Form

Namespace: `System.Windows.Forms`

Properties / Eigenschaften

- *Property – Standardwert*
  Beschreibung oder Erläuterung der Eigenschaft

- **AutoScaleMode** – *Font*
  Skalierung basierend auf Schriftgröße
- **AutoSize** – *$false*
  Passt die Form automatisch an den Inhalt an
- **BackColor** – *Control*
  Hintergrundfarbe
- **ClientSize** – *(300,300)*
  Innenbereich der Form (ohne Rahmen)
- **ControlBox** – *$true*
  Zeigt Schließen / Minimieren / Maximieren
- **ForeColor** – *ControlText*
  Standard-Textfarbe
- **FormBorderStyle** – *Sizable*
  Fensterrahmen (`None`, `FixedSingle`, `Sizable`, …)
- **Icon** – *$null*
  Fenster-Icon (`[Icon]::new("App.ico")`)
- **KeyPreview** – *$false*
  Form bekommt Key-Events vor Controls
- **MaximizeBox** – *$true*
  Maximieren erlauben
- **MaximumSize** – *(0,0)*
  Maximalgröße (0 = unbegrenzt)
- **MinimumSize** – *(0,0)*
  Minimale Größe
- **Opacity** – *1.0*
  Transparenz (0.0 – 1.0)
- **Padding** – *(0)*
  Innenabstand
- **StartPosition** – *WindowsDefaultLocation*
  Startposition des Fensters

- **ShowIcon** – *$true*
  Icon anzeigen
- **ShowInTaskbar** – *$true*
  In Taskleiste sichtbar
- **Size** – *(300,300)*
  Fenstergröße
- **Text** – ""
  Fenstertitel
- **TopMost** – *$false*
  Immer im Vordergrund
- **WindowState** – *Normal*
  (`Normal`, `Minimized`, `Maximized`)

Eigenschaften, die sich gegenseitig beeinflussen

- `AutoSize = $true` → ignoriert **Size**
- `Dock = "Fill"` → ignoriert **AutoSize**
- `Dock = "Top"` / `"Bottom"` → **Width** wird ignoriert
- `Dock = "Left"` / `"Right"` → **Height** wird ignoriert
- `FormBorderStyle = "None"` → keine ControlBox, kein **Icon** sichtbar

Eine `Form` ist das **Hauptfenster deiner Anwendung**.
Sie ist der Container für alle anderen Controls.

Grundidee

Die `Form` ist **die Bühne**.

- enthält alle Controls
- verwaltet Layout und Lebenszyklus
- steuert Anzeige (`Show` / `ShowDialog`)

Typischer Ablauf

1. Properties setzen
2. Controls hinzufügen
3. Events definieren
4. Form anzeigen (`Show()` / `ShowDialog()`)

Form erstellen

```
# Klassisch
$form = New-Object System.Windows.Forms.Form

# .NET-Style
$form = [System.Windows.Forms.Form]::new()
```

Form anzeigen

```
# Nicht blockierend
$form.Show()

# Modal (blockierend)
$form.ShowDialog()
```

Controls hinzufügen

```
$form.Controls.Add($button)
$form.Controls.AddRange(@($label, $textbox))
```

Layout & Verhalten

```
$form.Size = [System.Drawing.Size]::new(400, 300)
$form.StartPosition = "CenterScreen"
$form.TopMost = $true
$form.FormBorderStyle = "FixedDialog"
```

Events - *Form*

```
$form.Add_*
  param($sender, $e)
```

- `$sender` → die Form selbst
- `$e` → EventArgs

Load

Wird beim Initialisieren der Form ausgelöst

```
$form.Add_Load({
    Write-Host "Form lädt"
})
```

Shown

Wird nach dem Anzeigen ausgelöst

```
$form.Add_Shown({
    Write-Host "Form sichtbar"
})
```

FormClosing

Vor dem Schließen (kann verhindert werden)

```
$form.Add_FormClosing({
    param($sender, $e)

    $e.Cancel = $true  # verhindert Schließen
})
```

FormClosed

Nach dem Schließen

```
$form.Add_FormClosed({
    Write-Host "Form geschlossen"
})
```

Resize

Bei Größenänderung

```
$form.Add_Resize({
    Write-Host "Neue Größe: $($this.Size)"
})
```

Tipps & Tricks - *Form*

Typische Stolperfallen

- **Form schließt sofort**
  → kein `ShowDialog()` verwendet
- **Layout bricht auseinander**
  → `Dock` / `Anchor` falsch gesetzt
- **Größe ignoriert**
  → `AutoSize` aktiv
- **Fenster reagiert nicht auf Keys**
  → `KeyPreview = $true` fehlt

Mentales Modell

Die `Form` ist der **Lebenszyklus-Controller deiner UI**.

Sie bestimmt:

- wann etwas sichtbar ist
- wann etwas endet
- wie alles organisiert ist

Wann sinnvoll?

- Immer (jede WinForms-App braucht mindestens eine Form)

Wann problematisch?

- Zu viel Logik direkt in der Form
- Vermischung von UI und Business-Logik

Revision #5
Created 2026-03-03 13:16:21 UTC by John-Andreas Borinas
Updated 2026-04-01 19:50:26 UTC by John-Andreas Borinas