**Creating a new camera**

**Checklist and important reminders**

**Step 1**

Read the specification that specifies the structure of the file. You are looking for the fundamental parts that make the readable structures. Do not be too concerned with values held by variables/structures/containers in the file unless those *directly* affect the choice of object in the following data.

A record number that dictates the next structure to be read is of concern to the BDD and the AcceptRules. A value that effects 'the use' of a following field or structure is *not* of concern to the BDD.

**Step 2**

Create BDD to grab the structures of the file.

Using Miraplacid Data Viewer, test and refine the BDD until you are happy you are reading the file correctly and collecting all of the required parts into the tree structure.

Test your BDD using **bindirectoryreader** on large number of sample files.

Have the BDD reviewed by more experienced developers.

Note: Be aware data heap areas left in the document will be zeroed by the generated camera. We expect every part of the tree to be kept and the rest nulled for protection

**Step 3**

Create setup\_*camera-name*.bat in directory **BDD\_DVL\_to\_code\_scripts**.

* Ensure 'case' of **BDD\_ROOT** variable is correct and consistent otherwise names of classes may be wrong in the generated code.
* **BDD\_ROOT** must be the name of the first BDD object from which to start reading, not the structure type (which is often in uppercase)
* Check there are no trailing slashes on file paths

**Step 4**

Create DVLs using the specification to identify value ranges and cross references that should prove the document contains sensible data.

* You MUST NOT duplicate AcceptRules in your validations. If a rule fails 'Acceptance' in BDD it will never reach Validation.
* Remember BDD is the broad 'catch whole file' mechanism, it should not block the reading of files. Validation rules can apply much deeper more detailed tests. If a validation test turns out to be too pedantic, and a lot of files start failing with it, the client program (such as the Enterprise Product) can add that validation to its allowed list. If we are too strong in the BDD, we will reject files and not output any useful reason for the rejection and have to ship a replacement DLL which we would rather not do.

**Step 5**

Run **dvl\_checker** on your DVL directory to test all the DVL files and confirm there are no syntax errors.

**Step 6**

Use **generate\_code.bat** to create the camera code

**Step 7**

Create "c" interface file for the camera

Add **camera id** to **glasswall.classic/code/utils/camera.id.h**

**Step 8**

Create issue lookup table

**Step 9**

Add sanitisation lookup table

**Step 10**

Add analysis IDs and strings to **analysis\_schema.h** and **analysis\_schema.strings.h**

**Step 11**

Add file detection to the **filetypes.database.c**

**Step 12**

Add **camera id** to **getIssueID** function and **getErrorMsgID** function.

**Step 13**

Update Product document.