

Power Form User Guide

Version 2.0

The information contained in this guide is not of a contractual nature and may be subject to change without prior notice.

The software described in this guide is sold under a license agreement. The software may be used, copied or reproduced only in accordance with the terms of the agreement.

No part of this guide may be copied, reproduced or transmitted in any form, by any means or for any purpose other than the purchaser's own use without the written permission of TEKLYNX Corporation SAS.

©2026 TEKLYNX Corporation SAS,
All rights reserved

Table of Contents

Introduction.....	4
The layout.....	4
The Menu bar	5
The Tool bar	5
The Solution tree	7
The Properties Grid	8
The Main Window	9
The Status bar	9
The Basics.....	10
What are Actions?	10
Script action	15
Troubleshooting	23
Working with Form Variables	24
Update object action	24

Introduction

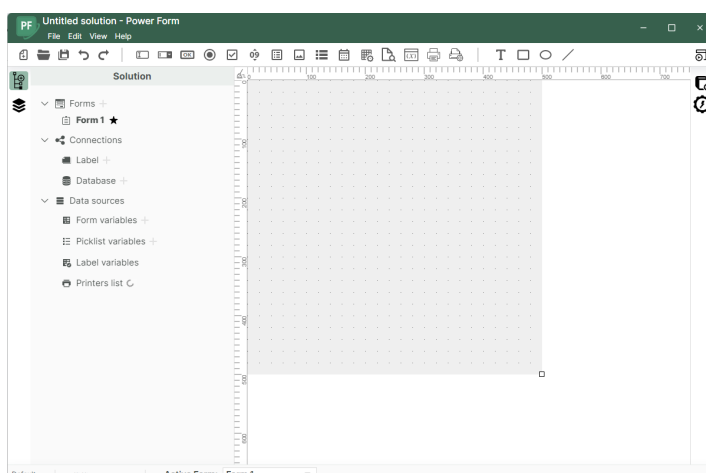
Power Form is a visual design tool that allows you to rapidly build front-end data entry and label printing applications.

It includes features such as:

- Label and printer type selection
- Print preview
- Page and printer settings
- Database connections
- Multiple screen/form application
- Pre-defined values and data control

The layout

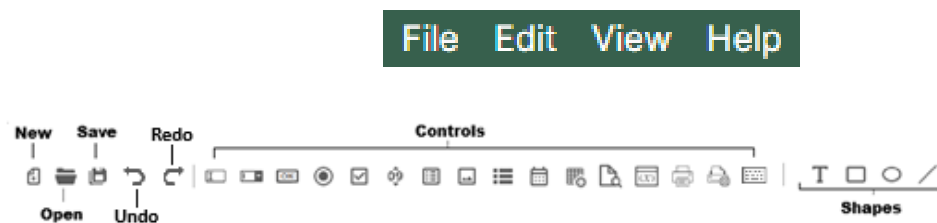
Power Form's layout is easy to use, reducing the time necessary to familiarize yourself with the software. It includes a Menu bar, a Tool bar, a Solution tree, a Properties grid, a Main window and a Status bar.



The Menu bar

The Menu bar contains 4 items: File, Edit, View and Help. When clicked, each item will expand showing a new list of items, to allow the user to perform a variety of functions. To access the menu items more quickly, press Alt + the letter from your desired choice:

- Alt+F will open the File menu.
- Alt+E will open the Edit menu.
- Alt+V will open the View menu.
- Alt+H will open the Help menu.
-



The Tool bar

The Tool bar is usually displayed just below the Menu bar. It is made up of icons which represent the most commonly used software functions and objects to operate the form. These icons can be clicked to quickly access these functions.

Objects are made up of 16 controls and 4 shapes that can be added to the form: Text input, Combobox, Button, Radio button, CheckBox, Numeric up/down input, GroupBox, PictureBox, ListBox, DatePicker, DataTableGrid, LabelPreview, Label variables data grid, Printer Selector, Print Settings, Text, Rectangle, Ellipse and Line.

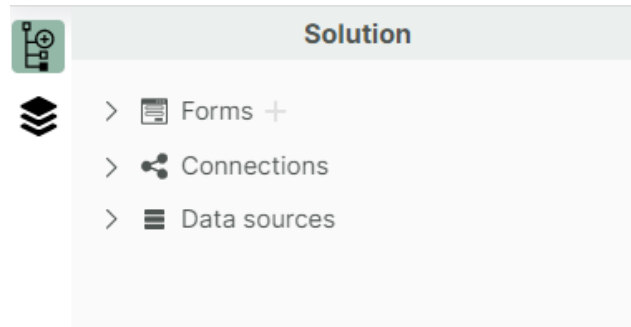
Each control serves a specific purpose. They can be linked to fixed or variable data and can be associated with actions. To insert a control into your form, simply click on the control and drag and drop it into the form.

Control	Description
Text input	Data entry field
ComboBox	Displays a drop-down list of data as defined in its properties
Button	An active control used to perform set functions/actions
Radio button	Displays a list of data as a radio group view.
CheckBox	A control used to activate/deactivate or select/deselect predefined features
Numeric up/down	Data entry field for the numeric values only
GroupBox	Non-active box placed around a group of controls for visual organization
PictureBox	A box where an image can be imported
ListBox	Displays a list of data as defined in its properties
DatePicker	Data entry field for the date format defined in its properties
DataTableGrid	Displays the database table defined in the properties along with some basic functionality such as filters and search fields
LabelPreview	A box that will display a preview of the label attached to the form
Label variables data grid	Displays “When printed” label variables as Name, Value view
Printer Selector	Allows the user to select active printer and change selected printer settings

Control	Description
Print settings	Gives the user ability to print selected label, and set label quantity, label copy, page copy, and intercut parameters
Virtual keyboard	An on-screen keyboard used for data entry.
Text	Non-active text used to show the name of an input field
Rectangle	Displays a shape formed as a rectangle
Ellipse	Displays a shape formed as an Ellipse
Line	Displays a shape formed as a Line

The Solution tree

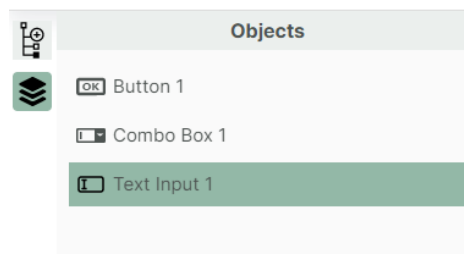
The **Solution tree** is set of tabs that is displayed on the left side of the application. It is made up of sections that are used to apply general configurations to the solution.



The **Solution tab** is made up of the 3 different sections that can be configured and applied to the whole form: Forms, Connections and Data sources.

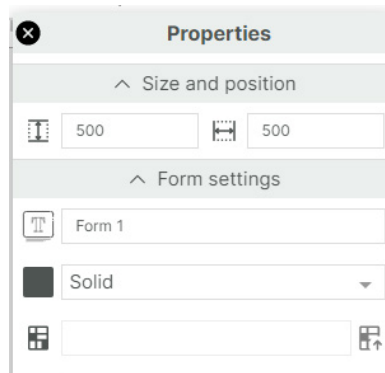
Section	Description
Forms	A list of forms that are used in the solution to represent different working areas
Connections	A list of available connections, that includes API connection, Label connection and Database connection.
Data sources	Contains different types of variables that can be used in the solution, there are 4 types of data source: Form variables, Picklist variables, Label variables and printer list.

The **Objects tab** is a list of objects that are placed on the active form.

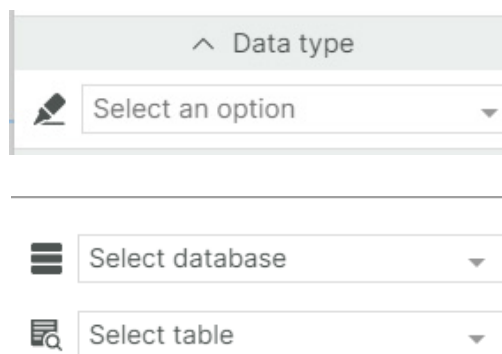


The Properties Grid

The **Properties Grid** lists all the properties for the selected control. Here you can change the characteristics of your controls such as control name, color, size, or actions. It is context-sensitive, so the list of properties will change according to the control selected. Having the grid displayed replaces the usual right-click menu command.

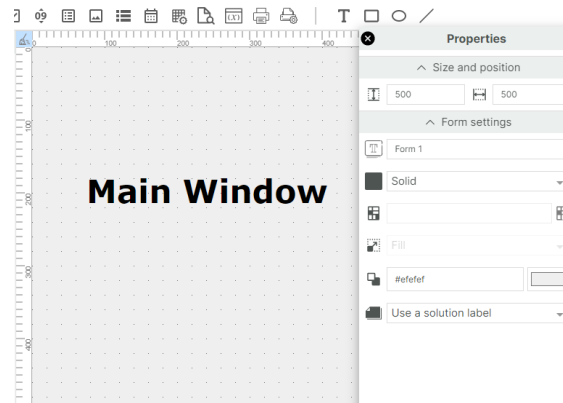


Some controls may include a Data Type section that allows users to assign a variable or link a database to that control. Once assigned, the control value will be bound with the selected data type and any changes made to either the control or the data type will be synchronized.



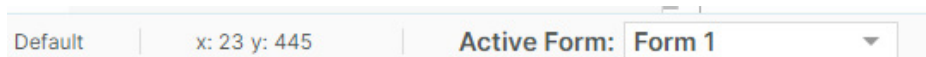
The Main Window

The **Main window**, displayed in the center, is your work area. Here, you will add controls and create the forms you will use to input data for label printing. Once the form has been created and configured, you can go to **File > Run** to run the form and see its final result.



The Status bar

The **Status bar** gives you the information about active printer, x axis and y axis location of your pointer and the active form selector dropdown.



The Basics

What are Actions?

Actions are functions that are assigned to buttons and to load/unload forms. In Power Form, 14 pre-defined Actions have been included in the **Actions** menu to save you time and to eliminate the need for programming. In some cases, where specific parameters must be configured, these have also been predefined in the action's configuration

The predefined actions are:

Action	Confirmation popup
	Displays the confirmation dialog box where the user can see a predefined message and buttons to accept or cancel the message.
Parameters	Title
	Allows the user to set headline title.
	Message
	Allows the user to set main text of the message.
	Buttons
	Dropdown with set of different button options to be displayed in the message popup.
	Icon
	Dropdown with set of different icons to be displayed in the message popup.

Action	Update variable value
	Allows the user to update variables.
Parameters	Variable name
	The user can select name of the variable to be updated.
	New value type
	Dropdown of value types that can be applied to variable to be updated.
	Option
	Allows the user to add new value to existing value of the selected variable.
	New value
	Field that allows the user to select a new value for the selected variable.

Action	Display form.
	Allows the user to open a new form.
Parameters	Form to display
	List of forms where the user can select what form to display.
	Display form behavior
	Opens the chosen form in a new window or in the same one.
	Option
	Allows the user to open previously opened form if it exists.

Action	Exit form.
	Allows the user to close form executor window.

Action	Set printer
	Allows the user to print from the selected printer and override any printers that may be associated with the document.
Parameter	Set Printer
	Allows you to set a printer as the default printer to use when the button is clicked, or to use the printer selected from a list - if a printer selection variable is used and defined in the form properties.

Action	Print Label
	Allows the user to print.
Parameters	Label count
	Sends the printer the number of labels to be printed. If the document includes a counter, the counter increments along with the number of labels.
	Label Copy
	Lets the printer know how many identical copies of each label should be printed.
	Page Copy
	Lets the printer know how many copies of each page to print.
	Inter Cut
	Lets the printer know how many labels to print before each cut.

Action	Read data from file
	Allows the user to read files data and apply it value to the variable or control.
Parameters	Choose file
	Allows the user to select file to read.
	Maximum size
	Sets maximum size of lines that will be read from the file.
	Apply value type
	Allows the user to select where apply data from file it can be variable or control.
	Select variable/control
	Allows the user to select to wich control data will be applied.

Action	Open dialog
	Allows the user to open the Label Designer dialogs
Parameter	Open dialog
	Dropdown with list of the available dialogs to open.

Action	Execute SQL request
	Allows the user to send SQL request to connected database for execution.
Parameter	Database
	Select database where SQL request will be executed
Parameter	SQL Statement
	Text area for SQL code

Action	Execute Script request
	Allows the user to run the native JavaScript code to compare values, update variables, write data to file, etc.
Parameter	Script
	Text area for code
Parameter	Category
	Dropdown with list of categories names for methods, constants, operators etc.
Parameter	Snippets
	List of snippets in selected category
Parameter	Add
	Button to add selected snippet to code text area
Parameter	Validate
	Button to validate the syntax of the code entered in the text area.

Action	Refresh database
	Allows the user to force a refresh of the database
Parameter	Take data from
	A dropdown with the option 'Database' to force refresh the database and 'From previous action result' to refresh the database using SQL action filtering.

Action	Open Web Link
	Allows the user to open a specified web link.
Parameter	Link address

	Field for entering the web link address to be opened.
--	---

Action	Set focus
	Allows the user to set focus to the selected control.
Parameter	Select control
	Selects the control that should receive focus.

Action	Update object
	Allows the user to update the selected control.
Parameter	Select control
	Selects the control that will be updated.
Parameter	List of properties
	Displays the list of properties available for update for the selected control

Script action

The "Execute Script Action" allows users to bind JavaScript code to UI elements (like buttons) in a Power Form application. This lets you extend the app's logic by writing scripts that respond to user interactions.

Step-by-Step Instructions:

1. Select a UI Element

1. Click on the UI element you want to attach a script to (e.g., a button).
2. Tip: You can select buttons, inputs, grids, or anything that supports events.

2. Open the Events Panel

1. Click the gear icon .
2. In the "Events" panel, find the event you want to handle—e.g., "Click."
3. Press the "..." button next to the event name to configure it.

3. Add a Script Action

1. In the "Configure Event Actions" dialog that opens:
 - a. Click the "Execute script action" icon .
 - b. A new action will appear in the list.

4. Write Your Script

In the "Properties" panel, enter your custom JavaScript logic.

Example:

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

* This JavaScript code snippet defines two constant variables, `a` assigned the value 12 and `b` assigned the value 32. It then returns the result of multiplying `b` by `a`.

5. Validate & Apply

1. Click "Validate" to check your script for syntax errors.
2. Once successful, press "Apply" to save the action.

Using Code Snippets

The scripting interface offers a list of predefined snippets organized by category. These snippets help insert commonly used expressions, constants, structures, and functions without needing to write them from scratch.

To use a snippet:

1. Select a category from the dropdown.
2. Choose a snippet from the second dropdown.
3. Click “Add” to insert it into the script editor.

All Available Snippets:

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
Math.log(x)	Natural logarithm (base e) of x
Math.exp(x)	Returns e to the power of x

Snippet	Description
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
Math.LOG2E	Base 2 log of e
Math.LOG10E	Base 10 log of e

Snippet	Description
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
$a \geq b$	Greater than or equal to
$a \leq b$	Less than or equal to
$a \&\& b$	Logical AND
$a \parallel b$	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
a / b	Division
$a \% b$	Modulo

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

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

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

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

Snippet	Description
<code>fs.existsSync("")</code>	Checks file existence
<code>fs.readdirSync("")</code>	Reads directory content
Functions	
<code>function myFunction() { }</code>	Declares a function
<code>const add = () => null;</code>	Declares an arrow function
<code>myFunction();</code>	Calls a function

Troubleshooting

Note: `console.log()` is not available in this scripting environment. To identify errors, review the error logs located at `%APPDATA%/TekLynx/web-forms/logs`

Issue	Solution
Script not running	Check if you clicked Apply and bound it to the right event
Validation fails	Review syntax and ensure all variables are declared

Tips

1. Use *return* to pass the result back to the form system.
2. Combine multiple lines of logic—all JavaScript features are supported.

Working with Form Variables

You can directly update the Form variables within your script by assigning a new value to them using their name. For example, if a Form variable is named `candy_amount`, you can set its value like this:

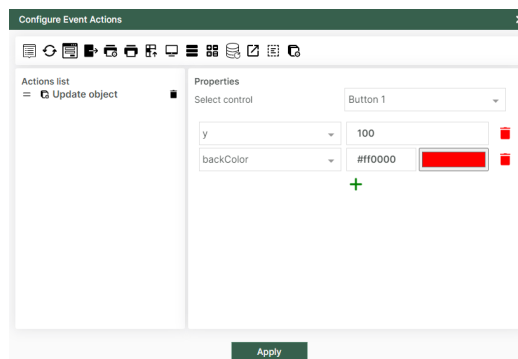
```
candy_amount = 12;
```

The script engine will automatically recognize this assignment and apply the value to the corresponding Form variable in the data source after script execution is complete.

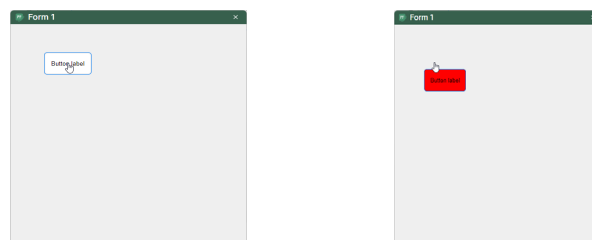
You can also use the result of a return statement from one script action in a subsequent action by referencing the “Previous action result” in the next action's configuration.

Update object action

Update object action allows the user to change properties of a selected control at runtime when an event is triggered. After adding this action in the [Configure Event Actions dialog](#), the user chooses the target control in Selected control and then defines one or more properties to update in the List of properties section.



This can be used to dynamically modify control behavior or appearance, for example changing position, size, text, color, visibility, or other supported single-value properties based on user interaction or the result of a previous action.



Limitations: this action cannot be used to set properties that require multiple options or complex collections, such as items for a ListBox or Radio Group. It also cannot be used to set or change the active printer.



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