

CheckedListBox



Namespace: [System.Windows.Forms](#)

Properties / Eigenschaften

- *Property - Standardwert*
Beschreibung oder Erläuterung der Eigenschaft

Verhalten (Behavior)

- **AllowDrop** - *\$false*
Drag & Drop erlauben
- **CheckOnClick** - *\$false*
Checkbox wird direkt beim Klick geändert (ohne zweiten Klick)
- **Enabled** - *\$true*
Aktiv / deaktiviert
- **FormattingEnabled** - *\$true*
Für komplexe Objekte
- **SelectionMode** - *One*
Auswahlmodus (meist irrelevant hier)
- **Sorted** - *\$false*
Automatisch sortieren
- **ThreeDCheckBoxes** - *\$false*
3D-Darstellung der Checkboxes
- **ScrollAlwaysVisible** - *\$false*
Scrollbar immer anzeigen
- **HorizontalScrollbar** - *\$false*
Horizontale Scrollbar
- **TabStop** - *\$true*
Fokus per Tab
-

Daten / Inhalt (Data)

- **Items** - *(leer)*
Alle Einträge

- **CheckedItems** - *(leer)*
Alle **aktiv angehakten** Items
 - **CheckedIndices** - *(leer)*
Indizes der angehakten Items
 - **SelectedItem** - *\$null*
Aktuell markiertes Item (nicht gleich checked!)
 - **TabIndex** - 0
Tab-Reihenfolge
-

Layout & Position

- **Location** - *(0,0)*
Position
 - **Size** - *(Width=120, Height=96)*
 - **Width** - *(abhängig vom Layout)*
 - **Height** - *(abhängig vom Layout)*
 - **Anchor** - *(Top, Left)*
Verhalten bei Größenänderung
 - **Dock** - *None*
Layout im Container
 - **ItemHeight** - *(abhängig von Font)*
-

Aussehen (Appearance)

- **BackColor** - *SystemColors.Window*
Hintergrundfarbe
 - **ForeColor** - *SystemColors.WindowText*
Textfarbe
 - **Font** - *Standard-Systemfont*
Schriftart
 - **BorderStyle** - *Fixed3D*
Rahmenstil (`None`, `FixedSingle`, `Fixed3D`)
 - **Visible** - *\$true*
Sichtbarkeit
-

Meta / System

- **Name** - ""
Interner Name
- **TopIndex** - 0
Oberstes sichtbares Item

Größe

Die **CheckedListBox** ist im Grunde eine [ListBox](#) mit eingebauten Checkboxes.

CheckedListBox erstellen

```
$clb = New-Object System.Windows.Forms.CheckedListBox  
$clbNew = [System.Windows.Forms.CheckedListBox]::new()
```

Items hinzufügen

```
$clb.Items.Add("Apfel")  
$clb.Items.Add("Banane")  
$clb.Items.AddRange(@("Orange", "Mango", "Traube"))
```

☐ Werte auslesen

```
# Alle angehakten Items  
$checked = $clb.CheckedItems  
  
# Indizes  
$indices = $clb.CheckedIndices  
  
# Einzelnes Item (nur Fokus, nicht Check!)  
$selected = $clb.SelectedItem
```

☐ Wichtiger Unterschied:

`SelectedItem` ≠ `CheckedItems`

- `SelectedItem` betrifft nur das Item, welches Ausgewählt ist.
 - `CheckedItems` beinhaltet alle Items welches auch einen Haken haben
-

Events - *CheckedListBox*

Events

- **Event**
Wird ausgelöst, ...
-

- **TextChanged**
... wenn sich die Text-Eigenschaft des Controls ändert

Interaktion

- **Click**
... wenn ein Eintrag der Liste angeklickt wird
- **DoubleClick**
... wenn ein Eintrag doppelt angeklickt wird
- **ItemCheck**
... wenn sich der Checked-Zustand eines Items ändert (**bevor** die Änderung übernommen wird)
- **SelectedIndexChanged**
... wenn sich die Auswahl (markiertes Item) ändert

Maus

- **MouseEnter**
... wenn der Mauszeiger die CheckedListBox betritt
- **MouseLeave**
... wenn der Mauszeiger die CheckedListBox verlässt
- **MouseMove**
... wenn der Mauszeiger innerhalb der CheckedListBox bewegt wird
- **MouseDown**
... wenn eine Maustaste auf der CheckedListBox gedrückt wird
- **MouseUp**
... wenn eine Maustaste auf der CheckedListBox losgelassen wird
- **MouseHover**
... wenn der Mauszeiger für kurze Zeit auf der CheckedListBox verweilt

Tastatur

- **KeyDown**
... wenn eine Taste gedrückt wird, während die CheckedListBox den Fokus hat

- **KeyUp**
... wenn eine gedrückte Taste wieder losgelassen wird
- **KeyPress**
... wenn eine Taste gedrückt wird, die ein Zeichen erzeugt (z. B. Buchstaben, Zahlen)

Fokus

- **Enter**
... wenn das Control Fokus erhält
- **Leave**
... wenn das Control Fokus verliert

Darstellung / Layout

- **Paint**
... wenn die CheckedLisBox neu gezeichnet wird
- **LocationChanged**
... wenn sich die Position der CheckedListBox ändert
- **SizeChanged**
... wenn sich die Größe der CheckedListBox ändert

Zustand

- **EnabledChanged**
... wenn sich der Enabled-Status der CheckedListBox ändert
- **VisibleChanged**
... wenn sich die Sichtbarkeit der CheckedListBox ändert

Aussehen

- **FontChanged**
... wenn sich die Schriftart der CheckedListBox ändert
- **ForeColorChanged**
... wenn sich die Textfarbe der CheckedListBox ändert
- **BackColorChanged**
... wenn sich die Hintergrundfarbe der CheckedListBox ändert

Datenbindung

- **DataSourceChanged**
... wenn sich die Datenquelle ändert
- **DisplayMemberChanged**
... wenn sich die Anzeigebindung ändert
- **ValueMemberChanged**
... wenn sich die Wertbindung ändert

`$sender`

→ Das Objekt, das das Event ausgelöst hat. Meistens die `CheckedListBox` selbst

`$e`

→ Zusatzinfos zum Event → **was genau ist passiert**

```
$CheckedListBox.Add_Click({  
    param($sender, $e)  
  
    $sender.BackColor = "Red"  
})
```

ItemCheck

```
$CheckedListBox.Add_ItemCheck({  
    param($sender, $e)  
  
    $e.Index  
    $e.NewValue  
    $e.CurrentValue  
})
```

- `$e.Index` → welches Item
- `$e.NewValue` → neuer Zustand
- `$e.CurrentValue` → alter Zustand

Tipps & Tricks - *CheckedListBox*

Check direkt beim Klick

```
$checkBox.CheckOnClick = $true
```

Alle gesetzten Werte holen

```
$clb.CheckedItems | ForEach-Object {  
    $_
```

```
}
```

☐ Check programmatisch setzen

```
$clb.SetItemChecked(0, $true)
```

⚠ Typische Stolperfallen

- `SelectedItem` benutzt statt `CheckedItems`
- `ItemCheck` falsch verstanden (Timing!)
- Doppel-Events durch Click + ItemCheck
- `CheckOnClick = $false` → wirkt wie Bug

☐ Mentales Modell

Die `CheckedListBox` hat **zwei Zustände gleichzeitig**:

1. Fokus (`SelectedItem`)
2. Check-Zustand (`CheckedItems`)

Wenn du das vermischst → Chaos.

☐ Best Practice

- Für mehrere unabhängige Optionen → perfekt
- Für „eine Auswahl“ → falsches Control
- Für komplexe Daten → lieber eigene UI bauen

Revision #12

Created 2026-03-10 03:18:06 UTC by John-Andreas Borinas

Updated 2026-04-11 02:05:45 UTC by John-Andreas Borinas