

Comment générer un fichier trx par Form, en une seule fois :

Il est facile de générer un fichier trx pour toutes les Forms d'une application, ou pour une seule Form, mais comment générer un fichier trx par Form ?

1. Solution pour U8, comme pour U9 :

Il faut utiliser icomp comme indiqué dans le tableau ci-dessous :

Export object in a trx file. Function 192 (common parameters)		
<i>Register</i>	<i>Description</i>	<i>Values</i>
\$99	Function code	192
\$98	Target File name	
\$97	Supersede	0 (no) 1 (yes)

Depending of the object you will export the registers you will need are:

Object	\$92	\$93	\$94	\$95	\$96
Component ¹				Component name	16
Start Up shell				Application name	18
Model				Model name	38
Library				Library name	58
Global registers			""	Library name	50
Global procs	"P"	Proc name	Library name	"P"	26
Includes	"P"	Include name ²	Library name	"I"	26
Messages	Language	Text name	Library name	"M"	26
Panels	"C"	Panel name	Library name	"C"	26
Devices T.T.	"T"	Device name	Library name	"D"	26
Keyboard T.T.	"T"	Keyboard name	Library name	"T"	26
Menus	Language	Menu name	Library name		88

Glyph	Language	Library name	Glyph name	"I"	92
Entity I. Template				EIT name	42
Field I. Template				FIT name	44
Field S. Template				FST name	46
Field L. Template				FLT name	48
Field Template				FT name	80
Signatures		Subsystem name (" works for all subsystems)	Signature name	"SYSTEM_LIBRARY"	100
Constants	"P"	"DEFPARAM"	Library name	"I"	26

¹ *To complete the component export operation you should do after the icomp :*

```

$96 = 26
$94 = <componentname>
$95 = "I"
93 = "!DEFPARAM" ; ! is <gold>!
$92 = "P"
perform "icomp"

```

² *Specify &!DEFPARAMS in the include name profile (&! Are <gold>&<gold>!*

2. En V9, nous pouvons utiliser \$ude export :

Syntax

[\\$ude](#) ("export", *ObjectType* , *ObjectProfile* ,{*ZipArchive* :}*Filename* {, *OptionList*})

→ Export & trx file

→ Original article from Community One