

Der Operator as

[Themengruppen](#)

[Siehe auch](#)

Der Operator **as** führt eine Typumwandlung mit Laufzeitprüfung durch. Der Ausdruck

Objekt **as** Klasse

gibt eine Referenz auf dasselbe Objekt wie Objekt, aber mit dem von Klasse angegebenen Typ zurück. Zur Laufzeit muß Objekt eine Instanz von Klasse oder einem ihrer Nachkommen bzw. **nil** sein. Andernfalls wird eine Exception ausgelöst. Wenn der deklarierte Typ von Objekt nicht mit Klasse verwandt ist (wenn die Typen also unterschiedlich und nicht voneinander abgeleitet sind), gibt der Compiler eine Fehlermeldung aus. Ein Beispiel:

```
with Sender as TButton do
begin
  Caption := '&Ok';
  OnClick := OkClick;
end;
```

Die Regeln der Auswertungsreihenfolge machen es häufig erforderlich, as-Typumwandlungen in Klammern zu setzen:

```
(Sender as TButton).Caption := '&Ok';
```

Schnittstellenabfragen

[Themengruppen](#)

[Siehe auch](#)

Mit Hilfe des Operators `as` können Schnittstellenumwandlungen durchgeführt werden. Dieser Vorgang wird als Schnittstellenabfrage bezeichnet. Eine Schnittstellenabfrage generiert auf der Basis des (zur Laufzeit vorliegenden) Objekttyps aus einer Objektreferenz oder einer anderen Schnittstellenreferenz einen Ausdruck vom Typ einer Schnittstelle. Die Syntax für eine Schnittstellenabfrage lautet

`Objekt as Schnittstelle`

Objekt ist entweder ein Ausdruck vom Typ einer Schnittstelle oder Variante, oder er bezeichnet eine Instanz einer Klasse, die eine Schnittstelle implementiert. Schnittstelle kann jede mit einer GUID deklarierte Schnittstelle sein.

Wenn Objekt den Wert `nil` hat, liefert die Schnittstellenabfrage als Ergebnis `nil`. Andernfalls wird die GUID der Schnittstelle an die Methode `QueryInterface` von Objekt übergeben. Wenn `QueryInterface` einen Wert ungleich Null zurückgibt, wird eine Exception ausgelöst. Ist der Rückgabewert dagegen Null (d.h. die Schnittstelle ist in der Klasse von Objekt implementiert), ergibt die Schnittstellenabfrage eine Referenz auf Objekt.