darktable - Feature #8585

presets handling

12/06/2011 08:09 AM - Zeus Panchenko

Status:	Triaged	**Start date:**	
Priority:	Medium	**Due date:**	
Assignee:		**% Done:**	20%
Category:	General	**Estimated time:**	0.00 hour
Target version:	Future		
Affected Version:		**bitness:**	64-bit
System:		**hardware architecture:**	amd64/x86

Description

when db removal necessity arise, we need to handle **presets to be not lost and to able to be imported back after db re-creation**

may be to handle them like styles are handled?

Related issues:

Duplicated by darktable - Feature #8788: save presets (and any non-database-s...	**Duplicate**	**06/19/2012**
Duplicated by darktable - Feature #9296: allow export/import of presets	**Duplicate**	**03/15/2013**
Duplicated by darktable - Feature #9702: Module presets should be stored else...	**Duplicate**	**12/02/2013**

History

#1 - 10/05/2012 12:47 PM - Tobias Ellinghaus

- % Done changed from 0 to 20

- Target version changed from --- to Future

- Priority changed from Low to Medium

- Status changed from New to Triaged

#2 - 10/05/2012 12:58 PM - Tobias Ellinghaus

From IRC discussion:

<parafin> we really should split presets from db before next release
<houz> maybe not split but do something like with xmp files: write to both db and external file. on startup the db should be purged (wrt. presets) and the files should be read

#3 - 01/06/2013 12:32 AM - Pascal de Bruijn

I agree that exporting our presets just as we do with the styles is a good idea with regard to database failure survivability.

But I don't think we should always import that on startup.

Our database is the leading source of information (for example like with the .xmp's), so to keep things clear it should be either leading for everything or nothing in my opinion.

#4 - 12/02/2013 07:39 PM - Dave Woodfall

Tobias Ellinghaus wrote:

> From IRC discussion:

<parafin> we really should split presets from db before next release
<houz> maybe not split but do something like with xmp files: write to both db and external file. on startup the db should be purged (wrt. presets) and the files should be read

The problem here is that presets are global, not connected to one image like xmp data. They would be better off in an presets.db or similar IMHO.