

## Objective

Import FID information into a Canvas Recipe

## Scope

- 1) Creating a means for us to accept fid information from outside Canvas and inserting the information into a Canvas recipe. Also, if there is a means to extract fid information (found locations of fids relative to the pattern/workpiece origin).

This is related to the idea that we might get fid information from some sort of upstream or downstream process equipment (e.g. die attach or pre-inspection equipment). Rather than having to re-run all of the fiducial captures, if we can just capture some workpiece or global level fids to identify where the part ended up from the conveying, if we can then use downloaded fid information to make the fine adjustments in the recipe/production execution can save us processing time. In this case, we would need to be able to

- a) identify within the recipe all the current/programmed fids and the relative locations that are using those fids for repositioning
- b) the coordinate system (machine origin, workpiece origin, pattern origin, etc.) that the fids are being tied back to
- c) inserting into the recipe or production run execution the fid information from the upstream equipment pass down
- d) processing the recipe with accurate dispense (or vision identification/representation of dispense) to targeted locations based on the received fid information

- 2) Creating a means for a user to insert “generic/synthetic” fids into a Canvas recipe.

This one is somewhat related, but more relevant to CAD Import or offline recipe creation. Today, once a recipe is created that doesn't utilize a captured image Canvas, the user still has to go to a machine and teach fiducials from the base part to complete their recipe/program. If there were another means to insert “synthetic” fiducials, this could help with saving on some time from the programming side of things. A “synthetic” fiducial might be like a BMP/JPG sketch of a part to identify like a corner of a die (white-black or black-white synthetic image) that could be used for our corner finder or like a drawn circle/dot from MS Paint or similar drawing utility (again black and white only) for creating a synthetic dot for a dot fid (location or size, etc). The point here being just to help save some programming time on our system (and potentially in the future allowing for a more fully automated programming method).

Target outputs:

- a) Need to identify our fiducial structure within a Canvas recipe to set the requirements for formatting any such synthetic fids (e.g. maximum/minimum model size)
- b) Need to identify which fid modes/algorithms where a synthetic fid could be used (e.g. dot finder, corner finder, edge finder, shape/template finder, etc.) & what are the requirements for successful detection with such a fiducial (e.g. light to dark scan vs dark to light scan, directionality of the scan for edge/corner finding, etc.)
- c) Define a methodology for creating a synthetic fiducial (use MS paint? Some other program? What should be the allowed file format for such a fid?)

- d) Ability to insert the synthetic fiducial into the process program at a relevant fid find operation/location within the program. (Likely requires a copy of Canvas to work with.)
- e) Ability to define rotation or other critical fid teaching information to be input to the process program/recipe with the fid information.

#### Deliverables

Deliver source code capable of importing fid information and location per the information provided above..