Power Form 使用者指南

2.0 □

□□□

目录

□ □ 4
□ □ 4
□ □ □ □ 5
□ □ □ 5
□ □ □ □ □ 7
□ □ □ □ 8
□ □ □ 9
□ □ □ 9
□ □ □ □ 10
□ □ □ □ □ □ 10
腳□ □ □ 15
□ □ □ □ 23
□ □ □ □ □ □ 24
□ □ □ □ □ □ 24

介紹

Power Form □ □ □

功能表列

□ □ □ □ □ □ 4 □ □ □ □ □

控制項	描述
□□□□	□□□□□□
□□□	□□□□□□□□□□□□□□□
□□	□□□□□□□□/□□□□□□□□
□□□□	□□□□□□□□□□□□□□
□□□□	□□啟用/□□□□□/□□□□□□□□□□□□□
□□□□/□□	□□□□□□值的資料字段
□□□	□□□□□□□□□□□□□□□□□□□□□□□□□
□□□	□□□□□□□□
□□□	□□□□□□□□□□□□
□□□□□	□□□□□□□□□□□□□□□□□□□□□
□□□□□	□□□□□□□□□□□□□□□□□□□□

解決方案樹

解決方案樹是顯示在應用程式左側的一組標籤頁。它由多個部分組成，用於將通用配置應用於解決方案。



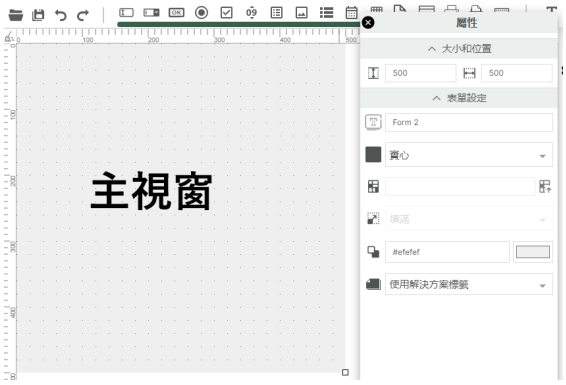
解決方案選項卡由 3 □□□□

屬性網格

屬性網格中列出所選控制項的所有屬性。在這裡，您可以變更控制項的特性，例如控制項名□□□

主視窗

顯示在中心位置的主視窗是您的工作區。在這裡，您可以新增控制項並建立用於輸入標籤列 □ **檔案 > 運行**□ □ □ □ □ □ 查看最終結果 □



狀態列

狀態列□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ x □ □ y □



基礎訊息

什麼是動作？

動作□□□□□□□□□□/□□□□□□□□□ Power Form □□□□□ 14 □□□□□□□□ **動作**□□□□□□□□□□

動作	**更新變數值**
	□ □ □ □ □ □ □ □ □ □
參數	**變數名稱**
	□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □
	新的值類型
	□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ 值類型。
	選項
	□ □ □ □ □ □ □ □ □ □ □ □ 值添加新的值。
	新值
	□ 值。

動作	**顯示表單。**
	□ □ □ □ □ □ 啟新表單。
參數	**要顯示的表單**
	□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □
	顯示表單行為
	□ □ □ □ □ □ □ □ □ □ □ □ 啟所選表單。
	選項
	□ □ □ □ □ □ 啟先前已開啟的表單（如果存在）。

動作	**退出表格。**
	□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

動作	**設定印表機**
	□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

	□□□□

動作	**開啟對話方塊**
	□ □ □ □ □ □ 啟標籤設計器對話方塊
參數	**開啟對話方塊**
	□ □ □ □ □ □ □ □ □ □ □ 啟的對話方塊。

動作	**開啟網頁連結**
	□ □ □ □ □ □ 啟選定的網頁鏈接
參數	**連結地址**
	□ □ □ □ □ □ □ □ □ □ □ □ □ □ □

動作	**執行 SQL 請求**
	□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ SQL □ □ □ □ □ □
參數	**資料庫**
	□ □ □ □ □ SQL □ □ □ □ □ □
參數	**SQL 語句**
	SQL □ □ □ □ □ □ □

動作	**執行腳本請求**
	□ □ □ □ □ □ □ □ □ □ JavaScript □ □ □ □ □ □ 值、更新變數、將資料 □ □ □ □ □ □
參數	**腳本**
	□ □ □ □ □ □ □
參數	**類別**
	□ □
參數	**程式碼片段**
	□ □ □ □ □ □ □ □ □ □ □ □ □
參數	**增加**

	□ □
參數	**驗證**
	□ □

動作	**刷新資料庫**
	□ □ □ □ □ □ □ □ □ □ □ □ □
參數	**從以下來源取得資料**
	□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

腳本動作

□ □ □ 腳本動作」允許使用者將 JavaScript □ □ □ □ □ □ Power Form □ □ □ □ □ □ UI □ 戶互動的腳本來擴展應用程式的邏輯。

步驟詳解：

1. □ □ □ □ □ □ □ □ □ □ □
    1. □ □ □ □ □ □ 腳本的 UI □ □ □ □ □ □ □ □ □ □
    2. □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

5. □ □ □ □ □

    1. □ □ □ □ □ □ □ □ □ 查腳本是否有語法錯誤。
    2. □

使用程式碼片段

腳本介面提供了一個按類別組織的預定義程式碼片段清單。使用這些程式碼片段可以插入常

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

Snippet	Description
Math.max(a, b)	Returns the maximum of two values
Math.min(a, b)	Returns the minimum of two values
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

Snippet	Description
Math.LN2	Natural log of 2
Math.LN10	Natural log of 10
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

Snippet	Description
a - b	Subtraction
a * b	Multiplication
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

Snippet	Description
switch (expression) { case: break;}	Switch-case conditional
try { } catch (e) { }	Try-catch block
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

Snippet	Description
obj.hasOwnProperty(key)	Checks if object has property
String	
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

Snippet	Description
map.get("key");	Gets value by key
map.has("key");	Checks key existence
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

Snippet	Description
fs.appendFileSync("", "")	Appends to file
const fd = fs.openSync("", "rw")	Opens file
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

疑難排解

□ □ `console.log()` □ □ 腳本環境中不可用。若要識別錯誤，請查看位於下列位置的錯誤

□ □ □ `%APPDATA%/TekLynx/web-forms/logs`

問題	解決方案
腳本未運行	□ 查您是否點擊了「應用」按鈕並將其綁定到了正確的事件
□ □ □ □	□ 查語法並確保所有變數都已聲明

提示

1. 使用*返回*將結果傳回表單系統。
2. 支援多行邏輯——所有 JavaScript 功能均受支援。

使用表單變數

□ □ □ □ 腳本中使用變數名為其賦新值，從而直接更新表單變數。例如，如果一個表單變數名為 `candy_amount` 您可以這樣設定它的值：

```
candy_amount = 12;
```

腳本引擎將自動識別此賦值，並在腳本執行完成後將該值套用到資料來源中對應的表單變數□

□ □ □ □

□□□



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