

公式

公式数据源

命令：Data source (数据源) > Formula (公式) > Add (添加)

Formula (公式) 数据源包含已创建数据源的列表。这些数据源由运算符、常量、数据源、控制变量、公式和函数填充。数据可能是数字或字母数字。

要在文档中执行计算，您必须先创建公式数据源。

此数据源具有特定对话框，您可在该对话框中为给定公式定义所需的函数。

关于函数

函数是使用叫做自变量

函数用于操作数字、字符串或逻辑值、计算或操作的结果。

公式定义中有 6 组函数公式：

- 检查字符计算函数
- 转换函数
- 日期和时间函数
- [逻辑函数](#)
- [数学函数](#)
- [字符串函数](#)

运算符

此程式包括数学、比较、级联和逻辑运算符。

算术运算符

运算符	目的
*	将两个数字相乘
+	将两个数字相加
-	从一个数字中减去另一个数字，或为运算对象指定一个负值
/	两个数字相除
^	指数幂的一个乘方数
%	模数

比较运算符

运算符	目的
<>	小于
<=	小于或等于
>	大于
>=	大于或等于
=	等于
	差异

级联运算符

用于组合两个字符串

&	两个字符串级联
---	---------

逻辑运算符

(同样参见逻辑函数)

!	非逻辑
---	-----

数学函数

以数值执行运算并得出数值结果。值可以是变量或常量。

int (值) 返回小于或等于值自变量的最大整数。

示例 :

$\text{int}(-5.863) = -6$

$\text{int}(5.863) = 5$

mod (val_1,val_2) 返回 val_1 自变量除以 val_2 自变量所得的余数。结果与除数有同样的符号。

示例 :

$\text{mod}(7,2) = 1$

$\text{mod}(-7,2) = 1$

$\text{mod}(7,-2) = -1$

$\text{mod}(-7,-2) = -1$

quotient (val_1,val_2) 返回 val_1 自变量除以 val_2 自变量的整数结果。

round (val_1,val_2) 返回自变量 val_1 , 该自变量四舍五入到 val_2 显示的位数。

如果 val_2 大于 0 , val_1 则四舍五入到说明的小数位数。

如果 val_2 等于 0 , val_1 则四舍五入到最接近的整数。

如果 val_2 小于 0 , val_1 则四舍五入到小数点左侧。

示例 :

$\text{round}(4.25,1) = 4.3$

$\text{round}(1.449,1) = 1.4$

$\text{round}(42.6,-1) = 40$

trunc (值) 返回值自变量的整数部分。

base10tobaseX(string_1,string_2) 将 string_2 从基数 10 转换为基数 string_1

示例

如果名为基数 16 的栏位含有字符串 "0123456789ABCDEF"

BASE10TOBASEX(Base16,12) 生成 C

BASE10TOBASEX(Base16,10) 生成 A

BASE10TOBASEX("012345","9") 生成 13

注意 : 此公式中字符串2参数不能为负数

baseXtobase10(string_1,string_2) 将 string_2 从基数 string_1 转换为基数 10

示例

如果名为 16 的栏位包含字符串 "0123456789ABCDEF"

BASEXTOBASE10(Base16,"E") 生成 14

BASEXTOBASE10(Base16,10) 生成 A

BASEXTOBASE10("012345","9") 生成 13

hex(val_1,val_2) 利用 val_ 总值将 val_1 十进制数字转换为十六进制格式。

注意 : 此公式中val_1参数不能为负数

逻辑函数

逻辑函数使您能检查是否符合一个或多个条件。

注意 : TRUE 等于 1 , FALSE 等于 0

如果两个自变量都为真 , 则 **and** (expr_1,expr_2) 返回 TRUE , 如果至少一个为假 , 则返回

FALSE。必须从逻辑值中计算自变量。

示例：

```
and(exact("string","string"),exact("string","string")) = 0
```

```
and(exact("string","string"),exact("string","string")) = 1
```

如果两个字符串相同，`exact (string_1, string_2)` 返回 TRUE，否则返回 FALSE。此函数区分大小写。

示例：

```
exact("software","software") = 1
```

```
exact("software","software") = 0
```

`if (expr,Val_if_true,Val_if_false)` 返回 Val_if_true 值，如果 Expr 为真；如果 Expr 为假，则返回 Val_if_false 自变量。

示例：

```
if(exact("string","string"),"true","false") = false
```

```
if(exact("string","string"),"true","false") = true
```

`not (logical)` 提供逻辑自变量的对立项。

示例： `not(exact("string","string")) = 1`

```
not(exact("string","string")) = 0
```

```
not(False) = 1 或 not(0) = 1
```

```
not(True) = 0 或 not(1) = 0
```

```
not(1+1=2) = 0
```

如果两个自变量中的一个为真，`or (expr_1,expr_2)` 将返回 TRUE，如果两个自变量均为假，则返回 FALSE。必须从逻辑值中计算自变量。

示例：

```
or(exact("string","string"),exact("string","string")) = 0
```

`or(exact("string","string"),exact("string","string")) = 1`

`or(true,true) = 1` 或 `or(1,1) = 1`

`or(true,false) = 1` 或 `or(1,0) = 1`

`or(false,false)= 0` 或 `or(0,0) = 0`

文本函数

如果每个框中都含有一个字符，字符串可以被同化到表格中。它由自身长度设定（字符串中字符的总数，包括空格。）字符串中字符的位置与表格中字符的位置相对应，即：第一个字符在位置 1 上。

示例：位置 3 对应于字符串中的第三个字符。

`cyclebasex ()`

可以在任何类型的数据库计数系统中进行计数。必须在链接的表达式中设定编号系统。还必须相互指定起始值、每个增量的值以及副本数量。所有这些值都可以链接于标签中的其他栏位，但不能使用引号将栏位名括起。

示例：

如果名为基数 16 的栏位包含字符串 0123456789ABCDEF，则：

`cyclebasex(base16,"8",1,1) = 8,9,A,B,C`

`cyclebasex(base16,"F",-1,1) = F,E,D,C,B,A 9,8,7`

`cyclebasex(base16,"B0 ",1,1) = B0,B1,B2`

`cyclebasex("012345","4",1,2) = 4,4,5,5,10,10,11,11;`

`cyclechar ()` 为完整的循环创建一组的用户设定字符。

示例：

`cyclechar("A","C") = A B C A B C A B C`

`cyclechar("A","C",1,2) = A A B B C C A A B B`

cyclenumber () 可以使用您自己设置的数字顺序，而无需使用数字或字母的正常顺序 (0,1,2 或 A,B,C)。

示例：

`cyclenumber(1,3)` 将会产生以下顺序的标签：1 2 3 1 2 3 1 2 3...

`cyclenumber(1,3,1,2)` 将会产生以下顺序的标签：1 1 2 2 3 3 2 2 1 1...

cyclestring () 可以使用完整的循环将一组字或字符创建为增量栏位。整个字符串必须使用引号 (")括起，并且必须使用分号 (;) 将每个字或每组字符与其他字或字符组隔开。

示例：

`cyclestring("Mon ; Tue ; Wed ; Thu ; Fri ; Sat ; Sun")` = Mon Tue Wed Thu Fri Sat Sun

以下示例用于使用除 O 和 I 以外所有字母的标签。

`cyclestring("A ; B ; C ; D ; E ; F ; G ; H ; J ; K ; L ; M ; N ; P ; Q ; R ; ST ; U ; V ; W ; X ; Y ; Z")`

如果两个字符串相同，`exact (string_1, string_2)` 返回 TRUE，否则返回 FALSE。

示例：

`exact("software","software")` = 1

`exact("sftware","software")` = 0

extract(«string», «sep», «pos») 从字符串 «string» 中返回处于指定位置 «pos» 并由字符串 «sep» 分隔的数据构成的子字符串。

例如：

`extract("1;2;3;4", ";", 3)` = 3

find (字符串,按键,起始开始) 返回自变量字符串中第一个按键自变量产生的位置。字符串自变量中的搜索是从起始自变量 (起始 ≥ 1) 返回的位置开始的。如果没有产生按键自变量,函数将复位为 0。此函数区分字母大小写。

示例 :

```
find("Peter McPeepert","P",1) = 1
```

```
find("Peter McPeepert","p",1) = 12
```

```
A0;
```

left (string,num_char) 返回从字符串自变量中提取的字符串。此字符串在字符串自变量的位置 1 处开始,其长度与 num_char 自变量相等。

示例 :

```
left("Peter McPeepert",1) = P
```

```
left("Peter McPeepert ",5) = Peter
```

len (字符串) 提供字符串自变量的长度。空格作为字符计算。

示例 :

```
len("Paris,New York") = 15
```

```
len("") = 0
```

```
len(" ") = 1
```

lower (字符串) 将文本字符串中的所有大写字母转换为小写字母。

示例 :

```
lower("Paris,New York") = paris,new york
```

mid (字符串,num_char) 返回从字符串自变量中提取的字符串。此字符串从起始自变量 (起始 ≥ 1) 的值相对应的位置开始,其长度等于 num_char 自变量的长度。

示例：

```
mid("Paris,New York",8,8) = New York
```

pad () 向栏位左侧添加字符，从而指定整个输入的预定长度。任何字符都可以选作填充字符。

示例：

如果名为 GREETING 的栏位显示值 HELLO，则：

```
pad(GREETING,8) = 000HELLO
```

```
pad(5,3,0) = 005
```

```
pad("Nine",6,"a") = aaNine
```

replace (字符串,起始,num_char,new_string) 返回转换的字符串自变量。来自在起始自变量中设定的位置上的字符数 (等于 num_char 自变量) 已被 new_string 自变量替代。

示例：

```
replace("Paris,New York",8,8,"Singapore") = Paris,Singapore
```

replacestring(«string», «old_string», «new_string») 使用指定的 «new_string» 在字符串 «string» 中替换另外指定的所有 «old_string»。

例如：

```
replacestring( "ABC12DEF12", "12", "" ) = ABCDEF
```

rept (字符串,num_char) 给出一个字符串自变量重复num_char自变量数值的字符串。

示例：

```
rept("Ah Paris!",2) = Ah Paris!Ah Paris!
```

right (字符串,num_char) 提供构成字符串最后字符的字符串，其长度等于 num_char 自变量。

示例：

```
right("Purchase order",5) = order
```

search (字符串,按键,起始) 提供字符串自变量中第一次产生按键自变量的位置。搜索从起始自变量 (起始 >= 1) 设定的位置开始。如果没有产生按键自变量,函数将复位为 0。

示例：

```
search("Purchase order","order",1) = 10
```

```
search("Purchase order","c",1) = 4
```

trim (字符串) 返回转换的字符串自变量。删除字符串起始和末尾处的所有空格。两个字之间的空格数减为 1 个。

示例：

```
trim(" Purchase order") = Purchase order
```

trimall (字符串) 返回转换的字符串自变量。删除所有空格。

示例：

```
trimall("Paris / New York / Rome") = Paris/NewYork/Rome
```

upper (字符串) 返回转换的为大写字母的字符串。

示例：

```
upper("Purchase order") = PURCHASE ORDER
```

ztrim () 在完全为数字的栏位中删除从左侧开始出现的所有 0。

示例：

如果名为 WEIGHT 的栏位显示值 000200, 则：

```
ztrim(weight) = 200
```

定义公式数据源的属性

命令：Data source (数据源) > Formula (公式) > Properties (属性)。

1. 在 Edit (编辑) 框中直接输入公式。

-或者-

选择所需元素，然后单击 Insert (插入)。

2. 单击

Test (测试) 以验证语法是否正确。如果出现错误，请按照屏幕上的说明操作，然后执行所有必需的更改。

3. 单击 OK (确定)。

提示：可通过双击某个元素来插入该元素。

注意：如果公式中使用的某个变量具有包含 &+*/<>=^%,\|" 之中某个字符的名称，则必须使用括号 {} 将其引起来。

注意：通过单击

Test (测试) 可以检查您的公式。如果消息显示公式值，则表示公式正确。如果值不正确，则请按照屏幕上的说明执行必需的修改。如果获得的值被截断，则必须在修改

Output (输出) 选项卡中指定的最大长度。

实践: 创建一个简单的公式

显示产品价格

产品标签必须以重量和每千克价格的函数显示产品价格。

1. 打开标签。必须创建两个变量：WEIGHT 和 PRICEPERKG.
2. 对于 WEIGHT 变量：输入 788 (产品重量为 788 克) 作为本地值，然后在 Prefix (前缀) 框中输入 "Please enter the weight in g"，然后单击 OK (确定)
3. 对于 PRICEPERKG 变量：输入 15.70 (每千克价格为 FF15.70) 作为该变量的本地值，然后在 Prefix (前缀) 框中输入 "Please enter the price per kg"，然后单击 OK (确定)
4. 添加公式，并将其命名为 price。
5. 输入公式 $WEIGHT*PRICEPERKG/1000$ ，然后单击 OK (确定)。
6. 保存此文档

演示：添加 "Warning" 公式变量以显示警告消息

在以下序列中，我们将创建一个公式来显示警告消息，告知用户 Total_Weight 共享变量的值超过 1000 千克。

如果重量值超过 1,000 千克，则显示 "Attention!Error!Total Weight exceeds maximum!" 消息。

1. 打开标签。
2. 创建一个公式，并将其命名为 "Warning"。
3. 在 Formula (公式) 对话框中，输入以下表达式：
`if(Total_Weight>1000, "注意！错误！总重量超过最大值", "")..`
4. 在 Output (输出) 选项卡中，在 Maximum length (最大长度) 中输入 50，然后单击 OK (确定)。
5. 将变量作为文本放置在标签中。
6. 在 Text (文本) 对话框，选择 Scalable (可缩放) 作为字体，将 Height (高度) 设置为 12.70 毫米。
7. 在 Paragraph (段落) 选项卡中，选中 Wordwrap (自动换行) 选项，然后在对齐方式中选中 Centered (居中)。

有关 IF 函数的信息

如果指定的条件为 TRUE，返回一个值，如果指定的条件为 FALSE，则返回另一个值。

使用 IF 函数对值和公式执行条件测试。

句法

if("expr","val_if_true","val_if_false") "expr" 代表任何数值或者表达式, 其结果要么为 真, 要不为 假.

val_if_true 是当表达式"expr" 为 真 时返回的数值. val_if_true

也可以为另一个公式, 从而实现公式套用.

val_if_false是当表达式"expr" 为 假 时返回的数值. val_if_true

也可以为另一个公式, 从而实现公式套用.

实践 - 计算特殊" 模数"

在此练习中我们将 EAN8 条形码"Customer_Code"转换为一个 2/5 Interleaved 条形码, 我们使用公式'Formula_4_NewCustCode'来完成这个转换.

条形码有如下的一些属性:

- 符号体系: 打印机,
- 高度: 4毫米,
- 窄条宽度: 1毫米,
- 比率: 2,
- 人工识别符: 下面 / 居中,
- 与条形码之间的距离: 0毫米,
- 字符字体: 打印机字体..

1. 打开标签文件 ORDER_WS2.LAB .

计算重量

创建一个公式并命名为 Formula_1_Weighted. 计算规则为：变量 Customer_Code 的第一个字符乘以1, 第二个乘以2, 第三个乘以1, 第四个乘以2, 等等.

变量的最大输出长度为 6.

Formula_1_Weighted:

```
mid(Customer_Code,1,1)*1&mid(Customer_Code,2,1)*2&mid(Customer_Code,3,1)  
1&mid(Customer_Code,4,1)*2
```

将计算出的重量结果加起来:

接下来的一步是将前面公式中得到的结果加起来. 最大的输入长度为2.

创建第二个公式并命名为 Formula_2_Sum.

计算校验位:

利用前面的结果，我们来计算校验位的值.

创建第三个公式并命名为 Formula_3_CheckDigit.

表达式如下:

```
if ((Formula_2_Sum % 10) > 0,10 - Formula_2_Sum % 10,0)
```

将原来条码数值与校验位组合起来:

当创建条形码的时候必须包括原有数值和校验位 (Formula_3_CheckDigit).

创建第四个公式并且命名为 Formula_4_NewCustCode. 此公式将变量 Customer_Code 与校验码 Formula_3_CheckDigit 串接起来.

创建条形码:

1. 选择公式 Formula_4_NewCustCode 并将它拖到设计区域 Customer_Code 条形码的位置.
2. 设置条形码的属性.

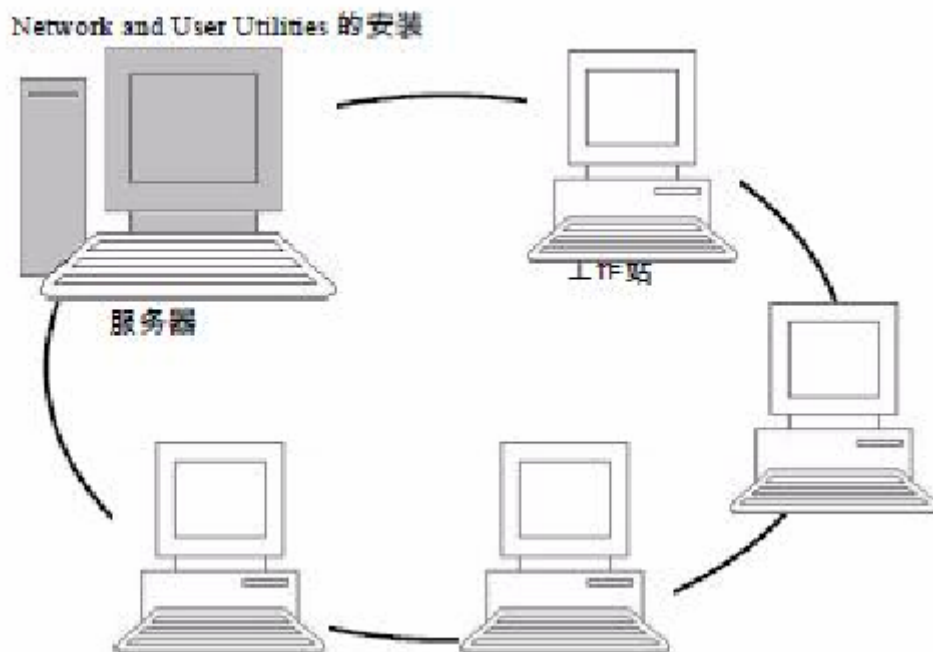
安装网络版

功能介绍

利用网络（多用户）软件包，您可以通过网络控制对标签设计软件许可证的访问权限。

使用此实用程序，您可以让很多用户同时从网络中的任意位置访问标签设计软件。

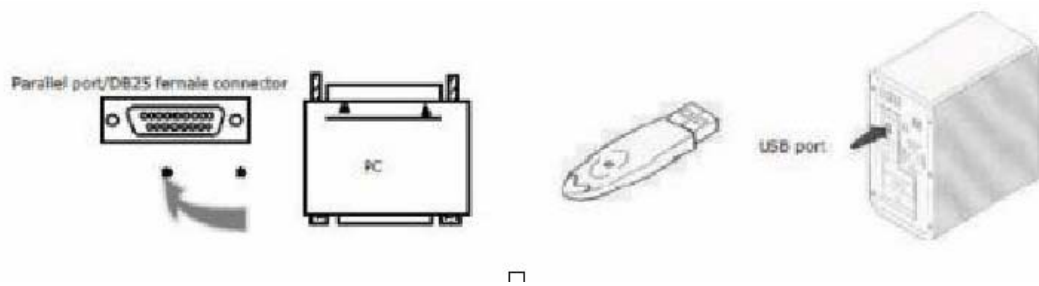
若要使用此标签软体的网路/多重使用者版，您必须在伺服器或作为伺服器的工作站上安装Network and User Utilities，然后在每个工作站上，安装标签设定软体。



安装加密狗

加密狗必须安装和License Manager 安装在同一台电脑上. 加密狗中包含能够同时使用的License数量.

在运行License manager 之前，必须先安装好加密狗.



注意: 当您启动程序时, 如果加密狗与产品版本不匹配, 将会自动跳出一个显示必要信息的提示框..如果您使用的是并口打印机端口, 并将此端口连接至密码狗上. 这种情况下, 您可能需要启动打印机来使得加密狗被电脑正确认出.

安装程序

网络配置

安装软件之前, 网络管理员必须先为用户组定义网络结构, 特别是:

- 定义要在其中安装 Network and Users Utilities 及加密狗的许可证服务器。
- 定义要使用标签设计软件的工作站或客户端工作站

Network Manager 介绍

Network and Users Utilities 可让您使用标签设计软件的网络 配置。Network Manager 包括:

- License Manager (**License Service**)
- **Network Settings Wizard**: **Network Settings Wizard** 可帮助您定义网络配置。
- **User Manager**: **User Manager** 装有 **Network and Users Utilities**, 因此您可以在网络设置中定义标签设计软件的 访问权限。

安装 Network and Users Utilities

在将使用标签设计软件的所有工作站上安装此软件之前，必须先在服务器上安装 License Service 实用程序，以对网络进行配置。

在服务器上安装 Network and Users Utilities。

1. 将安装 CD 插入适当的驱动器中。
显示安装窗口。
如果 CD-ROM 光盘不能自动运行：
 2. 选择开始> 运行，然后键入 CD-ROM 光驱号和 index.hta (例如：键入 D:\index.hta)。
 3. 选择 Network and Users Utilities，此实用程序包含 License Manager 和 User Manager。
然后单击安装按钮。
 4. 按照屏幕上的说明操作
 5. 以TKDONGLE 作为共用名称，共用可完整控制的[TKDONGLE]
资料夹。此资料夹的预设存取路径是C:\Documents and
Settings\AllUsers\ApplicationData\TKI\LicenseManager\TKDongle (Vista
上为：C:\ProgramData\TKI\LicenseManager\TKDongle) > 按一下滑鼠右键> 内容>
共用标签页以及权限按钮

对于管理员：

必须通过以下方式，向需要网络许可证写入访问权限的用户授予此权限：

1. 共享 TKDongle 文件夹并授权用户：C:\Documents and Settings\All
Users\ApplicationData\TKI\LicenseManager\TKDongle (Vista:C:\Program
Data\TKI\LicenseManager\TKDongle) > 右键单击 > 属性 > 共享选项卡和权限按钮。
2. 在 TKDongle 属性的安全选项卡中，给予用户写入访问权限

配置

通过选择开始 > 程序 > Teklynx > Network and Users Utilities，然后选择网络工具栏，可从网络工具栏中获得用来配置网络版本的所有必备工具。

Network Settings Wizard 可帮助您定义网络版本的设置。

1. 要启动 Network Settings Wizard，请单击图标。
2. 在向导的步骤 1 中，选择一个设置模式：普通、根据用户或根据工作站。
 - 普通：所有用户将在所有工作站上使用相同的设置。(user.ini)
 - 根据用户：每个用户可以在任何工作站上访问各自的设置。(user name.ini)
 - 根据工作站：每个工作站具有各自的设置(station.ini)
3. 在步骤 2 中，指定您要存储这些设置的位置。如果要在各个工作站之间共享这些设置，应指定所有工作站都可以访问的网络路径。（示例：TKDongle）。
4. 在步骤 3 中，指定您要存储共享数据（变量、列表、打印日志文件等）的位置。

请确保所有用户对这些文件夹都具有正确的访问权限。

配置 User Manager

如果您要为标签设计软件的所有用户定义网络访问权限，必须在安装过程中进行定义（请参考 User Manager 帮助系统）。

- 单击网络工具栏上的 User Manager 图标。

启动 License Manager


必须先确保已启动 License Manager，然后才能在所有工作站上安装标签设计软件。

License Manager 已安装为服务。

您无需启动它。实际上，服务会在工作站启动时启动，而且只要工作站处于打开状态，它就会作为一项后台任务运行。

如果将网络服务器安装为受软件密钥保护，则您必须对许可证进行激活，然后它才会自动启动。

启动服务控制器

- 单击网络工具栏上的图标双击 SLICENSECTRL.EXE 文件。(C:\PROGRAM FILES\TEKLYNX\NETWORK\ TOOLS\DONGLE)

注意：如果您想让WINDOWS开机自动运行此程序,将其快捷方式复制到WINDOWS 开始>程序>启动组即可

在工作站上安装软件

必须将标签设计软件安装在要使用该软件的所有工作站上。

在工作站上安装此软件

1. 将安装 CD 插入适当的驱动器中。

将显示安装窗口。

如果 CD 不能自动运行：

2. 请选择开始 > 运行，然后键入 CD 驱动器号和 INDEX.HTA (例如：D:\index.hta)。
3. 选择要安装的产品，然后单击安装按钮，并按照屏幕上的说明进行操作。
4. 启动标签设计软件。将会出现一条消息，通知您没有找到 加密狗。单击是启动软件。
5. 从工具菜单中，选择网络管理。
6. 启用使用网络许可证。
7. 单击修改，选择安装了 License Manager 和加密狗的服务器。



单击浏览，自动搜索已安装 License Manager 的服务器。

如果已经配置了网络设置，则会显示一条消息，询问您是否要使用当前的网络配置

8. 如果您要修改或配置网络设置，请单击 Network Settings Wizard 按钮。
9. 单击确定。
10. 重新启动该程序。

如果更改了服务器，您将需要更新所有工作站。在这种情况下，请启动标签设计软件并选择工具 > 网络管理。禁用并重新启用使用网络许可证选项。