

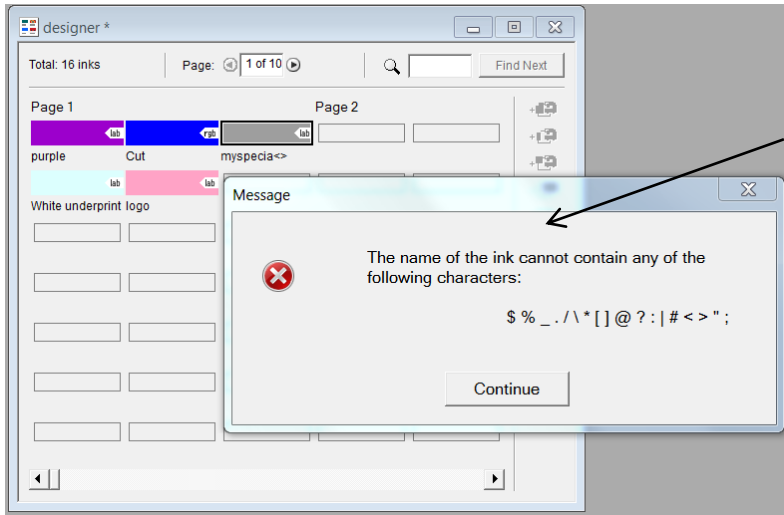
Color Engine Name Limitations in Suite 12.0 and Before

Contents

- 1. Color Engine Name Limitations Prior to Suite 12.1 (Assembly 3)3
- 2. No Name Limitations in Color Engine 12.15
- 3. Using Normalized PDF from 12.1 Applications in Older Versions6
- 4. Connecting Older Applications to a Color Engine 12.1 Database.....8
- 5. Color Engine 12.1 and DFEs10
- 6. Recommendations.....11



1. Color Engine Name Limitations Prior to Suite 12.1 (Assembly 3)



Color Engine 12.0 gives an error message when giving inks a name with special characters.

In Color Engine versions prior to 12.1 (released with Suite 12 assembly 3) it was not possible to create inks with names that contained certain characters (referred to as “special characters” in the rest of the document). The same was true for ink books, profiles, color strategies, device links, ink limits, etc.

More specifically, these were the limitations:

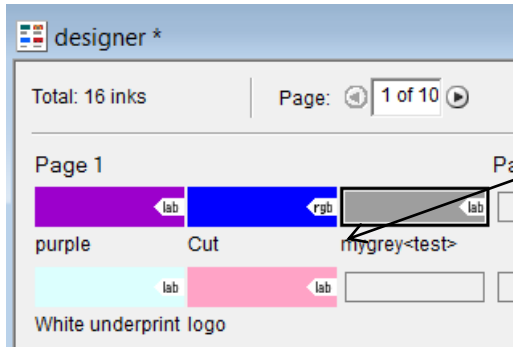
Color Engine Object	Characters that could not be used in names
Inks	<code>\$ % _ . \ ^ * [] @ ? : # < > ;</code>
Profiles, color strategies, ink books, device links, ink limits	<code>\$ % . \ ^ * [] @ & ? : # < > ;</code>

Remark that in ink names you can't use underscores but you can use space and ampersand. In all other objects you can't use space and ampersand but you can use underscores.

The limitations in ink names caused interoperability problems with desktop applications like Adobe Creative Suite. An Illustrator user can create files that use any character in the name of a spot color. This was the reason to eliminate these limitations in Color Engine 12.1.



2. No Name Limitations since Color Engine 12.1



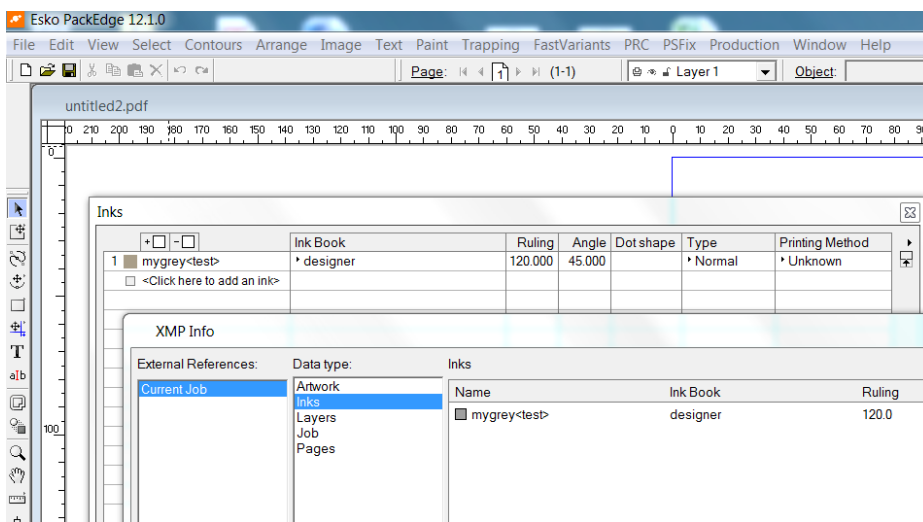
Ink with special characters in Color Engine 12.1.

In Color Engine 12.1 all limitations on names of color objects (inks, ink books, color strategies, etc..) have been removed. It is now possible to use all characters.

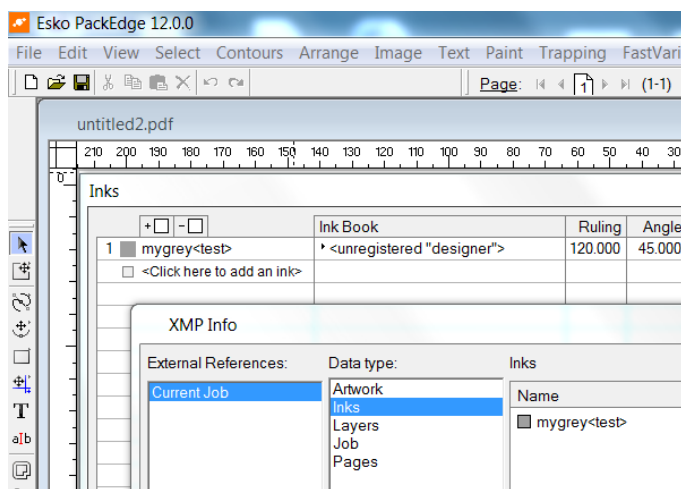
Esko applications in Suite 12.1 (Assembly 3) are capable of handling Color Engine color objects with such names. However this is not the case for older applications. So what happens when older applications connect to a Color Engine database that contain inks with names that contain characters that are not allowed in that version of the software? What happens when an application loads a file made in a newer version and that file contains ink names with characters that were not allowed before?

3. Using Normalized PDF from 12.1 Applications in Older Versions

Normalized PDFs written by 12.1 applications in Suite 12.1 (Assembly 3) can be processed in older applications with some limitations.



Normalized PDF created in PackEdge 12.1



Same Normalized PDF opened in PackEdge 12.0

As you can see on the screenshots above a Normalized PDF, created in PackEdge 12.1 using an ink with special characters, can be opened correctly in PackEdge 12.0. However PackEdge 12.0 is not able to link back the ink mygrey<test> to the



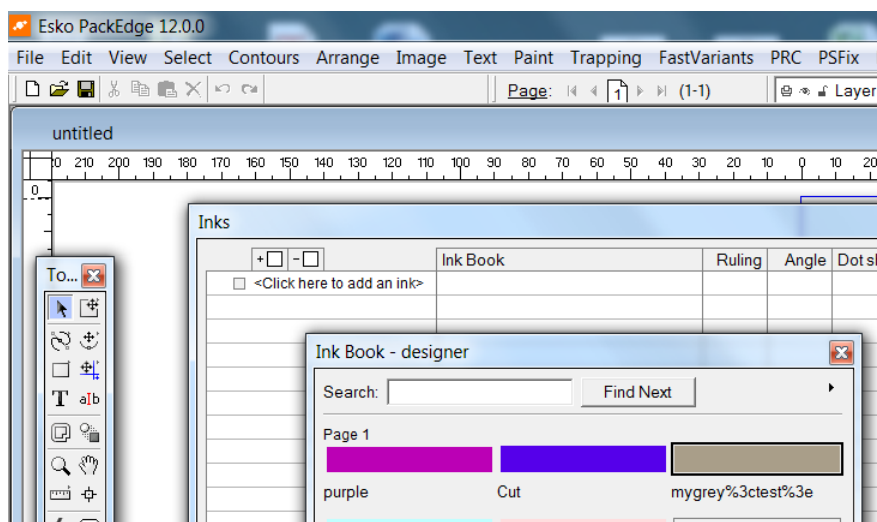
designer ink book. In other words: all inks using special characters become unregistered inks in older applications.

For example FlexProof 12.0 will use the RGB equivalents embedded in the Normalized PDF instead of the accurate color definitions in the Color Engine database when making a proof.

Accurate color management of Normalized PDFs with inks with special characters written by a 12.1 application is not possible in older versions of the Esko Suite.

4. Connecting Older Applications to a Color Engine 12.1 Database

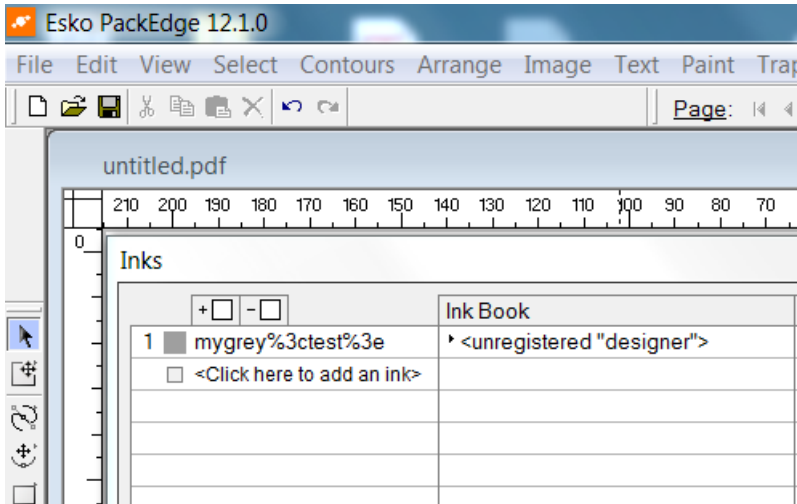
Older applications can connect to a Color Engine 12.1 database and work with inks with special characters. Again there are some limitations.



As you can see in the screenshot above older versions of PackEdge can work with inks with special characters in a Color Engine 12.1 database. The special characters are encoded in escape sequences e.g. "%3c" for the "<" character.

The older applications will display ink names with special characters in a 12.1 (or higher) database using % escape sequences. The actual ink name is obfuscated. These names will be correctly recognized by older applications. So a FlexProof 12.0 will correctly proof a Normalized PDF written by PackEdge 12.0 using the color definitions in Color Engine 12.1 (or higher) database.

Accurate color management of Normalized PDFs with inks with special characters written by a 12.0 application is possible in 12.0 versions of Esko applications.



However, when you use an ink with special characters from a Color Engine 12.1 (or higher) database in a 12.0 application and write a Normalized PDF, this PDF contains the obfuscated ink names e.g. "mygrey%3ctest%3e" instead of the real ink name e.g. "mygrey<test>". As a consequence applications since 12.1 will see this as an unregistered ink.

Accurate color management of Normalized PDFs with inks with special characters written by a 12.0 application is not possible in versions of Esko applications since 12.1.

5. Color Engine 12.1 and DFEs

None of the currently released DFEs by Esko to OEMs supports the usage of inks with special characters.



6. Recommendations

- Start using ink names with special characters in the workflow after upgrading all applications to 12.1 or higher.
- Avoid usage of older Esko applications connected to a Color Engine 12.1 (or higher) database when possible. When that is not possible avoid usage of ink names with special characters for jobs that need to be processed with these older applications.
- Avoid usage of ink names with special characters for all jobs that need to be sent to an Esko OEM DFE.
- Normalized PDFs from 12.1 (or higher) applications can be exchanged with other sites using older versions of applications as long as accurate color management is not required in the other site.