributeset "<attributevalue>"

This gives all the 'names' that have an attribute given.

For example, `May 2016` children attributeset "delivered in" would give any name with the attribute "DELIVERED IN" which was a member of the set 'May 2016'. (i.e any name delivered in May 2016)

<setname> **Hierarchy** <n>

When used as a row or column heading, returns a full array of the setname and it's subsets down to level <n> with subtotals. Note that the number of columns/rows that need to be allocated is n+1.

You can use total line formatting on hierarchy as in 'Permute'



Set operators

A + B return every name in either sets-date.

e.g August children + September children

A – B return everything in A and not in B.

e.g. `All Customers` children - `Sample buyers` children

A * B return everything in A and in B level All (i.e. B and all the children of B).

e.g. All Customers children * Cheltenham

(note that you do not have to specify 'children' in the second term. 'level all' is assumed)

A / B return everything in A and in parents of the children of B .

e.g `All Customers` children / `Product a` returns the set of customers who have bought Product A

The first set is considered the choice set in all operators except '+'. All other sets are expanded to discover the overlap with the first set.

e.g. A children * B is the same as A children * B level all

A children / B is the same as A children / B level lowest

as <targetSet>

Put the result in the target set. This creates a set that persists but is specific to the user. It may be used for data region definitions on any report.

Heading functions

There are a few functions which are specifically designed for headings. Some refer to sets while others refer to the data

COUNT

AVERAGE

MAX

MIN

PERCENTILE(<name>,<percentage>) - e.g. PERCENTILE(`population`, 0.25)

STDEVA



Most of these functions take one parameter, and apply to the set of data that is found.

These return the respective functions on the data that would otherwise be added. Other statistical functions can easily be added

SET
FIRST
LAST

These operate on the set given. The set name can include [ROWHEADING] or [COLUMNHEADING] to represent the headings on each row or column

e.g. SET(`All customers` children sorted * [ROWHEADING]) would return a comma-separated list of all the customers in the set defined by the row heading. If there is more than one row or column headings, then the number can be added to the end

e.g. `All customers` children sorted * [ROWHEADING1]

NAMECOUNT
PATHCOUNT
VALUEPARENTCOUNT
VALUESET

These functions work on sets.

NAMECOUNT(<Setname>) returns the count of the elements of the set specified

e.g The number of orders that any customer has bought could be shown as

NAMECOUNT(`All orders` children / [ROWHEADING])

If the row headings are customers – note the operator ‘/’

PATHCOUNT takes two parameters, comma separated, and represents the number of paths in common. Both parameters can be sets

e.g. if you want to know how many times you have mailed a customer, the result might be in

PATHCOUNT(`ALL mailings` level lowest,[ROWHEADING])

In this instance, the customer may occur in many different mailings, and this function counts them.

VALUEPARENTCOUNT(<heading>,<setname>) is a combined data/set function

It tells how many instances of elements of the set name were in the parents of the data found



e.g. if you want to know how many customers have bought a product then

VALUEPARENTCOUNT(` Order total`, `All customers` children) as a heading will, after all other headings have been considered, tell you how many customers are in the parents of the items found.

VALUESET(<set expression>) is an alternative to <set expression> AS <tempname>

The advantages of VALUESET over 'AS' are that you do not need to make temporary sets, and you can include [ROWHEADING] and [COLUMNHEADING] in the set expression (thus allowing a specific set for each cell). The disadvantage is that the set expression is recalculated for each cell, so the data region may take longer to load

5. The admin panel menu

Reports

You can execute reports from the list by clicking on them. You can also add explanatory notes, download, or delete reports from this menu item

Schedules

You can schedule any report to be sent at a specified interval. You can specify the parameters that should be set before sending the report as a list, including the 'special' values of `|FIRST|` and `|LAST|`. These are particularly useful if you want the latest version of the report to be sent.

e.g. `Business Unit` = `UK`, `month` = `|LAST|`

Recipients would usually be a comma-separated list of email addresses, independent of who are 'users' of the system. However it can also be specified as '**Report Users:<Report name>**' to indicate that the report should be sent to all users who have been set up as having access to the specified 'Report users' report

Databases

Here is where you can create, delete, backup, restore, and inspect databases as well as uploading reports and data.

Upload

You can upload workbooks and CSV files – the latter can be separated by tabs, commas or pipes. The system will automatically detect which version of CSV is uploaded.

If a workbook is uploaded with a range 'az_ReportName' then that workbook is treated as a report, and the contents of the range cell will be put into the list of reports.

Otherwise the workbook is treated as uploads and uses the rules above on each sheet. Any 'values' sheet is remembered, so that, if subsequent CSV sheets are loaded which start with the same name, then the headings on the original sheet are assumed to apply to the CSV sheet.

For example, if the original workbook contained a sheet named 'sales', and subsequently a csv file named 'sales_2017.csv' then the headings on the original sheet would apply to the CSV sheet.



Reports are uploaded to a specific database, but may apply to any database within the business.

The list of uploads shows the last 100 uploads for the business, and allows you to download the files for inspection.

DB Management

The buttons are

Inspect	Allows you to inspect the database.
Empty	Clears the database of all names and values, while not deleting the reports, permissions etc. associated with the database.
Delete	Deletes the database, reports, users, user choices, permissions etc associated with the database.
Unload	Databases usually persist in memory for up to 4 hours of idleness. If you want to clear the database from memory, use this button.
Backup/Restore	Takes you to a separate screen where you can create a new backup or restore from an old one.

There will also be an ability to transfer data from one database to another at any summary level – useful for historic data where you only need summary information.

Users

The 'email' field can be taken to encompass any logon.

Currently there are four statuses

ADMINISTRATOR. Has access to all options and all data

DEVELOPER can access the admin panel, but cannot create users or permissions. Each developer has his/her own development space so that databases are exclusive to the developer and administrators.

USER can only access the system through the report specified by 'Report Name' and 'Database'. This can be a standard report, or a menu that leads to other reports. If it is a menu, the spreadsheet will need to contain a region 'az_AllowableReports' which specify to which reports the user may have access.

Az_AllowableReports as a range may have one, two or three columns

- If it has 1 column, then the reports will all be on the same database as the menu



- If it has 2 columns, then the database for any report will be taken from the second column
- If it has 3 columns, then the name of the report will be taken from the second column, and the database from the third.

Note that the range 'az_AllowableReports' may appear on any report, and will extend the range of allowed reports for any user who has access to the given report

MASTER is a user who also has the ability to download and upload lists of other users. He/she can allocate permissions to other users which may not exceed the permissions given to the master. (i.e. he/she can specify, as their opening report, any report/menu to which he/she has access)

6. The Azquo Excel Plugin

The documentation for this will be available when the plugin is completed



7. The report popup menu

When you press the right button of the mouse while the cursor is on a data cell, two popup menus appear. One is similar to the standard Excel menu, the other is the Azquo menu. It may take slightly longer to appear as the system may need to marshal some facts before showing it.

Options on the Azquo menu which may appear are

Audit Shows you a list of all the values that have gone to making up the current cell, together with who put them in, when and where. The values are shown in reverse order of entry. From this point you can download the audit as an Excel worksheet, or you can, if the 'History' option is enabled, see past values of the same data. You may also, if the data was entered in a spreadsheet, have the option of being taken to that spreadsheet.

Debug On cells which are the result of a calculation, shows the individual elements that have gone to creating the current value

Drilldown (maybe multiple times to different sheets) allows drilldown as specified by the template designer

Region Definition, shows the specification of the current region

Highlight Days allows the user to specify whether he/she wants to highlight recently changed data, and, if só, for what period (from 1 hour to 90 days)

