

# Status

Unter [Workflow](#) werden für jeden Status den es in Primus geben soll ein eigener Abschnitt abgelegt. Der Name der Abschnitte ist frei wählbar und muss eindeutig sein. In Primus wird das Namensschema **Status\_<WorkflowStatus>** verwendet. Dadurch ist die Konfiguration übersichtlich und leicht anpassbar. Die Standardstatus in Primus sind:

Status	Name
001	Verfügbar
002	Wird geprüft
003	Freigegeben
004	Veraltet
005	Ungültig

In der Liste [List\\_WorkflowStatus](#) sind noch weitere Status definiert, diese werden z.B. für den [JobServer](#) gebraucht, haben hier aber keine Workflowdefinition.

## Beispiel

```
<Status_001>
```

## Aufbau eines Status

Ein Status besteht wiederum aus vier optionalen Abschnitten. Zwischen den Abschnitten sind [Kommentare](#) erlaubt, innerhalb der Abschnitte nicht.

	Abschnitt	Funktion
1	<a href="#">PreCalls</a>	Funktionen die vor dem zuweisen des Status aufgerufen werden und z.B. prüfen ob alle Vorbedingungen erfüllt sind
2	<a href="#">SetDataValues</a>	Enthält die Felder und die Werte die einem Dokument mit diesem Status zugewiesen werden
3	<a href="#">PostCalls</a>	Funktionen die nach dem Zuweisen dieses Status aufgerufen werden
4	<a href="#">TargetStatus</a>	Zielstatus den Dokumente mit diesem Status als nächstes annehmen können

Die Abschnitte werden in der aufgelisteten Reihenfolge abgearbeitet.

## SetDataValues

Hier werden den Datenfeldern Werte zugewiesen. Der Name des Tag ist der Name des Datenfelds, der Wert wird mit dem **Attribut value** zugewiesen. Eine wichtige Rolle spielt die Zuweisung **WorkflowStatus**, an diesem Feld erkennt Primus die Nummer des Status. Ohne eine Zuweisung dieses Wertes, wird nur ein leerer [Status Workflow Dialog](#) angezeigt.

## Beispiel

```
<!--Wertzuweisung beim Status_003-->  
<SetDataValues>  
  <WorkflowStatus value="003" />  
  <WriteStatus value="Read" />  
  <CheckedName value="%USER%" />  
  <CheckedDate value="%TODAY%" />  
  <MODIFIED value="0" />  
</SetDataValues>
```

## TargetStatus

In diesem Abschnitt wird definiert, welchen Folge- bzw. Zielstatus ein Dokument im [Lebenszyklus](#) als nächstes annehmen kann.

## Beispiel

```
<!--Folgestatus beim Status_003-->  
<TargetStatus>  
  <Status_001 condition="TRUE" direction="-1" />  
  <Status_005 condition="TRUE" direction="+1" />  
</TargetStatus>
```

## PostCalls

In **PostCalls** werden die [Methoden](#) aufgerufen, die abgearbeitet werden nach dem ein Dokument diesen Status erreicht hat. Es können [AddIns](#) Befehle, [externe](#) Programme oder prinzipiell auch [Menübefehle](#) sein. Mit [CreateJob\\_StatusPostCalls](#) können auch ein oder mehrere Methoden in eine Jobdatei geschrieben werden, welche dann von einem [JobServer](#) abgearbeitet wird.

## Funktionen für PostCalls

Funktion	Beschreibung
<a href="#">CopyFile</a>	Kopiert eine bestimmtes Dokument unter Angabe von Quell- und Zielpfad
<a href="#">SetParentToStatus</a>	Bezieht auch übergeordnete Elemente in den Statuswechsel mit ein und darf nur in den PostCalls verwendet werden
<a href="#">SetXMLValue</a>	Setzt ein bestimmtes Datenfeld eines Dokuments auf einen Wert
<a href="#">seUpdatePropertiesByPath</a>	Öffnet und aktualisiert Das Dokument unter dem angegebenen Pfad

Siehe auch [SEInterface](#).

## Beispiel

```
<!--Funktionsaufrufe beim Abschluss von Status_003-->
<PostCalls>
  <Call method="SEInterface.seUpdatePropertiesByPath"
parameter="%FILEPATH%|1" condition="$(Condition.isSEDocument)" />
  <Call method="SEInterface.seSetSolidEdgeFileStatusByPath"
parameter="%FILEPATH%|3" condition="$(Condition.isSEDocument)" />
  <Call method="SEJob_Export.seExportJobByPath"
parameter="%FILEPATH%|C300|SE-
&gt;PDF|. \attachments\%FILENAME%%EXTENSION%\#(Number)|XML_DocumentAttachment
s" condition="$(Condition.isSEDraft)" />
</PostCalls>
```

## PreCalls

**PreCalls** werden zu Beginn eines Status aufgerufen, hier kann geprüft werden ob alle Vorbedingungen erfüllt sind damit der Status dem Dokument zugewiesen werden kann. Ein typischer Befehl ist [CheckChildrenStatus](#), mit dem z.B. geprüft werden kann ob alle untergeordneten Dokumente ebenfalls freigegeben sind. Alle Prüfungen müssen als Ergebnis TRUE zurück liefern, sonst wird der Workflow für dieses Dokument abgebrochen. Liefert eine Prüfung FALSE als Ergebnis wird **PreCalls** sofort beendet und die weiteren Prüfungen werden nicht mehr durchgeführt.

## Funktionen für PreCalls

Funktion	Beschreibung
<a href="#">CheckChildrenStatus</a>	Prüft ob verlinkte untergeordnete Dokumente einen bestimmten Status haben
<a href="#">CheckParentsStatus</a>	Prüft ob verlinkte übergeordnete Dokumente einen bestimmten Status haben
<a href="#">CheckParentsWriteable</a>	Prüft ob übergeordnete Elemente beschreibbar sind
<a href="#">CheckXMLValue</a>	Prüft ob ein Feld einen bestimmten Wert hat
<a href="#">IgnoreStatus</a>	Dokumente mit bestimmten Status werden für den Workflow ignoriert
<a href="#">SetChildrenToStatus</a>	Bezieht auch untergeordnete Elemente in den Statuswechsel mit ein
<a href="#">SetPrevVersionToStatus</a>	Setzt das Dokument der vorherigen Revision auf einen bestimmten Status
<a href="#">SetSubFilesToStatus</a>	Bezieht auch abgeleitete Dokumente wie DXF / DWG in den Statuswechsel mit ein

## Beispiel

```
<!--Prüfung beim Aufruf von Status_003-->
<PreCalls>
  <Call method="CheckChildrenStatus"
parameter="%FILEPATH%|WorkflowStatus|003"
condition="$(Condition.isSEDocument)" />
</PreCalls>
```

From:

<https://dokuwiki.unitec.de/> - **Unitec Wiki für Produktdokumentationen**

Permanent link:

<https://dokuwiki.unitec.de/doku.php?id=primus:status>

Last update: **2023/07/20 11:59**

