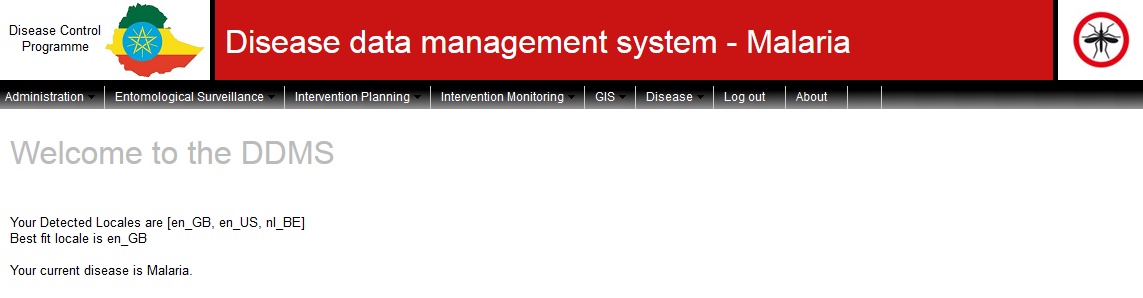
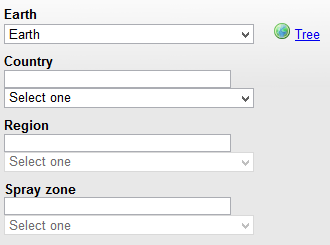
# DDMS - What’s new in 1.05



## General

### Cascaded geo select lists improved in CRUD and QB

In the CRUD and QB, when clicking on the Geo globe http://127.0.0.1:8080/Ghana/imgs/icons/world.png, you get a geo picker which is commonly referred to as the cascaded geo select list.



This feature was not entirely consistent in its functionality. In the CRUD the available universals did not always match the universals that were actually selectable. The cascading sometimes went haywire (cascading = a choice at level *X* activates the select list at level *X*+1). A selection of a geo entity did not always populate the levels above. The reason for this was that there was no logic to differentiate between ‘this is an OK selection to populate the next level, but not for selection purposes’ and ‘this selection will populate the next level but it is also a valid selection’.

Similar problems existed in the QB geo picker. In addition, to safe the selection you simply clicked on the close X at the top right – very unintuitive.

These issues have been addressed. The geo picker should be more intuitive, functional in all circumstances and in the QB, you now have OK and Cancel buttons.



Note: Clicking on the X now closes the window without saving!

## Forms

Quite a few enhancements have been added to forms. All of these will be demonstrated in an exercise.

### Define which fields show in the form table

When you access a form from the menu, you are first presented with a search facility to find and edit an existing form instance. At the bottom of this page is a table showing values of all existing form instances (paginated per 20 instances).

Previously it was hardcoded which field types showed here. Some useful fields were missing (term and geo fields) whilst many irrelevant fields showed and made the table unwieldy.

Now you can define in the form definition which fields you want to see. This includes geo fields and term fields, but not grid fields or multi-select fields.

To display a field value in this table, select True in the Show on view all radio button control.



### Comprehensive search facility on forms

Previously it was possible to search for an instance on its Form ID. You did this in the first CRUD page where the instance table is as discussed in the previous topic.

Now you can select which fields are available to search on.

To display a field value in this table, select True in the Show on search radio button control.



Note that patching to 1.05 sets the default value of all the fields to True for this field. That means that the initial CRUD page will display all the fields in the search facility and will look like the CRUD input screen. It is not the CRUD input screen. You cannot enter new form instances here. It is purely a search screen. If you have forms defined and you do not want all the fields to display on the initial search screen, you must go to the form definition and exclude the fields you do not want to see here.

### Defining calculated fields in forms

Previously you could define fields in a form without the ability to define no-input fields that calculated a value from other values.

It is now possible to define calculated fields in a form. This is the basis of powerful functionality but is currently still limited:

* You can define calculations, but only based on fields of the form itself
* These calculated fields show in the QB and can show on the CRUD search screen but you cannot define aggregation fields for the QB. This means that you can define a field that shows calculated data for the form instance itself, but no QB calculations that aggregate across all form instances. It is in the roadmap to do this but this is a major change that will require funding beyond the maintenance contract\*

The GUI allows basic field calculations (e.g. field1 + field2 / 100 \* 0.7) and JavaScript coding very similar to the BIRT expression builder. It was not possible to replicate the BIRT expression builder due to security concerns. A web programming tool called OGNL (Object Graph Navigation Language) has been integrated into the DDMS. It helps secure the integrity of the DDMS. If we were to open the form builder to JavaScript, a competent user would have a portal to affect the DDMS code in unsavoury ways.

In theory, you can use any JavaScript but in practise there are discrepancies. One deviation that has been found thus far is field type conversion. An example is this:

If an expression is: myRatio = myInteger1 / myInteger2

OGNL will determine that the expression uses integers and therefore the result is typecast as an integer (typecast means that the new, previously undefined variable is automatically allocated a type – in this case Integer). Therefore the result will be forced into an integer and 3/2 will be 2. There is no easy solution to this because changing this OGNL behaviour would mean creating a complete layer around OGNL.

The way to get around this issue is by either defining all your fields as Float, or by converting the field type in the expression (e.g. use the expression myRatio = (float)myInteger1 / (float)myinteger2 ). In fact, only one term needs to be converted to Float to convince OGNL that the type of the result needs to be float.

Grid fields and multiple term fields cannot be used in form calculations.

**Important note:** If you edit a form calculation, the system will update all existing form instances to reflect the new calculation. If there are a lot of form instances, this can take a long time.

### Vector collection form

A new standard form has been added – the Vector collection form. This is an empty form with the usual Form ID and a Collection ID field. The Collection ID field is linked to the collection ID of vector collections.

By using this form you can create a user defined extension to the vector collection component.

The Collection ID field has a search facility on collection IDs and will only accept an existing ID.

### Ability to clone forms

It is possible to clone any existing form as often as you like. This is useful if two forms are very similar in look and feel, and it is particularly useful for the Vector collections form. This feature allows you to create as many vector collection extensions as you like.

When creating a vector collection extension, first clone the Vector collection form so that the base form template is left untouched. Then add any fields you like.

Exercise

Human Blood Index assays are not part of the standard DDMS functionality.

* Clone the Vector collection form
  + Open the form
  + Click on the  button
  + Select a new Form name and Display label  
    
  + Save the clone by clicking on the  button
* Create the integer fields *Number of vectors tested* and *Number of vectors testing positive for human blood*
* Create a calculated field *Human Blood Index*
  + Add a Float field
  + Set Is Expression Attribute to *true*
  + Build the expression @Float@parseFloat(numberOfVectorsTestingPositiveForHumanBlood) / numberOfVectorsTested
* If you like, you can also add a grid called Other *Blood and* add an integer field to the grid called *Host*. You would need a term tree branch listing all animal types you might be checking for.

To test your handiwork you will have to add this form to the menu and give yourself permission to use it. You can create a form instance and change the calculation to see that existing form instances are updated.

## Ability to exporting and importing forms and queries

It is now possible to export and import form definitions and queries. This allows you to work on queries for reports and forms on a copy of a live system and when you are happy with your work, you do not have to recreate them on the live system. You can export them in an XML file and import them in the live system.

This is complex new technology and is probably not bug free.

A known limitation for forms is that you only export the form definition. This is all the work you do in the Form administration component. It does not include work you do in the term tree or the GUI configuration components.

This is not oversight. For forms you often define new terms in the term tree. However, you can also refer to existing branches or create new branches using existing terms. The work you do in the term tree is not recorded in the Form definition. The reference to these terms is also not recorded in the form definition. To create an XML file that tries to understand what you have done in the term tree is very challenging. Therefore when you export a form you only export the work you did in Form administration. Any other work required to get the form working needs to be repeated on the target installation.

If you create new terms for your form, use an Excel sheet and import the terms. Then you can use the same Excel sheet to import them in your target system. The GUI configuration needs to happen manually. There is no import for that.

Queries should export and import without any issues.

To export a form, do the following:

* Select the form in the Form administration page
* Click on the  button (top of form)
* Select Save file and navigate to the folder you want to save the XML file to
* Type in the name of the file you want to save it as and click on the Save button

This will create an XML file you can import on a different system.

To import:

* click on the  option at the top left
* Browse to the folder where the XML file is
* Click on Submit

For queries, you will see the  and  buttons at the bottom of the page.

## Mass Data Deletion

Mass importation of data can lead to mass errors. Before importing data a backup should always be made. Even manually entered data are sometimes found to be consistently wrong.

Because of the data update behaviour on import, sometimes corrections can be made by correcting the data in an import file and re-importing. At times this is not possible – mostly due to unique identifier issues.

The functional DDMS components that have a search facility on their CRUD page also have a mass delete feature. This is hidden from any role other than the System role. Deleting data does not have an undo option and because it should be rare that mass deletions are required, it has been made intentionally difficult to access this feature.

* Add the System role to your user profile and log out and in again
* Go to Entomological Surveillance > Search/Enter a collection

Next to the Search and Create buttons, there is now also a Delete matches button



Clicking this button will delete all the records that match the search criteria, as well as any dependent records (e.g. assays) so as not to leave orphans in the database.

Clicking it without entering search criteria will delete all the collections in the database.

## Entomology

### Abundance calculations are more flexible

It is now possible to use the abundance calculations with any of the Collection fields.

### Sub collection label changed to Collection detail

The sub collection label in the query builder was misleading. The fields in that section apply to any collection, even if no sub collections are used. The label has now been changed to Collection detail, which is more accurate and less confusing.

## IRS

### Intervention teams can be imported

Import functionality for teams has been added. You can define teams with or without operators. Following standard DDMS import functionality, the import sheet can have several tabs. Thus it is possible to import the operators in the first tab and the teams with operator allocation in the second. The operator tab must come first.

### Number of structures validation on level 1 is configurable

Previously only the values 0 and 1 were accepted for number of structures on the SP1 form. This is because SP1 is by definition at structure level. To accommodate countries that work at the household level, you can now switch off that validation.

* Go to Intervention Planning > Configure IRS
* By ticking the check box in the Validation section, you are removing the validation



### 2 new term fields are available in IRS level 1

In the IRS SP1 tale there are now 2 term fields, meant for recording a configurable reason for not spraying a structure and the structure type.

Since these are term fields, you can re-use them for any purpose. They are at level 1 only. Since levels 2 and 3 do not record structure detail, these fields would be meaningless.

## Case surveillance

It is now possible to query calculated population figures and weekly thresholds.

The population figures are annual figures. They are calculated as the last known population figures increased with the annual growth rate.

In the QB, both population and thresholds are inner joins with cases. If there are no cases you will not see thresholds or population. To see population or thresholds for other periods, refer to the population and threshold CRUDs.

## Administration

### Scheduler: clear history

The Scheduler has a History tab that shows all the jobs that have run. This table grew indefinitely. Now it is possible to truncate the table by clicking on the  button.

### See who is logged in

In a multi-user environment it can be useful to see who is online if you want to bring the system down for maintenance. By default you now have this option in the Administration > Person administration sub menu.

### Command Line Interface

The DDMS now comes with a Powershell script that allows you to start and stop Tomcat (similar to using the Start and Stop buttons in the Manager GUI). The Powershell script is called manager.psi and sits in the C:\MDSS\manager directory. Using this script in the Windows Powershell is quite tedious, so two Windows batch files have been added: ddmscli.bat and ddmsschedule.bat. These batch files ensure that the Powershell script is called in the correct way.

To use the command line interface (CLI), open a DOS box by running cmd.exe. Browse to C:\MDSS\manager by using the cd command:

*C:\Users\miguel> cd C:\MDSS\manager*

By typing ddmscli –help you will see the following explanation:

-startApp <appName> Starts the DDMS application by name <appName>.

-stopApp <appName> Stops the DDMS application by name <appName>.

-backup <appName> -filename <filePath> Starts backing up the DDMS application <appName> to a file <filePath>. This path can be either absolute or relative.

-restore <appName> -filename <filePath> Starts a restore on <appName> from file <filePath>. This path can be either absolute or relative.

-backupAll <directory> Backs up all applications to the specified directory.

-startTomcat Starts Tomcat and all DDMS applications.

-stopTomcat Stops Tomcat and all DDMS applications.

-getTomcatStatus Returns the current status of Tomcat.

-scheduler Prints some information about running scripts in the Windows scheduler.

The CLI allows you to stop, backup or restore and start individual applications, which was not possible before. With this functionality it is possible to run a server with 10 apps and back them all up or restore backups without causing more downtime than necessary.

Another benefit of the CLI is that these commands can be put in a batch file and scheduled using the Windows scheduler. An example of such a batch file is:

PATH = %PATH%;C:\MDSS\manager

Call ddmsschedule –stopApp Ethiopia > logEthiopia

Call ddmsschedule –backup Ethiopia –filename “D:\DDMS files\Backups\Ethiopia” >> logEthiopia

Call ddmsschedule –startApp Ethiopia >> logEthiopia

Call ddmsschedule –stopApp Ghana > logGhana

Call ddmsschedule –backup Ghana –filename “D:\DDMS files\Backups\ Ghana” >> logGhana

Call ddmsschedule –startApp Ghana >> logGhana

### CLI to install and patch

It is also possible to run an installation or a patch from the command line. This is useful if you want to set up a new or patch an existing test environment with multiple apps. Previously this could require a lot of attention because the process was interactive and you manually had to kick off each next step.

Here is an example of how you can automate it in a batch file:

start /wait ddms1516.exe /S -app\_name India -master -install\_number 0 > C:\MDSS\logs\automated\_install.log

Call ddmscli -restore India -filename "C:\DDMS Deliverables\India(1514)-2015\_03\_31-16\_55\_36.zip" >> C:\MDSS\logs\automated\_install.log

start /wait patch-1.05.0016.exe /S >> C:\MDSS\logs\automated\_install.log

Call ddmscli -startTomcat >> C:\MDSS\logs\automated\_install.log

### Import error file is saved on the server

Previously errors during import were written to an Excel spreadsheet and passed to the browser (Firefox). Unfortunately it could happen that the connection was somehow lost. The import continued but there was no way to get the error file.

Now the error file is also written to the server in C:\DDMS\Import errors. Note that subsequent imports of the same file will overwrite the error file as it will have the same name.

## Enhancements in the immediate roadmap

* QB performance enhancements
* Cascaded parameters in BIRT
* Mapping – toggle visibility of layers
* Batch jobs in DDMS scheduler
* Term tree search facility