

TaskDependency-Objekt

Autor & Copyright: Dipl.-Ing. Harald Nahrstedt

Version: 2016 / 2019 / 2021 / 365

Erstellungsdatum: 08.01.2023

Überarbeitung:

Beschreibung:

Das *TaskDependency*-Objekt stellt die Verbindungsart und den Zeitabstand zwischen Vorgängen dar. Das *TaskDependency*-Objekt ist ein Element der *TaskDependencies*-Objektliste.

Anwendungs-Datei: 17-01-05_TaskDependencyObjekt.mpp

1 Die TaskDependencies-Objektliste

Wie bei jeder Objektliste werden ihre Elemente über einen Index angesprochen, oder mit der *For-Each-Next*-Anweisung aus der Liste gelesen. Dazu betrachten wir ein einfaches Projekt im Gantt-Diagramm (Bild 1).

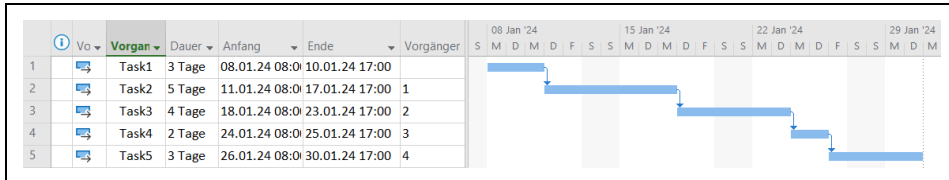


Bild 1. Einfaches Projekt im Gantt-Diagramm

Codeliste 1. Die Prozedur zeigt die Abhängigkeiten zwischen den Vorgängen

```

Sub ReadDependencies ()
    Dim objTask As Task
    Dim sText As String
    Dim iCount As Integer

    For Each objTask In ActiveProject.Tasks
        sText = objTask.Name
        For iCount = 1 To objTask.TaskDependencies.Count
            sText = sText & " / " & _
                Str(objTask.TaskDependencies(iCount).From)
        Next iCount
        Debug.Print sText
    Next
End Sub

```

Das Ergebnis befindet sich nach dem Aufruf im Direktfenster (Bild 2).

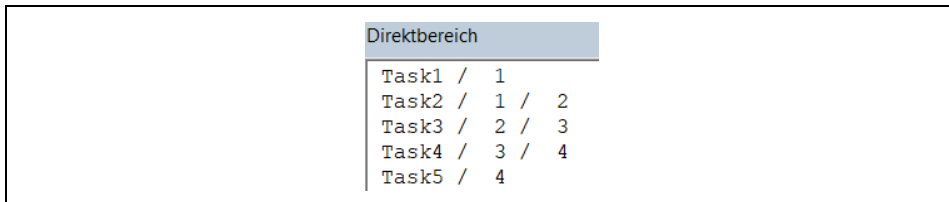


Bild 2. Ausgabe der Abhängigkeiten im Direktfenster

1.1 Methoden

Die *TaskDependencies*-Objektliste verfügt lediglich über die *Add*-Methode. Ihre Syntax lautet

```
Ausdruck.Add (From, Type, Lag)
```

Tabelle 1. Die Parameter der Add-Methode

Parameter	Erforderlich	Datentyp	Beschreibung
From	Erforderlich	Objekt	Das angegebene <i>Task</i> -Objekt wird zum Vorgänger des durch <i>Ausdruck</i> angegebenen Vorgangs.

Type	Optional	Long	Der Typ der Beziehung zwischen den verknüpften Vorgängen. Dies kann eine der <i>PjTaskLinkType</i> - Konstanten sein. Der Standardwert ist <i>PjFinishToStart</i> .
Lag	Optional	Variant	Die Dauer des Zeitabstand zwischen Vorgängen. Um Zeitabstand zwischen Vorgängen anzugeben, verwenden Sie einen negativen Wert ein. Zeichenfolge standardmäßig Tage, sofern nicht anders angegeben. Andere Werte werden als Minuten interpretiert. Der Standardwert ist 0.

Tabelle 2. Die *pjTaskLinkType*-Konstanten

Konstante	Wert	Beschreibung
<i>pjFinishToFinish</i>	0	Vorgangsverknüpfungen von Ende zu Ende.
<i>pjFinishToStart</i>	1	Vorgangsverknüpfungen von Ende zu Anfang.
<i>pjStartToFinish</i>	2	Vorgangsverknüpfungen von Anfang zu Ende.
<i>pjStartToStart</i>	3	Vorgangsverknüpfungen von Anfang zu Anfang.

1.2 Eigenschaften

Die *Application*-Eigenschaft liefert das zugehörige *Application*-Objekt. Die *Count*-Eigenschaft liefert die Anzahl der Elemente in der *TaskDependencies*-Objektliste und wurde bereits in der Codeliste 1 verwendet.

Die *Item*-Eigenschaft ruft ein einzelnes *TaskDependency*-Objekt aus der Objektliste ab. Dazu ist die Angabe des *Index* erforderlich mit der Syntax

```
Ausdruck.Item(Index)
```

Die *Parent*-Eigenschaft ruft das übergeordnete Element ab. Sie wird verwendet, um auf die Eigenschaften oder Methoden des übergeordneten Elements zuzugreifen.

2 Das TaskDependency-Objekt

Das *TaskDependency*-Objekt stellt die Verbindungsart und den Zeitabstand zwischen Vorgängen dar. Das *TaskDependency*-Objekt ist ein Element der *TaskDependencies*-Objektliste.

2.1 Methoden

Auch das Einzelobjekt verfügt lediglich über eine *Delete*-Methode, mit der das *TableField*-Objekt aus einer *TableFields*-Objektliste gelöscht wird.

Codeliste 2. Die Prozedur löscht die erste Verbindung von *Task3*

```
Sub DeleteDependency ()
    ActiveProject.Tasks(3).TaskDependencies(1).Delete
End Sub
```

Dadurch wird der Projektablauf im Beispiel unterbrochen (Bild 3).

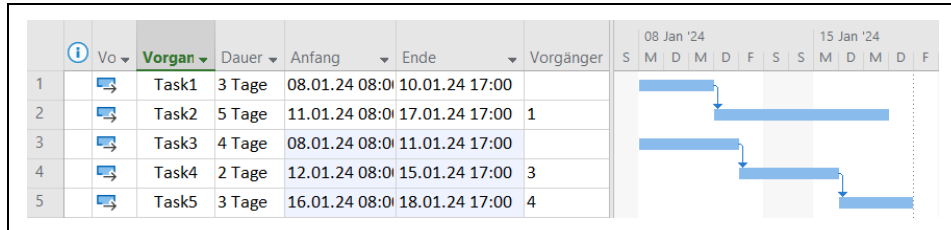


Bild 3. Auswirkung der Delete-Methode

2.2 Eigenschaften

Auch hier liefert die *Application*-Eigenschaft das zugehörige *Application*-Objekt. Die *From*-Eigenschaft ruft das Task-Objekt, von dem das Task-Objekt abhängig ist. Die folgende Prozedur liefert die abhängigen Tasks zur Task3 und gibt deren Namen und Task-ID aus.

Codeliste 3. Die Prozedur liefert die Abhängigkeiten von Task3

```

Sub TestDependenciesFrom()
    Dim objTask As Task
    Dim iDeps As Integer
    Dim lTaskId As Long
    Dim iCount As Integer

    iDeps = ActiveProject.Tasks(3).TaskDependencies.Count
    For iCount = 1 To iDeps
        Set objTask =
ActiveProject.Tasks(3).TaskDependencies(iCount).From
        lTaskId = ActiveProject.Tasks(3).TaskDependencies(iCount).From
        Debug.Print objTask.Name, lTaskId
    Next iCount
End Sub

```

Im Beispiel geben wir die Vorgänge Task1 und Task2 als Vorgänger von Task3 an (Bild 4) und erhalten nach dem Aufruf eine Ausgabe im Direktfenster (Bild 5).

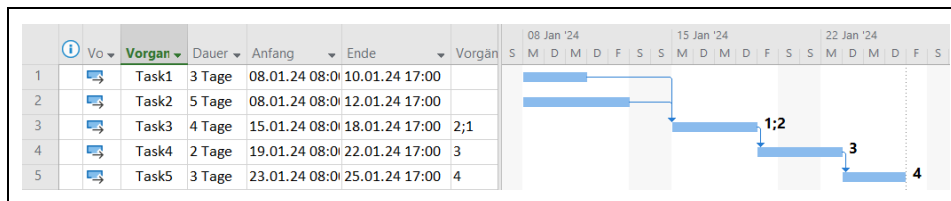


Bild 4. Task1 und Task2 sind Vorgänger von Task3

Direktbereich	
Task2	2
Task1	1
Task3	3

Bild 5. Ausgabe im Direktfenster

Die *Lag*-Eigenschaft (Verzögerung) beschreibt den Zeitabstand zwischen den Vorgängen. Die folgende Prozedur setzt die Verzögerung zwischen Task2 und Task3 auf 2 Tage.

Codeliste 4. Die Prozedur setzt eine Verzögerung von 2 Tagen zwischen Task2 und Task3

```
Sub SetLag()
    Dim objTask As Task

    Set objTask = ActiveProject.Tasks(3)
    objTask.TaskDependencies(1).Lag = "2t"
    Set objTask = Nothing
End Sub
```

Mit dem Aufruf verschiebt sich der Anfang von Task3 um 2 Tage (Bild 6).

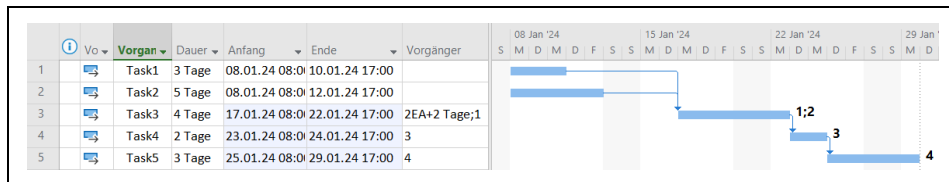


Bild 6. Auswirkung der Verzögerung

Die Angabe von negativen Werten erzeugt eine Verkürzung. Die *LagType*-Eigenschaft gibt die Einheit der Verzögerung an. Der angegebene Wert entspricht einer *pjFormatUnit*-Konstanten.

Tabelle 3. pjFormatUnit-Konstante

Konstante	Wert	Beschreibung
<i>pjDays</i>	7	Tage
<i>pjDaysEstimated</i>	39	Geschätzte Tage
<i>pjElapsedDays</i>	8	Verstrichene Tage
<i>pjElapsedDaysEstimated</i>	40	Verstrichene geschätzte Tage
<i>pjElapsedHours</i>	6	Verstrichene Stunden
<i>pjElapsedHoursEstimated</i>	38	Verstrichene geschätzte Stunden
<i>pjElapsedMinutes</i>	4	Verstrichene Minuten
<i>pjElapsedMinutesEstimated</i>	36	Verstrichene geschätzte Minuten
<i>pjElapsedMonths</i>	12	Verstrichene Monate
<i>pjElapsedMonthsEstimated</i>	44	Verstrichene geschätzte Monate
<i>pjElapsedWeeks</i>	10	Verstrichene Wochen
<i>pjElapsedWeeksEstimated</i>	42	Verstrichene geschätzte Wochen
<i>pjHours</i>	5	Stunden
<i>pjHoursEstimated</i>	37	Geschätzte Stunden

Konstante	Wert	Beschreibung
<code>pjMinutes</code>	3	Minuten.
<code>pjMinutesEstimated</code>	35	Geschätzte Minuten.
<code>pjMonths</code>	11	Monate.
<code>pjMonthsEstimated</code>	43	Geschätzte Monate.
<code>pjWeeks</code>	9	Wochen.
<code>pjWeeksEstimated</code>	41	Geschätzte Wochen.
Name	Wert	Beschreibung
<code>pjDays</code>	7	Tage.
<code>pjDaysEstimated</code>	39	Geschätzte Tage.
<code>pjElapsedDays</code>	8	Verstrichene Tage.
<code>pjElapsedDaysEstimated</code>	40	Verstrichene geschätzte Tage.
<code>pjElapsedHours</code>	6	Verstrichene Stunden.
<code>pjElapsedHoursEstimated</code>	38	Verstrichene geschätzte Stunden.
<code>pjElapsedMinutes</code>	4	Verstrichene Minuten.
<code>pjElapsedMinutesEstimated</code>	36	Verstrichene geschätzte Minuten.

Die *Parent*-Eigenschaft liefert das übergeordnete Element. Die *Path*-Eigenschaft liefert den Pfad des Projekts, das den externen Vorgang enthält oder legt diesen fest. Die *To*-Eigenschaft liefert den Vorgang, der in der Abhängigkeit der Nachfolger ist. Die folgende Prozedur liefert in der Abhängigkeitsfolge von Task3 logischerweise Task3, da Task1 und Task2 Vorgänger sind.

Codeliste 5. Die Prozedur liefert den Nachfolger in der Abhängigkeitsliste

```

Sub ReadTo()
    Dim objTask As Task

    Set objTask = ActiveProject.Tasks(3)
    Debug.Print objTask.TaskDependencies(1).To
    Set objTask = Nothing
End Sub

```

Die *Typ*-Eigenschaft bestimmt den Verknüpfungstyp und kann eine der *pjTaskLinkTyp*-Konstanten sein: *pjFinishToFinish*, *pjFinishToStart*, *pjStartToFinish* oder *pjStartToStart*.