

Cadelec 2006 – New Functions and Capabilities

Users often need to re-use pages of existing projects that are similar to the current project in order to save time. After importing the raw drawing, modifications must be made to make the imported drawing compatible with the current project. The most frequently required changes are related to identifiers (names) for the following Cadelec objects:

<i>Cadelec Object</i>	<i>Cadelec Attribute names</i>
Terminal strip	PREFIX
Installation	LOCATION
Wiring Zone	ZONE
Circuit	CIRCUIT
Cross Reference Function	XR_FUNCTION
Cable	CC01, CC02, T_CA, F_CA

A simple search and replace function named `cxrf` was available in prior versions of Cadelec to change Cross Reference Functions. This function performed a simple text *find and replace*, but did not check for consistency with the new project. This was not a serious drawback for Cross Reference Functions as Cadelec would check for errors during the Cross Reference Analysis.

Other objects needed to be manually checked and modified and mistakes would occur that were not easy to locate and would take long time to correct..

Users have suggested implementing a more general way to convert pages imported from other projects more systematically while checking for consistency with the new project. The following new functions have been implemented:

Attribute Search and Replace functions

The following new functions have been implemented to search and replace attributes:

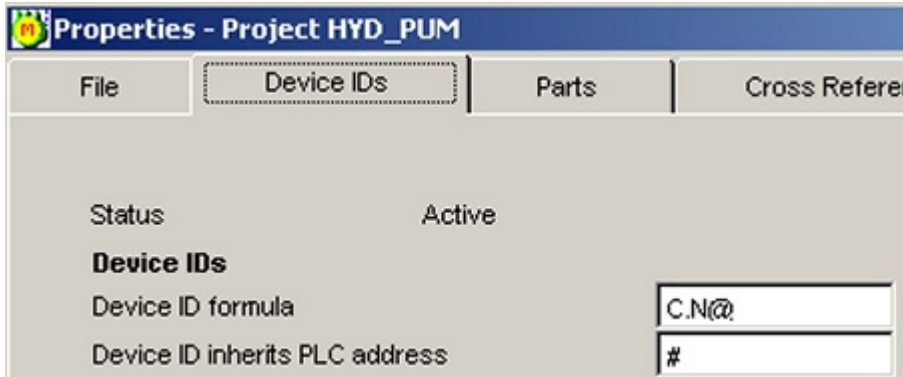
<i>Cadelec Object</i>	<i>Attribute name</i>	<i>Cadelec command</i>
Terminal strip identifier	PREFIX	ctic
Installation	LOCATION	cloc
Wiring Zone	ZONE	czon
Circuit	CIRCUIT	ccir
Cross Reference Function	XR_FUNCTION	cxrf
Cable	CC01, CC02, T_CA, F_CA	ccab

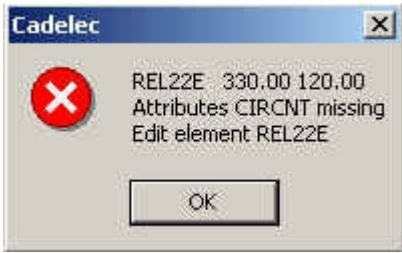
The working of this group of commands will be described in detail with the *Find and Replace Circuit command*. Although working differently internally and having different controls, the user interface is similar.

Find and Replace Circuit

Cadelec command `ccir`

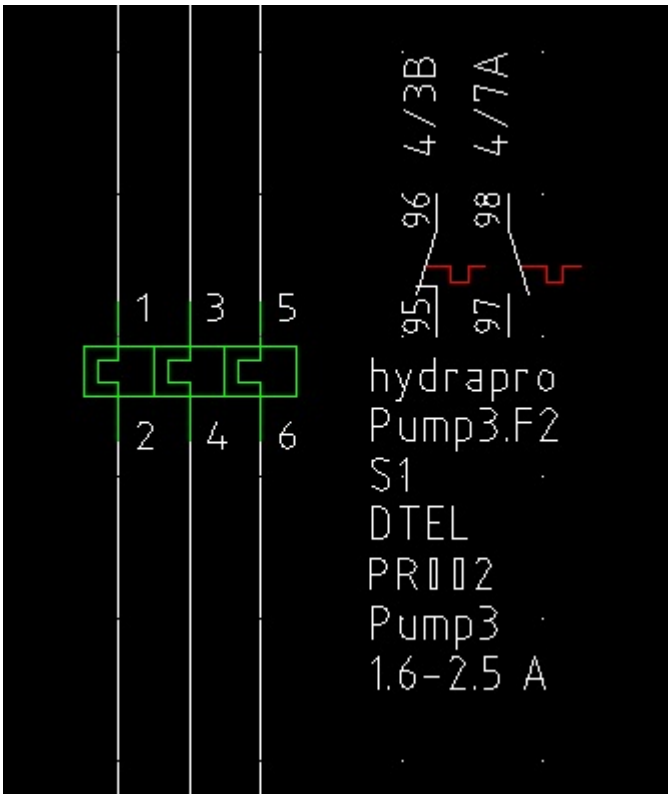
The following explanation and example will deal with the most complicated combination, when the device numbering formula contains a *Circuit Counter* (Symbol: @ in the Device ID formula);



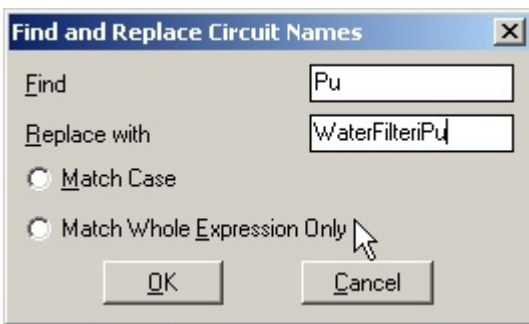


When changing the Circuit name, the circuit counter must be updated to reflect the new circuit. Cadelec will automatically renumber the object(s) whose circuit has been changed.

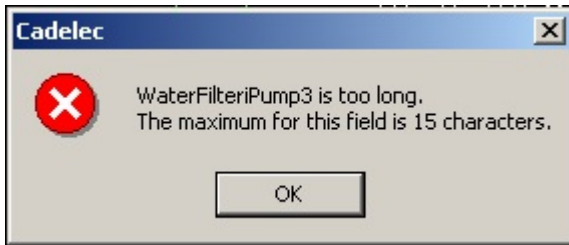
Cadelec 2006 checks that the block(s) have both the `CIRCUIT` and the `CIRCNT` attributes. If not, an error message is displayed



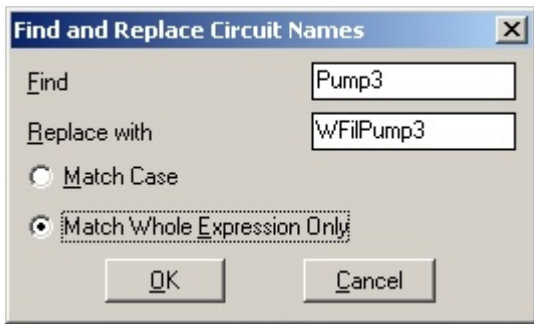
This picture shows the block with the previous Circuit name `Pump3`.



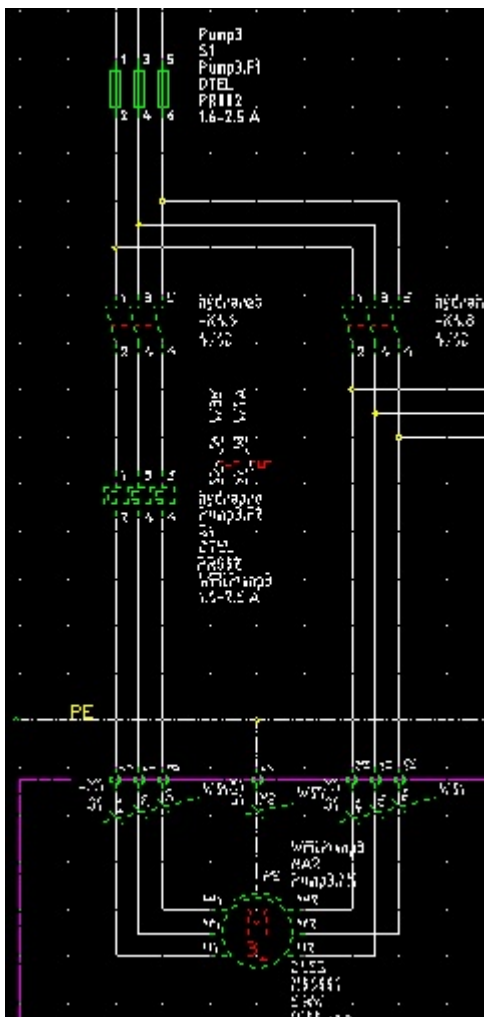
We want to change `Pump3` to `WaterFilterPump3`. The resulting string is too long (longer than the database field).



Cadelec will display an Error Message

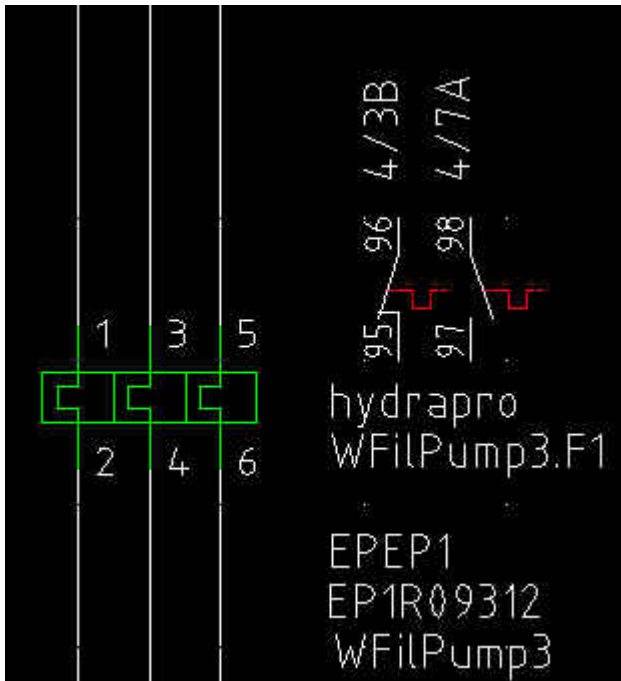


We shall therefore use the new circuit name [WFilPump3](#)



We then set a selection over the objects we want to attribute to the circuit. Note that the fuse at the top is not part of the selection.

Some elements are not numbered by circuit, such as terminals. Cadelec does not include in the selection set. Other elements, such as contacts, get their information from the master (coil) and these, too, are automatically excluded from the selection set.



The component now displays the new circuit name [WFilPum3](#). The Device ID has been automatically updated and is now [WFilPum3.F1](#)