Power Form 用户指南

2.0 版

本指南中包含的信息不具有合同性质，如有更改，恕不另行通知。

本指南中介绍的软件按许可协议销售。软件只能按照该协议的条款进行使用、复制或翻录。

未经 TEKLYNX Corporation SAS
的书面授权，不得以任何形式复制、翻录或传播本指南的任何部分，或用于购买人个人使用以外的任何用途。

© 2026 TEKLYNX Corporation SAS,
保留所有权利。

目录

介绍 4
布局 4
菜单栏 4
工具栏 5
解决方案树 7
属性网格 8
主窗口 9
状态栏 9
基础知识 10
什么是行动？ 10
脚本操作 15
故障排除 23
使用表单变量 24
更新对象动作 24

介绍

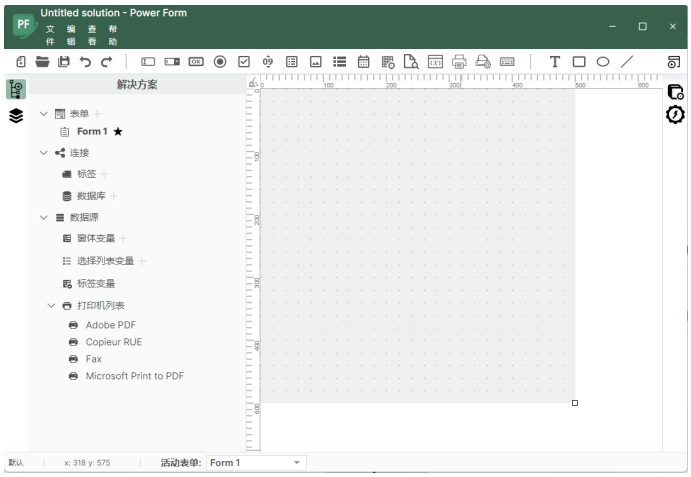
Power Form 是一款可视化设计工具，可让您快速构建前端数据输入和标签打印应用程序。

它包含以下功能：

- 标签和打印机类型选择
- 打印预览
- 页面和打印机设置
- 数据库连接
- 多屏幕/表单应用程序
- 预定义值和数据控制

布局

Power Form 的布局简单易用，减少了用户熟悉软件所需的时间。它包括菜单栏、工具栏、解决方案树、属性网格、主窗口和状态栏。



菜单栏

菜单栏包含 4 个项目：文件、编辑、查看和帮助。点击后，每个项目都会展开并显示一个新的项目列表，让用户能够执行各种功能。要更快地访问菜单项，请按 Alt + 所需选项中带下划线的字母：Alt+F 将打开“文件”菜单。



工具栏

工具栏通常显示在菜单栏的正下方。它由图标组成，代表操作表单时最常用的软件功能和对象。点击这些图标即可快速访问这些功能。

表单中可添加的对象包括 16 个控件和 4 种形状：文本输入框、组合框、按钮、单选按钮、复选框、数值上下输入框、分组框、图片框、列表框、日期选择器、数据表网格、标签预览、标签变量数据网格、打印机选择器、打印设置、文本、矩形、椭圆和线条。

每个控件都有其特定用途。它们可以关联固定数据或可变数据，并且可以与操作关联。要将控件插入表单，只需单击该控件并将其拖放至表单中即可。

控制	**描述**
文本输入框	数据录入字段
组合框	在下拉列表中显示在属性中定义的数据
按钮	用于执行预设功能/操作的活动控件
单选按钮	用一组单选按钮显示数据列表。
复选框	用于激活/停用或选择/取消选择预定义功能的控件
数字向上/向下	仅用于输入数值的数据字段
分组框	为了便于视觉组织，在控件组周围放置了一个非活动框。

图片框	可以导入图像的框
列表框	在列表中显示在属性中定义的数据
日期选择框	数据输入字段，用于输入其属性中定义的日期格式。
数据表网格	显示属性中定义的数据库表，以及一些基本功能，例如筛选器和搜索字段。
标签预览	在框架中显示附加到表单的标签预览
标签变量数据网格	视图中以“打印时输入”标签变量的形式显示名称和值
打印机选择器	允许用户选择活动打印机并更改所选打印机的设置

控制	**描述**
打印设置	允许用户打印选定的标签，并设置标签数量、标签份数、页数和切割间距参数。
虚拟键盘	用于数据输入的屏幕键盘。
文本	用于显示输入字段名称的非活动文本
矩形	显示一个矩形
椭圆	显示一个椭圆形
线条	显示线条

解决方案树

解决方案树是显示在应用程序左侧的一组标签页。它由多个部分组成，用于将通用配置应用于解决方案。



解决方案选项卡由 3 个不同的部分组成，可以配置并应用于整个表单：表单、连接和数据源。

部分	描述
表单	解决方案中用于表示不同工作区域的表单列表
连接	可用连接列表，包括 API 连接、标签连接和数据库连接。
数据来源	包含不同类型的变量 可用于解决方案中。数据源有 4 种类型：表单变量、选择列表变量、标签变量和打印机列表。

对象选项卡是放置在活动窗体上的对象列表。



属性网格

属性网格列有所选控件的所有属性。您可以在此处更改控件的属性，例如控件名称、颜色、大小或操作。它具有上下文相关性，因此属性列表会根据所选控件而改变。显示网格可取代通常的右键菜单命令。

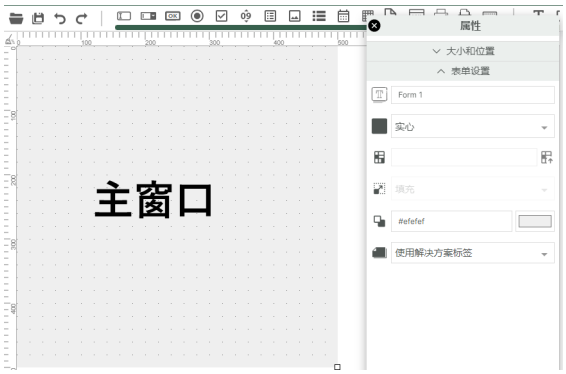


某些控件可能包含“数据类型”部分，允许用户为该控件分配变量或链接数据库。一旦分配，控件值将与选定的数据类型绑定，对控件或数据类型所做的任何更改都将同步。



主窗口

显示在中心位置的**主窗口**是您的工作区。在这里，您将添加控件并创建用于输入标签打印数据的表单。表单创建和配置完成后，您可以前往**文件** > **运行**来运行表单并查看最终结果。



状态栏

状态栏提供有关活动打印机、指针的 x 轴和 y 轴位置以及活动表单选择器下拉列表的信息。



基础知识

什么是行动？

动作是分配给按钮和加载/卸载表单的功能。在 Power Form 中，已包含 14 个预定义动作。**动作**菜单可以节省您的时间，并免去您编程的需要。在某些情况下，如果需要配置特定参数，这些参数也已在动作配置中预定义。

预定义动作包括：

动作	**确认弹出窗口**
	显示确认对话框 用户可以看到预设的消息以及接受或取消消息的按钮。
参数	**标题**
	用户可设置标题。
	信息
	用户可设置消息正文。
	按钮
	下拉菜单中提供在消息弹出窗口中显示的各类按钮。
	图标
	下拉菜单中提供在消息弹出窗口中显示的各类图标。

动作	**更新变量值**
	允许用户更新变量。
参数	**变量名称**
	用户可以选择要更新的变量名称。
	新的值类型
	下拉菜单显示可应用于待更新变量的值类型。
	选项

	允许用户为所选变量的现有值添加新值。
	新值
	允许用户为所选变量选择新值的字段。

动作	**显示表单。**
	允许用户打开新表单。
参数	**要显示的表单**
	用户可以从列表中选择要显示的表单。
	显示表单行为
	在新窗口或当前窗口中打开所选表单。
	选项
	允许用户打开之前打开过的表单（如有）。

动作	**退出表单。**
	允许用户关闭表单执行窗口。

动作	**设置打印机**
	允许用户从选定的打印机进行打印，而不使用任何与该文档关联的打印机。
参数	**设置打印机**
	允许您将打印机设置为单击按钮时要使用的默认打印机，或者使用从列表中选择的打印机——如果在表单属性中使用并定义了打印机选择变量。

动作	打印标签
	允许用户打印。
参数	标签计数
	向打印机发送要打印的标签数量。如果文档包含计数器，则计数器会随着标签数量的增加而递增。
	标签副本
	告知打印机每种标签需要打印多少份相同的副本。
	页面副本
	告知打印机每页需要打印多少份。
	切割位置
	告知打印机在每次切割前要打印多少个标签。

动作	从文件读取数据
	允许用户读取文件数据并将其值应用于变量或控件。
参数	选择文件
	允许用户选择要读取的文件。
	最大尺寸
	设置从文件中读取的最大行数。
	应用值类型
	允许用户选择将文件中数据的应用位置，可以是变量或控件。
	选择变量/控件
	允许用户选择要应用哪些控件数据。

动作	打开对话框
	允许用户打开标签设计器对话框
参数	打开对话框
	下拉菜单中列出可打开的对话框。

动作	**打开网页链接**
	允许用户打开选定的网页链接
参数	**链接地址**
	输入要打开并插入的链接。

动作	**执行 SQL 请求**
	允许用户向连接的数据库发送 SQL 请求以执行。
参数	**数据库**
	选择要执行 SQL 请求的数据库。
参数	**SQL 语句**
	SQL 代码文本区域

动作	**执行脚本请求**
	允许用户运行原生 JavaScript 代码来比较值、更新变量、将数据写入文件等。
参数	**脚本**
	代码文本区域
参数	**分类**
	下拉菜单，包含方法、常量、运算符等类别名称。
参数	**片段**
	所选类别中的代码片段列表
参数	**添加**
	按钮可将选定的代码片段添加到代码文本区域
参数	**验证**
	用于验证文本区域中输入的代码语法的按钮。

动作	刷新数据库
	允许用户强制刷新数据库
参数	从以下来源获取数据
	下拉菜单中，“数据库”选项用于强制刷新数据库；以及“从上一次操作结果”选项用于使用 SQL 操作筛选刷新数据库。

动作	打开网页链接
	允许用户打开指定的网页链接。
参数	链接地址
	该字段用于输入要打开的网页链接地址。

动作	设定焦点
	允许用户将焦点设置到选定的控件上。
参数	选择控件
	选择要获得焦点的控件。

动作	更新对象
	允许用户更新选定的控件。
参数	选择控件
	选择要更新的控件。
参数	属性列表
	显示所选控件可更新的属性列表

脚本操作

“执行脚本操作”允许用户将 JavaScript 代码绑定到 Power Form 应用程序中的 UI 元素（如按钮）。用户即可通过编写响应用户交互的脚本来扩展应用程序的逻辑。

步骤详解：

1. 选择一个用户界面元素
    1. 单击要附加脚本的 UI 元素（例如：按钮）。
    2. 提示：您可以选择按钮、输入框、网格或任何支持事件的对象。

2. 打开事件面板
    1. 点击齿轮图标 。
    2. 在“事件”面板中，找到要处理的事件，例如“点击”。
    3. 点击事件名称旁边的“...”按钮进行配置。

3. 添加脚本动作
    1. 在打开的“配置事件动作”对话框中：
        a. 点击“执行脚本动作”图标 。
        b. 列表中将出现一项新动作。

4. 编写脚本

    在“属性”面板中，输入您的自定义 JavaScript 逻辑。

    例如：

    ```
    const a = 12;
    const b = 32;
    return b * a;
    ```

    * 这段 JavaScript 代码定义了两个常量变量，`a` 赋值为 12，`b` 赋值为 32。然后返回 `b` 乘以 `a` 的结果。

5. 验证并应用
    1. 点击“验证”以检查脚本是否存在语法错误。

2. 操作成功后，按“应用”保存操作。

使用代码片段

脚本界面提供了一组按类别分类的预定义代码片段。这些代码片段可帮助用户插入常用的表达式、常量、结构和函数，而无需从头开始编写。

使用代码片段：

1. 从下拉菜单中选择一个类别。
2. 从第二个下拉菜单中选择一个片段。
3. 点击“添加”将其插入脚本编辑器。

所有可用代码片段：

Snippet	**Description**
Math	
Math.sqrt(number)	Returns the square root of a number
Math.abs(number)	Returns the absolute value
Math.cos(number)	Returns the cosine of a number (radians)
Math.sin(number)	Returns the sine of a number (radians)
Math.ceil(number)	Rounds up to the nearest integer
Math.floor(number)	Rounds down to the nearest integer
Math.round(number)	Rounds to the nearest integer
Math.random()	Returns a random number between 0 and 1
Math.pow(a, b)	Raises a to the power of b
Math.max(a, b)	Returns the maximum of two values
Math.min(a, b)	Returns the minimum of two values

Snippet	Description
Math.log(x)	Natural logarithm (base e) of x
Math.exp(x)	Returns e to the power of x
JSON	
JSON.stringify(object)	Converts an object to a JSON string
JSON.parse(jsonString)	Parses a JSON string into an object
Variables	
let x = value;	Declares a variable with block scope
const PI = 3.14;	Declares a constant
let result;	Declares an uninitialized variable
var total = 0;	Declares a function-scoped variable
Constants	
Number.MAX_VALUE	Maximum numeric value
Number.MIN_VALUE	Smallest positive numeric value
undefined	Primitive indicating no value
null	Intentional absence of any object value
NaN	Not-a-Number value
Math.PI	The ratio of circumference to diameter
Math.E	Euler's number
Math.LN2	Natural log of 2
Math.LN10	Natural log of 10

Snippet	Description
Math.LOG2E	Base 2 log of e
Math.LOG10E	Base 10 log of e
Math.SQRT1_2	Square root of 0.5
Math.SQRT2	Square root of 2
Logical	
a > b	Greater than comparison
a < b	Less than comparison
a >= b	Greater than or equal to
a <= b	Less than or equal to
a && b	Logical AND
a || b	Logical OR
!a	Logical NOT
a != b	Inequality comparison
a !== b	Strict inequality comparison
TRUE	Boolean true
FALSE	Boolean false
Operators	
a + b	Addition
a - b	Subtraction
a * b	Multiplication

Snippet	Description
a / b	Division
a % b	Modulo
a ** b	Exponentiation
a += b	Add and assign
a -= b	Subtract and assign
a *= b	Multiply and assign
a /= b	Divide and assign
a == b	Equality
a === b	Strict equality
++a	Pre-increment
--a	Pre-decrement
Statements	
if (condition) { }	Conditional execution
else { }	Executes if condition is false
else if (condition) { }	Checks another condition
for (let i = 0; i < n; i++) { }	For loop
while (condition) { }	While loop
do { } while (condition);	Do-while loop
switch (expression) { case: break;}	Switch-case conditional
try { } catch (e) { }	Try-catch block

Snippet	Description
return value;	Returns a value
break;	Breaks out of loop
continue;	Skips to next loop iteration
Array	
array.push(item)	Adds item to end of array
array.pop()	Removes last item from array
array.map(x => x)	Transforms each array element
array.filter(x => x > 0)	Filters elements based on condition
array.find(x => x === 0)	Finds first match
array.reduce(...)	Reduces array to a single value
array.includes(value)	Checks if value exists in array
array.slice(start, end)	Returns a shallow copy of array
array.indexOf(value)	Returns index of value in array
Object	
Object.keys(obj)	Returns keys of object
Object.values(obj)	Returns values of object
Object.entries(obj)	Returns key-value pairs
Object.assign({}, source)	Copies values into a new object
obj.hasOwnProperty(key)	Checks if object has property
String	

Snippet	Description
str.toUpperCase()	Converts to uppercase
str.toLowerCase()	Converts to lowercase
str.includes("text")	Checks for substring
str.replace("", "")	Replaces text
str.split(",")	Splits string into array
str.trim()	Removes whitespace
str.slice(start, end)	Extracts part of string
Number	
Number(value)	Converts to number
num.toFixed(2)	Formats number to fixed decimals
num.toString()	Converts number to string
Boolean	
Boolean(value)	Converts to boolean
!value	Negation
!!value	Double negation (truth check)
MapSet	
const map = new Map();	Creates new map
map.set("key", value);	Sets key-value pair
map.get("key");	Gets value by key
map.has("key");	Checks key existence

Snippet	Description
map.delete("key");	Deletes key
const set = new Set();	Creates new set
set.add(value);	Adds value to set
set.has(value);	Checks value existence
set.delete(value);	Removes value from set
DateTime	
new Date("")	Creates date from string
dateVar.toISOString()	Converts to ISO string
dateVar.getTime()	Gets time in ms
Date.now()	Gets current timestamp
Timers	
setTimeout(() => {}, 1000);	Delays function execution
setInterval(() => {}, 1000);	Repeats function execution
clearTimeout(timeoutId);	Clears timeout
clearInterval(intervalId);	Clears interval
FileSystem	
fs.readFileSync("", "utf8")	Reads file content
fs.writeFileSync("", "")	Writes to file
fs.appendFileSync("", "")	Appends to file
const fd = fs.openSync("", "rw")	Opens file

Snippet	Description
fs.fsyncSync(fd);	Flushes file content
fs.closeSync(fd);	Closes file
fs.existsSync("")	Checks file existence
fs.readdirSync("")	Reads directory content
Functions	
function myFunction() { }	Declares a function
const add = () => null;	Declares an arrow function
myFunction();	Calls a function

故障排除

注： console.log()在此脚本环境中不可用。要识别错误，请查看位于以下位置的错误日志：%APPDATA%/TekLynx/web-forms/logs

问题	解决方案
脚本未运行	检查您是否点击了“应用”按钮并将其绑定到了正确的事件
验证失败	检查语法并确保所有变量都已声明

提示

1. 使用*返回*将结果传递回表单系统。
2. 合并多行逻辑——支持所有 JavaScript 功能。

使用表单变量

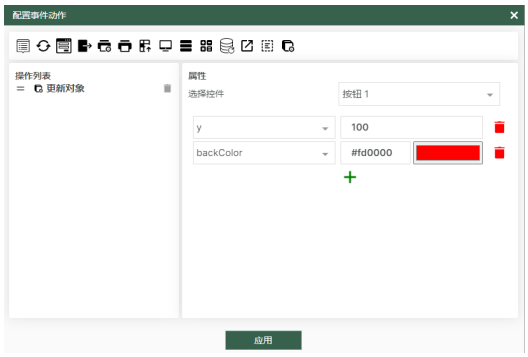
您可以通过在脚本中使用变量名为其赋新值，直接更新表单变量。例如：如果一个表单变量名为 candy_amount，您可以这样设置它的值：

```
candy_amount = 12;
```

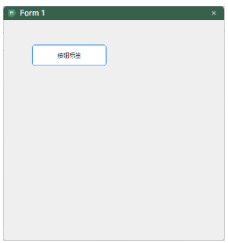
脚本引擎将自动识别此赋值，并在脚本执行完成后将该值应用到数据源中相应的表单变量。您还可以通过在下一个动作的配置中引用“上一个动作的结果”，在后续操作中使用一个脚本动作的返回语句的结果。

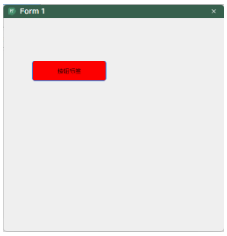
更新对象动作

更新对象动作允许用户在事件触发时，在运行时更改所选控件的属性。在配置事件操作对话框中添加此动作后，用户在“选定控件”中选择目标控件，然后在“属性列表”部分定义一个或多个要更新的属性。



这可以用来动态修改控件的行为或外观，例如根据用户交互或先前操作的结果更改位置、大小、文本、颜色、可见性或其他支持的单值属性。





限制：此操作不能用于设置需要多个选项或复杂集合的属性，例如列表框或单选按钮组的项目。它也不能用于设置或更改活动打印机。



France
+33 (0) 562 601 080

Germany
+49 (0) 2103 2526 0

Singapore
+65 6908 0960

United States
+1 (414) 837 4800

Copyright 2026 TEKLYNX Corporation SAS. All rights reserved. LABEL MATRIX, LABELVIEW, CODESOFT, LABEL ARCHIVE, SENTINEL, PRINT MODULE, BACKTRACK, TEKLYNX CENTRAL, TEKLYNX, and Barcode Better are trademarks or registered trademarks of TEKLYNX Corporation SAS or its affiliated companies. All other brands and product names are trademarks and/or copyrights of their respective owners.

www.teklynx.com