

Bedingung

Bedingungen werden für Primus Funktionen verwendet um z.B. bestimmte Menüeinträge nur unter bestimmten Bedingungen anzuzeigen oder zu aktivieren. Es gibt mehrere Möglichkeiten Bedingungen zu definieren.

Globale Bedingung definieren

Im Abschnitt [Public](#) können Bedingungen definiert werden um sie an anderer Stelle in der Konfiguration zu nutzen. Dadurch erhöht sich die Lesbarkeit der Konfiguration und es werden Fehler durch unterschiedliche Vergleiche vermieden.

Das [Tag](#) heißt **Condition**, der Name der Bedingung ist das einzige [Attribut](#), der Wert besteht aus einem oder mehreren Vergleichen. Mehrere Vergleiche können durch **AND** oder **OR** verknüpft werden. Dabei müssen die einzelnen Vergleiche in ' eingeschlossen werden.

Vergleichsoperator	Funktion
==	Gleich
!=	Ungleich
←	Kleiner oder gleich
>=	Größer oder gleich
<	Kleiner als
>	Größer als
~	Enthält
!~	Enthält nicht

```
<!--Eine Bedingung die prüft ob es sich bei dem aktuellen Dokument um eine Solid Edge Datei handelt-->
<Condition isSEDocument="#(FILE_TYPE)==PAR' OR '#(FILE_TYPE)==ASM' OR '#(FILE_TYPE)==PSM' OR '#(FILE_TYPE)==DFT'" />
```

Direkt definieren

Eine Bedingung kann auch direkt in dem Parameter der *TRUE* oder *FALSE* erfordert, eingegeben werden. Beispiel:

```
<!--Prüfung ob der FILE_TYPE = PAR ist-->
<SAMPLE visible="#(FILE_TYPE)==PAR'" />
```

Primus interne Bedingungen

Es gibt auch Bedingungen die in Primus intern bereits vordefiniert sind. Die Bedingungen geben immer *TRUE* oder *FALSE* zurück. Optional gibt es Parameter mit denen die Bedingung näher

spezifiziert werden kann. Hier eine Liste der bisher bekannten:

Bedingung	Parameter	Erklärung
CheckFileLocalSite	0	
	1	
CheckActiveLicenseType	CAD	Prüft ob Primus mit einer bestimmten Lizenz gestartet wurde
	OFFICE	
	VIEWER	
	<LICFILENAME>	Beliebiger Name einer Lizenzdatei, z.B. JobServer (ab Primus 2021)
ContainsListValue	String List	Prüft ob in String einer der Einträge von List enthalten ist
IsActiveRegTab		Liefert den Namen des aktiven Datenregisters
IsCellForPaste		
IsCtrlKey		Abfrage ob die STRG-Taste gedrückt ist
IsDataForPaste		Sind Einfügedaten in der Zwischenablage
IsElementsForCopyPaste		
IsElementsForCopyPaste		
IsElementsForPaste		Kann in dieses Element eingefügt werden
IsJobServerEnabled		Ist die Jobserverfunktion aktiv
IsJobServerHost		Ist dies der Jobserver
IsListAllSubProjects		Ist die Listeneinstellung „alle Unterprojekte auflisten“ aktiv
ISLPrimusAddIn.islGetInsertFolder		klaerungsbedarf
IsNodeTypeActive	jobserver	
	recyclebin	Ist der Papierkorb gewählt
	stdparts	Ist der Knoten Standard Parts gewählt
	virtfolders	Ist der gewählte Knoten ein virtueller Ordner
IsProjectActive		Ist der aktiv gewählte Knoten „Project“
IsSearchServiceDBActive		Ist der PrimusSearchService aktiv/erreichbar
LocalWorkMode	0	Server Arbeitsweise aktiv
	1	Lokale Arbeitsweise aktiv
	2	Offline Arbeitsweise aktiv
PrimusSmartCopy.IsMarkTarget	0	Ist ein Einfügeordner gesetzt
	1	Ist ein Zielordner gesetzt
PrimusSmartCopy.IsSessionActive		Ist eine SmartCopy Session aktiv
PrimusSmartCopy.IsSmartCopySessionActive		Ist eine SmartCopy Sitzung aktiv
Workflow.newversioncond		klaerungsbedarf

Bedingung in einem Attribut nutzen

Die Bedingung wird aufgerufen mit '**CALL(<Bedingung>)**'

Beispiel:

```
<SAMPLE visible="'CALL(IsCtrlKey)'" />
```

Ausnahme

\$(Workflow.newversioncond)

Funktionen mit Rückgabewerten

Es gibt auch die Möglichkeit eine Funktion aufzurufen die als Rückgabewert *TRUE* oder *FALSE* zurück liefert. Können oder müssen mehrere Werte angegeben werden so sind diese mit dem entsprechenden Trennzeichen zu trennen. Im Attribut parameter ist dies ein |.

Funktion	Attribute	Funktion
CheckChildrenStatus	condition	Eine Bedingung die prüft ob diese Datei geprüft werden soll. Siehe PreCalls
	parameter 1	1 Pfad zur Datei, z.B. %FILEPATH%
		2 Feld dessen Wert verglichen werden soll, z.B. WorkflowStatus
	3	Vergleichswert für den Feldinhalt, mehrere mögliche Werte werden durch einen . voneinander getrennt, z.B. .003.004 .
CheckParentsStatus	condition	Eine Bedingung die prüft ob diese Datei geprüft werden soll. Siehe PreCalls
	parameter 1	1 Pfad zur Datei, z.B. %FILEPATH%
		2 Feld dessen Wert verglichen werden soll, z.B. WorkflowStatus
	3	Vergleichswert für den Feldinhalt, mehrere mögliche Werte werden durch einen . voneinander getrennt, z.B. .003.004 .
	4	Bedingung die das zu prüfende Dokument erfüllen muß, z.B. eine bestimmte Dateiendung
5	Text der im Fehlerfall angezeigt werden soll. Es kann ein Interfacetext eingetragen werden	
IgnoreStatus	condition	Eine Bedingung die prüft ob diese Datei geprüft werden soll. Siehe PreCalls
	parameter 1	1 Pfad zur Datei, z.B. %FILEPATH%
		2 Feld dessen Wert verglichen werden soll, z.B. WorkflowStatus
	3	Vergleichswert für den Feldinhalt, mehrere mögliche Werte werden durch einen . voneinander getrennt, z.B. .003.004 .

AND / OR Verknüpfung

Bedingungen können mittels **AND** und **OR** miteinander verknüpft werden. Dabei gilt die Regel AND vor OR, ähnlich wie Punkt- vor Strichrechnung in der Mathematik. Es ist aber auch möglich die Vergleiche in Klammern zu setzen, diese Klammern kommen in der Reihenfolge zuerst.

```
(' %EXTENSION%==.par' OR '%EXTENSION%==.asm') AND '#(Title)==''
```

Es wird zuerst verglichen ob es eine .par **oder** .asm handelt, **und** ob der Title leer ist.

Verschachtelte Verknüpfungen

Bei Verschachtelten Verknüpfungen kann es evtl. notwendig sein mit Klammern zu arbeiten.

Beispiel

```
<ReservationCall
condition="($(Condition.isSEDocument))AND'#(WorkflowStatus)!=003'"
<-- Definition der Bedingung -->
<Condition isSEDocument="#(FILE_TYPE)==PAR' OR '#(FILE_TYPE)==ASM' OR
'#(FILE_TYPE)==PSM' OR '#(FILE_TYPE)==DFT' OR '#(FILE_TYPE)==PWD'" />
```

Für die Verarbeitung werden von Primus intern die Bedingungen in ihre Strings aufgelöst, was dann so aussieht:

```
<ReservationCall condition="('#(FILE_TYPE)==PAR' OR '#(FILE_TYPE)==ASM' OR
'#(FILE_TYPE)==PSM' OR '#(FILE_TYPE)==DFT' OR
'#(FILE_TYPE)==PWD')AND'#(WorkflowStatus)!=003'"
```

Ohne die zusätzliche Klammer um die Bedingung `$(Condition.isSEDocument)` würde lediglich AND vor OR gelten, was letztlich dann auf

```
... OR'#(FILE_TYPE)==PWD'AND'#(WorkflowStatus)!=003'
```

Die UND Bedingung wäre also nur bei PWDs erfüllt worden, bei keinem anderen Dateityp.

Ausnahme bei Call

Werden Bedingungen mittels `Call` abgefragt, dann ist die Verwendung von Klammern nicht möglich. In dem Fall muss „ausgeklammert“ werden:

```
'CALL(IsNodeTypeActive stdparts)'AND'$(Condition.isCADLic)' OR
'CALL(IsProjectActive)'AND'$(Condition.isCADLic)'
```

Bedingte Wertzuweisung in Parametern mittels IF

Es ist möglich in Attributen die Parameter einzelner Funktionen sind, Werte mithilfe von Bedingungen zu definieren. Dies ist z.B. bei der Definition von Dateinamen wichtig. Die Bedingungen hat folgenden allgemeinen Aufbau `IF ('<WERT><VERGLEICHSOPERATOR><VERGLEICHSWERT' ?<TRUE FALL>:<FALSE FALL>)`

Ein verschachteln der IF Bedingungen ist nicht zulässig!

Beispiele

```

<!-- Wenn das FileNamePrefix leer ist, dann nimm den Counter, sonst nichts.
Wenn die Revision nicht leer ist, dann nimm - mit Revision dahinter, sonst
nichts. -->
<Kommando
filename_format="#(FileNamePrefix)IF('#(FileNamePrefix)=='?#(FileNameSuffix)
%COUNTER%: )IF( '%REVISION%!='?-%REVISION%: )" />
<!-- Macht ein Feld im Datenblatt beschreibbar (W), wenn der Benutzer Admin
ist, sonst nur lesbar (R) -->
<DataSheet_Feld mode_singlesel="IF('$ (Condition.isAdmin)'?W:R)" />

```

Liste der unterstützten Parameter

Die Auswertung von IF erfolgt nicht in allen Parametern, die Liste ist Möglicherweise noch unvollständig.

[Klärungsbedarf](#)

Parameter	Verwendung
mode_singlesel	Data Sheets
parameter	Diverse Befehle
filename_format	NewFileNameDialog

[menu](#), [GUI](#), [Kontext](#), [Bedingung](#), [Copy](#), [Liste](#), [Umstellung P7 auf P2021](#)

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

Permanent link: <https://dokuwiki.unitec.de/doku.php?id=primus:bedingung&rev=1723625647>

Last update: **2024/08/14 08:54**

